



# ترجمة انتقائية لمنهج سيسكو: اساسيات لينكس

Cisco: NDG Linux Essentials

الإصدار الثاني



المهندس لؤي ناصر باماقوس

الهداء

الحمد لله الذي هدانا لهذا الذي كنا لنهتدي لولا أن هدانا الله..

وإبتعثتني مدرسا..

وإعترضتني مبادرا..

الحمد لله المؤسسة العامة للتدريب التقني والمهني بكافة منسوبيها.

م. لؤي ناصر باماقوس

١٤٣٩/٧/٢٦هـ

## مقدمة

بسم الله الرحمن الرحيم والصلاة والسلام على خاتم النبيين سيدنا محمد وعلى آله وصحبه وسلم، أما بعد:

فقد خطت المملكة العربية السعودية في ظل قيادتنا الرشيدة خطوات واسعة من أجل تحقيق الرقي والرخاء للوطن والمواطن. وستكون رؤية ٢٠٣٠ خير شاهد على مستوى التطور والحضارة التي ستشهدها المملكة والتي ستأخذها إلى مصاف الدول المتقدمة من خلال تنويع مصادر الدخل والاهتمام بتنمية الكفاءات الوطنية للمساهمة في بناء اقتصاد قوي ومستديم. وتعتبر المؤسسة العامة للتدريب التقني والمهني من باكورة المشاريع الحقيقية للاستثمار في الإنسان السعودي في الاستراتيجيات السابقة واللاحقة وذلك بدورها الرائد بتدريب وتأهيل سواعد أبناء هذا البلد المعطاء من أجل إدارة الدفة الصناعية والحرفية بنا يحقق أعلى معايير الكفاءة والسلامة المهنية.

وفي ظل ذلك الزخم من الاهتمام والرعاية قامت المؤسسة العامة للتدريب التقني والمهني بإبرام عقود مع كبرى الشركات العالمية بهدف الرقي بمستوى أبنائنا المتدربين وتمكينهم من الحصول على المناهج الحديثة والمتطورة من خلال المراكز والأكاديميات الرائدة. ونظراً لأن بعض تلك المناهج لا توجد لها إصدارات تخدم اللغة العربية وما لذلك من تبعات تحول بين المتدرب وبين الاستقاء الكامل لتلك المعارف والمناهج، فقد بادرت بترجمة لأحد تلك المناهج بتصريف وهو منهج أكاديمية سيسكو: أساسيات نظام التشغيل لينكس ليصبح الحقيبة التدريبية المكافئة لمادة نظم تشغيل ٢.

يحتوي منهج أساسيات لينكس NDG Linux Essentials على ستة عشر فصلاً، ثلاثة فصول نظرية بحتة وهي الثلاثة الفصول الأولى، واما باقي الثلاثة عشر فصلاً وتحديداً من الفصل الرابع إلى الفصل السادس عشر فتشتمل على جزء نظري وجزء عملي. في هذا العمل قمت بترجمة الثلاثة الفصول النظرية الأولى وهي الفصل الأول والثاني والثالث، ثم ترجمة المعامل من المعمل الرابع إلى المعمل السادس عشر.

وأخيراً، أرجو من القارئ الكريم ابداء اي آراء أو ملاحظات على المحتوى على بريدي الالكتروني

[lbamagos@tvtc.gov.sa](mailto:lbamagos@tvtc.gov.sa)

والله ولي التوفيق...

## جدول المحتويات

٢	اهداء .....
٣	مقدمة .....
٤	جدول المحتويات .....
٧	تمهيد .....
٧	الهدف .....
٧	الأهداف التفصيلية .....
٧	الفئة المستهدفة .....
٨	المهارات المطلوب إتقانها .....
٨	الوقت المتوقع لإنهاء هذا المنهج: .....
٩	ما الجديد في الإصدار الثاني .....
٩	التطوير في الإصدارات القادمة .....
١٠	قبل البدء: معلومات عن دورة اساسيات لينكس .....
١١	الفصل الأول – مقدمة إلى لينكس .....
١٢	١,١ مقدمة .....
١٣	١,٢ تطور لينكس وأنظمة التشغيل المشهورة .....
١٩	١,٣ اختيار نظام التشغيل .....
٢٥	الفصل الثاني – تراخيص وتطبيقات المصدر المفتوح .....
٢٦	٢,١ مقدمة .....
٢٧	٢,٢ اهم تطبيقات المصدر المفتوح .....
٤٠	٢,٣ فهم البرمجيات المفتوحة المصدر والتراخيص .....
٤٨	الفصل الثالث – استخدام لينكس .....
٤٩	٣,١ مقدمة .....
٤٩	٣,٢ الوضع الرسومي مقابل غير الرسومي .....
٥٤	٣,٣ سطر الأوامر .....
٥٥	٣,٤ المحاكاة الافتراضية والحوسبة السحابية .....
٥٧	٣,٥ استخدام لينكس للعمل: .....
٥٩	٣,٦ حفظ حاسب لينكس الخاص بك بشكل آمن: .....
٦٢	٣,٧ حماية نفسك .....
٦٥	المعمل الرابع – أساسيات سطر الأوامر .....
٦٦	٤,١ مقدمة .....
٦٦	٤,٢ الملفات والمراجع .....

٧٠	٤,٣ متغيرات القشرة .....
٧٢	٤,٤ Globbing .....
٧٨	٤,٥ الاقتباس .....
٨٢	٤,٦ جمل التحكم .....
٨٥	٤,٧ سجل القشرة .....
٨٩	المعمل الخامس – الحصول على المساعدة .....
٩٠	٥,١ مقدمة .....
٩٠	٥,٢ الحصول على المساعدة .....
١٠٩	٥,٣ ايجاد الملفات .....
١١٢	المعمل السادس – العمل مع الملفات والمراجع .....
١١٣	٦,١ مقدمة .....
١١٣	٦,٢ الملفات والمراجع .....
١٢٣	٦,٣ سرد الملفات والأدلة .....
١٢٩	٦,٤ نسخ ونقل وإعادة تسمية الملفات والدلائل .....
١٣٩	المعمل السابع – الأرشفة والضغط .....
١٤٠	٧,١ مقدمة .....
١٤٠	٧,٢ أوامر الأرشفة .....
١٥٥	المعمل الثامن – القنوات القياسية والأنابيب وإعادة التوجيه .....
١٥٦	٨,١ مقدمة .....
١٥٦	٨,٢ قنوات سطر الأوامر وإعادة التوجيه .....
١٧١	٨,٣ استخدام الأمر find للبحث عن الملفات .....
١٧٦	٨,٤ رؤية الملفات النصية الكبيرة .....
١٩٠	٨,٥ البحث عن النص عن طريق التعبيرات العادية .....
١٩٨	المعمل التاسع – البرمجة الاساسية .....
١٩٨	Basic Scripting .....
١٩٩	٩,١ مقدمة .....
١٩٩	٩,٢ تحرير النص الأساسي .....
٢٣٢	٩,٣ برمجة القشرة الأساسية .....
٢٤٠	٩,٤ التنفيذ المشروط والمتكرر .....
٢٥٥	المعمل العاشر – فهم عتاد الحاسب .....
٢٥٦	١٠,١ مقدمة .....
٢٥٦	١٠,٢ قائمة عتاد الحاسب .....
٢٦٩	المعمل الحادي عشر – ادارة الحزم والعمليات .....
٢٧٠	١١,١ مقدمة .....

٢٧٠	..... /proc و النواة و ١١,٢
٢٧٤	..... إدارة العمليات ١١,٣
٢٨٢	..... استخدام أمر top لعرض العمليات ١١,٤
٢٨٨	..... استخدام pkill و kill لإنهاء العمليات ١١,٥
٢٩١	..... استخدام ps لتحديد وفرز العمليات ١١,٦
٢٩٥	..... عرض سجلات النظام ١١,٧
٣٠٥	..... المكتبات المشتركة ١١,٨
٣٠٨	..... المعمل الثاني عشر – تكوين الشبكة
٣٠٩	..... مقدمة ١٢,١
٣٠٩	..... استكشاف الشبكة ١٢,٢
٣٢٦	..... المعمل الثالث عشر – النظام وحماية المستخدم
٣٢٧	..... مقدمة ١٣,١
٣٢٧	..... حسابات المستخدم والمجموعة ١٣,٢
٣٣٦	..... من في هذا النظام ١٣,٣
٣٣٨	..... المستخدم الرئيسي أو الجذر ١٣,٤
٣٤٥	..... المعمل الرابع عشر – إنشاء مستخدم
٣٤٦	..... مقدمة ١٤,١
٣٤٦	..... إنشاء المستخدمين والمجموعات ١٤,٢
٣٥٧	..... المعمل الخامس عشر – الملكيات والاذونات
٣٥٨	..... مقدمة ١٥,١
٣٥٨	..... أذونات الملفات والملكيات ١٥,٢
٣٧٥	..... المعمل السادس عشر – أذونات خاصة، روابط ومواقع الملفات
٣٧٦	..... مقدمة ١٦,١
٣٧٦	..... الاطلاع على الأذونات الخاصة ١٦,٢
٣٨١	..... الروابط الصلبة واللينة ١٦,٣
٣٨٧	..... الملاحق
٤٣٠	..... المراجع

## تمهيد

### الهدف

منهج أساسيات لينكس هو منهج الكتروني موجود في موقع أكاديمية سيسكو العالمية في الشبكة العنكبوتية، ويهدف إلى تعليم المتدربين مفهوم البرمجيات المفتوحة المصدر وأساسيات نظام التشغيل لينكس، ثم الانتقال إلى القدرة على اتقان التعامل مع عمليات سطر الأوامر الأساسية.

### الأهداف التفصيلية

- التعرف على البرمجيات الحرة المفتوحة المصدر والتراخيص.
- التعرف على التطبيقات المفتوحة المصدر في مكان العمل وعلاقتها بالتطبيقات ذات المصدر المغلق.
- التعرف على المفاهيم الأساسية لنظام التشغيل لينكس من أجهزة وعمليات وبرامج ومكونات.
- القدرة على العمل على سطر الأوامر والملفات.
- القدرة على إنشاء واستعادة النسخ الاحتياطية المضغوطة وحفظها.
- القدرة على استكشاف وتكوين أجهزة الحاسب الالى على الشبكة.
- القدرة على تأمين النظام للمستخدمين والمجموعات وإعطاء أذونات ملفات الدلائل العامة والخاصة.
- القدرة على إنشاء وتشغيل البرامج النصية البسيطة.

### الفئة المستهدفة

يستهدف هذا العمل متدربي قسم الحاسب الالى بالكليات التقنية في تخصصي دعم فني وشبكات، حيث يدرس محتواه كمنهج لمادة نظم تشغيل ٢ الموجودة ضمن الخطة التدريبية لمرحلة الدبلوم، هذا المحتوى يعتبر اللبنة الاساسية لعالم الاحتراف في استخدام نظام التشغيل لينكس.

## المهارات المطلوب اتقانها

- اتقان استخدام برنامج القشرة.
- اتقان كيفية الحصول على المساعدة في الأوامر والبحث عن الملفات.
- اتقان كيفية التنقل وإدارة الملفات والدلائل.
- اتقان كيفية التعامل مع ملفات الأرشيف.
- اتقان كيفية إعادة توجيه حزم النص واستخدام التعبيرات العادية وأوامر تصفية الملفات النصية.
- اتقان كيفية استخدام محرر النصوص لإنشاء البرامج النصية في القشرة الأساسية.
- اتقان أوامر عرض المعلومات عن أجهزة الحاسب والعتاد.
- معرفة موقع النواة وبياناتها وايضا معلومات معالجة المكتبات وملفات السجل وحزم البرمجيات.
- معرفة كيفية حماية المستخدم للنظام.
- اتقان كيفية إنشاء حساب لمستخدم جديد، وإنشاء كلمة مرور وإجراء التعديلات لهذا الحساب.
- اتقان كيفية إعداد وعرض الملكيات وأذونات الملفات والدلائل.
- معرفة الأذونات الخاصة والأنواع المختلفة من ملفات الارتباط.

## الوقت المتوقع لإنهاء هذا المنهج:

يحتوي هذا المنهج على ستة عشر فصل ومعمل، الثلاثة الفصول النظرية يمكن تغطيتها في الاسبوعين الأولى من الفصل التدريبي، وتكرس بقية الاسبوع على المعامل والتي تحتاج لوقت اطول للممارسة والتدريب.

## ما الجديد في الإصدار الثاني

هذا هو الإصدار الثاني لترجمة منهج سيسكو أساسيات لينكس، وتم فيه تحديث مماثل للمنهج الرئيسي الموجود في اكااديمية سيسكو العالمية، بالإضافة إلى معالجة الملاحظات السابقة مثل:

- مراجعة شاملة كاملة للنص للتأكد من خلوه من الأخطاء.
- تحديث المحتوى تبعاً لمنهج اكااديمية سيسكو العالمية.
- إعادة شرح بعض المواضيع وتبسيط محتواها.
- مطابقة تنسيقات اكااديمية سيسكو.
- تغيير بعض المصطلحات المترجمة لتتواءم مع لمصطلحات المعربة سابقاً.
- تطعيم النص بالمزيد من المصطلحات الإنجليزية التخصصية.
- الاستغناء عن الصور قدر الإمكان وتحسين جودة ما تبقى.
- عمل فواصل بين الفصول.
- إعادة صياغة الملحقات وتصحيحها.

## التطوير في الإصدارات القادمة

- متابعة ما يتم من تجديد للمنهج في موقع اكااديمية سيسكو وتحديث المحتوى تبعاً لذلك.
- إضافة تمارين تدريبية على كل فصل أو معمل في المنهج.
- عمل فهرسة للأشكال الموجودة في المنهج.

## قبل البدء: معلومات عن دورة اساسيات لينكس

دورة أساسيات لينكس NDG Linux Essentials والمقدمة من قبل اكااديمية سيسكو العالمية متوفرة عبر الويب The web based content. ورغم ان المحتوى يتم عرضه عن طريق الانترنت، الا ان الجهاز الظاهري virtual machine معزول عن الإنترنت، حيث انه صُمم ليكون بيئة تعلم آمنة safe sandbox. يعمل الجهاز الظاهري virtual machine على نظام التشغيل أوبونتو Ubuntu الاصدار 12.10. والذي يستخدم نسخة مخففة من نظام التشغيل، حيث تم إزالة بعض الأوامر التي لم يتم شرحها في هذه الدورة.

النافذة الافتراضية Virtual Terminal تتوفر أثناء قراءة محتوى أي فصل أو معمل. ويتم فيها ادخال وتنفيذ الأوامر دون الحاجة الى تثبيت نظام تشغيل جديد او نظام افتراضي على الحاسب الخاص بك. وبما ان هذه النافذة افتراضية فهي لا تقوم بتخزين أي بيانات أو تعديلات فيها عند اغلاقها.

لعرض المحتوى التدريبي ووحدة تحكم لينكس Linux console viewer يجب استخدام أحد برامج استعراض الويب. اما الحد الأدنى من تلك البرامج فهو كالتالي:

- برنامج جوجل كروم Google Chrome الإصدار 30 أو أعلى.
- برنامج سفاري Safari الإصدار 6.1 أو أعلى.
- برنامج فاير فوكس موزيلا Mozilla Firefox الإصدار 25 أو أعلى.
- برنامج انترنت إكسبلورر Internet Explorer الإصدار 10 أو أعلى.

يحصل المتدرب بعد انتهاء البرنامج على شهادة من اكااديمية سيسكو العالمية تفيد بتجاوزه لهذا المقرر وانه مؤهل للتقديم على اختبار LPI Linux Essentials Exam ولحصول على شهادة اساسيات نظام التشغيل لينكس

.LPI Linux Essentials Exam

## الفصل الأول – مقدمة إلى لينكس

## Introduction to Linux

## ١,١ مقدمة

لينكس Linux هو نظام تشغيل مفتوح المصدر؛ بمعنى ان الكود أو التعليمات البرمجية الذي يحرك لينكس صُمم من قبل نفس المستخدمين للنظام، وبالتالي لا تمتلكه جهة معينة. وميزة هذا النظام انه يسمح لأي شخص استخدامه من غير تكاليف ترخيص، وأيضاً يتيح لذلك الشخص إمكانية تعديل الكود وتغييره بما يتوافق مع الاحتياجات المتجددة.

لينكس هي علامة تجارية مسجلة في الولايات المتحدة وبلدان أخرى للسويدي لينوس تور فالديس.

The infographic is titled "Linux is Open Source. What does that mean?". It features a red banner at the top with the title. Below it, a white box contains the text: "The code that powers Linux isn't owned by a company." To the right, another white box says: "Instead, it is developed by the community that uses it." Below this is a silhouette of a group of people sitting around a table. At the bottom, a dark blue banner asks "Why is this good?". Below that, a white box explains: "It frees users from licensing costs & allows the code to be modified to meet changing needs."

معلومات عن نظام التشغيل لينكس

## ١,٢ تطور لينكس وأنظمة التشغيل المشهورة

لينكس يعني نواة النظام kernel، وهي وحدة تحكم مركزية لكل ما يحدث على جهاز الحاسب الآلي. عندما نقول ان جهاز الحاسب "يعمل بنظام لينكس" فذلك اشارة إلى النواة ومجموعة من الأدوات tools التي تأتي معها (تسمى التوزيع distribution). وإذا كانت لديك خبرة في نظام التشغيل لينكس فأنت على الأرجح تتحدث عن البرامج نفسها.

مصطلح UNIX يربك قليلا، حيث ان يونيكس أصلا هو نظام تشغيل تم تطويره في مختبرات Bell Labs وAT&T وذلك في عام ١٩٧٠. ثم أجريت عليه تعديلات (تم التعديل من قبل المجتمع ثم أصبح ذلك التعديل أساس لأنظمة تشغيل أخرى) لينتج عن ذلك التعديلات في الوقت الحالي العديد من الأنواع المختلفة لنظام UNIX. ومع ذلك، يونيكس الآن علامة تجارية trademark ومحددة specification؛ ومملوكة من قبل اتحاد صناعي يسمى المجموعة المفتوحة Open Group. البرنامج الوحيد الذي تم اعتماده من قبل المجموعة المفتوحة هو UNIX. وعلى الرغم من اعتماد نظام لينكس جميع متطلبات مواصفات يونيكس الا انه لم يعتمد لديهم، لذلك لينكس في الحقيقة ليس يونيكس! ولكنه "مثل يونيكس" UNIX-like.

### ١,٢,١ دور النواة

نواة نظام التشغيل kernel تدير النظام؛ فمثلا تخصص للبرامج جزء من الذاكرة، وتقوم بتشغيل وانهاء البرامج، وتتعامل مع عرض النص على الشاشة. وعندما يحتاج تطبيق إلى الكتابة على القرص الصلب فيجب أن يطلب ذلك من نظام التشغيل. ايضا إذا قام تطبيقان بطلب نفس الموارد فان النواة هي التي تقرر من التطبيق الذي سيحصل عليها، وفي بعض الحالات تنهي تطبيق من أجل إنقاذ النظام.

أيضا من مهام النواة التبديل بين التطبيقات، يعني أن الحاسب يقوم بالعديد من التطبيقات المفتوحة في آن واحد، فتعالج النواة التنقل بين تلك التطبيقات. اما تعدد المهام الاستباقي pre-emptive multitasking يعني أن النواة هي التي تحدد موعد تبديل التركيز بين المهام وذلك لأن الحاسب لديه عدد محدود من الموارد كالذاكرة مثلاً؛ فلا يريد ان يستحوذ برنامج واحد على كل الموارد.

قد يظن كل تطبيق ان لديه جزء كبيرة من الذاكرة في النظام، لكن في الحقيقة النواة هي التي تحافظ على هذا الوهم.

عند بدء تشغيل الحاسب يتم تحميل جزء صغير من التعليمات البرمجية يسمى محمل الإقلاع boot loader. وظيفته تحميل نواة النظام وبدء التشغيل. حيث يقوم محمل الإقلاع بتحميل نواة لينكس، ثم نقل السيطرة إلى نظام التشغيل. ثم يستمر نظام التشغيل لينكس بتشغيل البرامج اللازمة لجعل الحاسوب مفيد، مثل الاتصال بالشبكة أو تشغيل خادم ويب.

## ١,٢,٢ التطبيقات

مثل وحدة تحكم الحركة الجوية، النواة ليست مفيدة من دون شيء للسيطرة عليه. فإذا كانت النواة هي البرج والتطبيقات هي الطائرات؛ فالتطبيقات ترسل طلباتها إلى النواة وذلك لاستقبال الموارد مثل الذاكرة ووحدة المعالجة المركزية.

العملية process هي مهمة تقوم النواة بتحميلها ثم تتبع نتائجها. وأحيانا يحتاج التطبيق إلى أكثر من عملية فالنواة هي التي تهتم بذلك.

## ١,٢,٣ دور المصدر المفتوح

بدأ لينكس في عام ١٩٩١ كمشروع هواية من قبل لينوس تور فالديس. وقام بعرض كود المصدر (التعليمات البرمجية) بحرية وانضم آخرون اليه لتشكيل هذا النظام.

مشاريع البرمجيات تأخذ شكل الكود المصدر source code، والذي هو عبارة عن مجموعة من تعليمات الحاسب المقروءة للإنسان والتي يمكن كتابتها بأي من لغات البرمجة المختلفة. الجدير بالذكر أن كود المصدر نظام التشغيل لينكس مكتوب بلغة C، مما يجعله شبيهاً بنظام يونيكس UNIX ايضاً.

الترجم compiler، يقوم بتحويل كود المصدر (الغير مفهوم مباشرة بواسطة الحاسب) إلى أوامر وتعليمات. حيث يجمع المحول كافة ملفات المصدر ويحولها إلى صيغة يمكن تشغيلها على جهاز الحاسب. مثل نواة لينكس.

المصادر المفتوحة open source هي أن لديك الحق في الحصول على البرنامج وتعديله كما تريد. فقام مستخدمو لينكس بأخذ كود المصدر وإجراء تغييرات عليه ثم مشاركته مرة أخرى. وقد أدى اعتماد لينكس على هذه الفلسفة إلى نجاح كبير.

تاريخياً؛ اغلب البرامج تكون مغلقة المصدر closed-source license، بمعنى أنك تستطيع تشغيل البرنامج لكنك لا تستطيع ان تعدل على الكود الاصلي المكون للبرنامج. وذلك عكس البرامج المفتوحة المصدر Open source والتي تخولك من الدخول على الكود الاصلي والتعديل فيه.

في البداية كان هذا النظام اسمه جنو GNU وليس يونيكس. بينما كان مطوري جنو يبنون نظام التشغيل الخاص بهم وجدوا انهم أكثر فعالية في بناء الأدوات التي تتماشى مع نظام التشغيل يونيكس ولكنهم لم يبنوا النواة بعد، فتم استخدام نواة لينكس والتي وفرة نظام تشغيل كامل. من ذلك يتضح لنا أن معظم الأدوات (التي هي جزء من نظام لينكس) تأتي من مشروع جنو.

هناك أنواع مختلفة من المصدر المفتوح – والتي ستدرس في فصل لاحق – تتفق جميعاً على وجوب حرية الوصول إلى كود المصدر، ولكن الاختلاف بينهم هو في كيفية ذلك الوصول؛ أو في ضرورة إعادة مشاركة أي تغييرات تمت بواسطة المجتمع.

#### ١,٢,٤ توزيعات لينكس

عند أخذ نواة لينكس وأدوات جنو، ثم إضافة تطبيقات مرئية للمستخدمين مثل عميل البريد الإلكتروني؛ عند اذن يصبح لدينا نظام لينكس كامل. وتقريباً منذ بدأ استعمال لينكس؛ بدأ المستخدمون بتجميع كل تلك التطبيقات والبرامج في حزمة متكاملة تسمى التوزيع distribution. التوزيع distribution يعني بتخزين وتركيب النواة، وتنزيل بقية البرنامج. وتشمل نسخة التوزيعات الكاملة على أدوات لإدارة النظام؛ وأيضاً على مدير مجموعة package manager للمساعدة في إضافة وإزالة البرامج بعد اكتمال التثبيت.

تختلف التوزيعات باختلاف استخدامها؛ فبعضها يهتم بالخوادم servers والبعض الآخر يهتم بالجهاز المكتبي desktops؛ وهناك توزيعات مصممة لأغراض تجارية كأدوات التصميم الإلكتروني أو الحسابات الاحصائية. ويمكن القول ان اهم توزيعيتين في فترة البدايات هما ريد هات Red Hat وديبيان Debian. وتوجد اختلافات

بينهما في كل شيء من مواقع الملفات file locations إلى الفلسفات السياسية political philosophies؛ ولكن الفرق الأكثر وضوحاً بينهما هو في مدير المجموعة package manager.

ريد هات Red Hat نظام تشغيل حاول تسويق لينكس المكتبي للأعمال التجارية. ومع مرور الوقت؛ بدأ ريد هات بزيادة التركيز على تطبيقات الخادم server مثل شبكة الانترنت، وخدمة الملفات، وأصدر ريد هات إنتربرايز لينكس Enterprise Linux وهي خدمة مدفوعة على دورة إصدار طويلة long release cycle. أما دور دورة الإصدار release cycle فهو لتوضيح متى يحتاج البرنامج إلى ترقية.

نظام Debian هو جهد مجتمعي وهو يشجع على استخدام البرمجيات المفتوحة المصدر. جاء ديبان مع نظام إدارة المجموعة الخاصة به استناداً إلى تنسيق ملف ديب deb file format، وهو عكس ريد هات حيث يدعم المعالجات التي لا تكون إنتل Intel أو أم د AMD.

لينكس العلمي Scientific Linux هو مثال آخر على التوزيعات المخصصة specific distribution والمبنية على ريد هات. والذي هو عبارة عن توزيعات distribution تتعامل مع الحسابات العلمية.

أوبونتو Ubuntu هو أكثر نظام تشغيل شعبية، وتوزيعاته مأخوذة من ديبان. شركة كانون يكال Canonical انشأته بغرض اخذ ارباح من تقديم الدعم.

لينوكس منت Linux Mint بدأ بمثابة فرع لأوبونتو لينكس Ubuntu Linux، حيث لا يزال يعتمد على مستودعات أوبونتو Ubuntu repositories. ولديه إصدارات مختلفة جميعها مجانية، ولكنها تحتوي على بعض برامج الترميز والملكية، والتي لا يمكن توزيعها دون قيود الترخيص في بعض البلدان. لينوكس منت سرعان ما حل كأفضل مكتبي في العالم لأوبونتو.

لقد ناقشنا التوزيعات المذكورة على وجه التحديد في أهداف أساسيات لينكس. يجب أن تكون على علم بأن هناك مئات - إن لم يكن الآف - من التوزيعات المتوفرة. من المهم أن نفهم أنه في حين أن هناك العديد من توزيعات المختلفة من لينكس، ولكن العديد من البرامج والأوامر كما هي أو مشابهة جداً لما سبق.

١,٢,٤,١ ما هو الأمر

الأمر `command` هو برنامج `software program` عند نفيذه على سطر الأوامر يقوم الحاسب بعمل معين، عندما تكتب امر فان العملية ككل `process` تنفذ بواسطة نظام التشغيل الذي يقرأ المدخلات ويعدل البيانات ثم يخرج النتائج. وبالتالي فان الأمر يقوم بتنفيذ عملية `process` لإنجاز مهمة معينة `job`. ايضا يمكن تعريف الأمر من زاوية اخرى وهي مكان الاصدار لذلك الأمر، فهناك مصادر مختلفة للأوامر داخل القشرة `shell` في سطر الأوامر `CLI` الخاص بك مثل:

- **الأوامر المضمنة في نفس القشرة shell:** وخير مثال هو الأمر `cd` لأنه جزء من قشرة الباش `bash shell`. عندما يكتب المستخدم الأمر `cd`، فان القشرة `shell` تنفذ الأمر لأنها تعرف تفسير هذا الأمر؛ ولا تتطلب برامج إضافية لذلك.
- **الأوامر المخزنة في ملفات وتكون القشرة مسؤولة عن البحث عنها:** مثل الأمر `ls` والذي عند كتابته تقوم القشرة بإرجاع قائمة الملفات الموجودة في ذلك المسار `PATH variable`.
- **الأسماء المستعارة Aliases:** وهي تستخدم لاستحداث أوامر جديدة وذلك بينائها من المهام والأوامر الموجودة `Functions`.
- **وظائف Functions:** ويمكن أيضا أن تبنى الوظائف باستخدام الأوامر الموجودة أيضا مثل الأسماء المستعارة `Aliases` وذلك لإنشاء أوامر جديدة بهدف تجاوز الأوامر المضمنة في القشرة `shell` أو الأوامر المخزنة في الملفات.

يتم تحميل الوظائف والأسماء المستعارة من ملفات التهيئة عندما تبدأ القشرة.

الاسم المستعار هو في الأساس اختصار لأمر أو سلسلة من الأوامر الأخرى. على سبيل المثال، فإن الأمر `cal` 2014 يعرض التقويم لعام ٢٠١٤. بدلا من تنفيذ الأمر الكامل في كل مرة، يمكنك إنشاء اسم مستعار يسمى `mycal`، كما هو موضح في الصورة التالية:

```
sysadmin@localhost:~$ alias mycal="cal 2014"
```

```
sysadmin@localhost:~$ mycal
2014
```

```

January                February                March
Su Mo Tu We Th Fr Sa  Su Mo Tu We Th Fr Sa  Su Mo Tu We Th Fr Sa
      1  2  3  4              1              1
5  6  7  8  9 10 11    2  3  4  5  6  7  8    2  3  4  5  6  7  8
12 13 14 15 16 17 18  9 10 11 12 13 14 15    9 10 11 12 13 14 15
19 20 21 22 23 24 25 16 17 18 19 20 21 22    16 17 18 19 20 21 22
26 27 28 29 30 31     23 24 25 26 27 28     23 24 25 26 27 28 29
                                30 31

April                    May                    June
Su Mo Tu We Th Fr Sa  Su Mo Tu We Th Fr Sa  Su Mo Tu We Th Fr Sa
      1  2  3  4  5              1  2  3              1  2  3  4  5  6  7
      6  7  8  9 10 11 12    4  5  6  7  8  9 10    8  9 10 11 12 13 14
13 14 15 16 17 18 19    11 12 13 14 15 16 17    15 16 17 18 19 20 21
20 21 22 23 24 25 26    18 19 20 21 22 23 24    22 23 24 25 26 27 28
27 28 29 30             25 26 27 28 29 30 31    29 30

July                      August                    September
Su Mo Tu We Th Fr Sa  Su Mo Tu We Th Fr Sa  Su Mo Tu We Th Fr Sa
      1  2  3  4  5              1  2              1  2  3  4  5  6

```

التقويم الميلادي لعام ٢٠١٤

١,٢,٥ منصات الأجهزة

بدا لينكس بدعم معالج ٣٨٦؛ ومع التقدم المستمر تطور النظام ليدعم منصات اخرى من شريحة انتل Intel chip إلى الحاسبات الكبيرة supercomputers.

ثم استخدم لينكس في أنظمة الهواتف الخلوية cellular phones واللوحية tablets. اخيرا بيعت إلى جوجل Google الذي جاء مع منصة أندرويد Android الذي هو عبارة عن لينكس والبرامج اللازمة لتشغيل الهاتف أو الحاسب اللوحي. حيث تستطيع الشركات المنتجة للهواتف المحمولة استخدامه كنظام لهواتفها، وبالتالي يكون

تركيزهم على انتاج التطبيقات لا على نظام التشغيل مما سهل عملية التصنيع. لهذا فإن أندرويد يعتبر أحد قادة الانظمة market leaders في عالم الهواتف.

يمكن ايضا العثور علي لينكس في العديد من الأجهزة الاستهلاكية consumer devices مثل موجهات الشبكة اللاسلكية Wireless routers والتي في كثير من الأحيان تعمل بنظام لينكس لأنه غني بمميزات الشبكة network features . ايضا هناك تيفو TiVo مسجل فيديو الرقمي خاص بالمستهلك consumer digital video recorder والذي يستخدم أيضا نظام لينكس.

### ١,٣ اختيار نظام التشغيل

لينكس هو نظام تشغيل مماثل لنظام يونيكس UNIX-like، مما يعني أنه ليس له شهادة رسمية، وبالتالي لا يمكن استخدام العلامة التجارية لنظام يونيكس. UNIX هناك العديد من البدائل الأخرى؛ بعضها مماثل لنظام يونيكس والبعض معتمد باسم كنظام يونيكس UNIX. وهناك أيضا أنظمة ليست مماثلة لنظام يونكس non-Unix مثل مايكروسوفت ويندوز.

السؤال الأهم لاختيار نظام التشغيل هو "ما الذي تفعله هذه الآلة؟".

#### ١,٣,١ نقاط القرار

إن أول شيء عليك أن تقررته هو ما الدور الذي سيؤدي به هذا الجهاز. هل ستعمل على وحدة التحكم console وتقوم بتشغيل التطبيقات الإنتاجية أو إنك ستصفح الانترنت؟ إذا كانت هذه هي الحالة فانت تحتاج إلى حاسب سطح مكتب. أما إذا كان سيستخدم كخادم ويب Web server وسيكون في رف الخوادم؛ فانت اذن تحتاج إلى خادم.

عادة الخوادم Servers تكون في رف rack وتتقاسم شاشة ولوحة مفاتيح مع العديد من أجهزة الحاسب الأخرى، ايضا وحدة التحكم console تستخدم فقط لإعداد واستكشاف الاخطاء وبالتالي سيعمل الخادم في الوضع الغير رسومي non-graphical mode. بعكس الحاسب العادي الذي يعمل بواجهة رسومية GUI.

ثانيا حدد وظائف الجهاز. هل هناك برامج معينة سيقوم الجهاز لتشغيلها، أو وظائف محددة عليه أن يفعلها؟ يجب عليك أيضا تحديد مدى الحياة lifetime والمخاطر المحتملة risk tolerance للخادم، فعادة أنظمة التشغيل والبرامج يصدر لها تحديث بشكل دوري وهو ما يسمى بدورة الاطلاق release cycle. لكن منتجي البرامج سيدعمون الإصدارات القديمة من البرامج لفترة من الوقت ثم لا يكون لها أي تحديث، وهو ما يسمى دورة صيانة maintenance cycle أو دورة الحياة life cycle. على سبيل المثال؛ التحديثات الرئيسية للينكس تصدر كل 6 أشهر تقريبا. الجدير بالذكر ان الإصدارات تعتبر منتهية الحياة (EOL) End of Life بعد تحديثين رئيسيين بالإضافة إلى شهر واحد، بحيث يكون المجموع ما بين 7 و 13 شهر بعد التنصيب قبل أن الحاجة إلى عمل ترقيية. على العكس من ذلك مع البديل التجاري مثل ريد هات إنتربرايز لينوكس Red Hat Enterprise Linux والذي يمكن أن تستخدمه لمدة 13 سنة قبل أن تحتاج إلى ترقية.

الصيانة ودورة الاصدار مهمة في الخادم التجاري enterprise server لأنه في بيئة تستهلك بالوقت، لكن الوقت، لكن الترقية فيه غير هامة لأن الخادم نفسه يستبدل في حال وجود تحديثات كبيرة في أنظمة التشغيل وتطبيقاتها، لذلك تكون دورة الاصدار القصيرة release cycle بطيئة، وذلك لأن التطبيقات غالبا ما تستهدف الإصدار الحالي من نظام التشغيل بغرض تجنب عناء ترقية الخوادم وأنظمة التشغيل باستمرار.

إذا كنت مطور للبرمجيات أو مستخدم تقليدي للينكس المكتبي فانت تحتاج لأحدث البرامج والتي تختلف في المظهر والوظائف، في الغالب يخزن المستخدم المكتبي عمله في أحد الخوادم وبالتالي يستطيع الترقية سواء نظام التشغيل أو البرامج دون القلق على عمله السابق.

ان أحد الأشياء العظيمة في المصدر المفتوح هو أن المطور يمكنه إطلاق البرنامج الجديد والحصول بسرعة على التغذية الراجعة feedback من المستخدمين. بيتا beta هو إصدار فيه العديد من الميزات الجديدة ولكن لم يتم اختبارها بدقة، بعد أن يتم اختبار النسخة يتم إطلاق اصدار جديد وهو المستقر stable. لذلك يمكن وصف إصدارات البرامج الفردية بانها اما بيتا beta أو مستقرة stable.

مفهوم التوافق backward compatibility وهو قدرة نظام التشغيل الجديد على التعامل مع البرامج التي انتجت لأنظمة التشغيل السابقة.

التكلفة cost هي دائما أحد العوامل المأخوذة في الحسبان عند الاختيار، على سبيل المثال نظام تشغيل لينكس في حد ذاته مجاني ولكن المستخدم قد يحتاج إلى دفع تكاليف للدعم اعتمادا على المميزات التي يختارها. وفي الجهة المقابلة أيضا مايكروسوفت Microsoft لديها تكاليف ترخيص الخادم وربما تكاليف الدعم الإضافية على مدى عمر الخادم.

١,٣,٢ مايكروسوفت ويندوز

مايكروسوفت قسمت انظمة التشغيل وذلك وفقا للغرض من الجهاز إلى: مكتبي desktop أو الخادم server؟ اصدارات ويندوز المكتبي مثل ويندوز ٨ تخرج كل ٣ إلى ٥ سنوات، وتميل إلى الدعم لسنوات عديدة. التوافق هو أيضا أولوية لمايكروسوفت، بحيث يمكن للمستخدمين تشغيل البرامج القديمة.

في عالم الخادم server هناك نظام التشغيل Windows Server (نسخة ٢٠١٢ للدلالة على تاريخ اطلاقه) وله واجهة المستخدم الرسومية، وهو منافس للينكس إلى حد كبير.

١,٣,٣ نظام تشغيل أبل OS-X

أبل Apple تجعل من OS-X نظام التشغيل، الذي خرج بشهادة UNIX. ويستند OS-X جزئيا إلى مشروع FreeBSD.

يعتبر OS-X نظام تشغيل مكتبي ولكن هناك حزم اختيارية تساعد في إدارة خدمات الشبكة، وذلك الحزم packages تسمح للعديد من الاجهزة المكتبية OS-X desktops بالتعاون مع بعضها البعض، مثل مشاركة الملفات أو تسجيل الدخول للشبكة.

استخدام OS-X على مكتبي عادة ما يكون قرار شخصي نتيجة سهولة الاستخدام. وبتزايد شعبية OS-X توفر دعم مناسب من المنتجين. أيضا OS-X ذا شعبية كبيرة في الصناعات الإبداعية مثل إنتاج الفيديو.

## ١,٣,٤: BSD

هناك العديد من المشاريع المفتوحة المصدر "توزيع برامج بيركلي Berkely Software Distribution"، مثل OpenBSD و FreeBSD و NetBSD، وهي تعتبر بدائل لنظام لينكس في كثير من النواحي لأنها تستخدم كمية كبيرة من البرامج المشتركة. تُدمج BSDs عادة في الخادم، وفي المقابل هناك أنواعا تم تطويرها للقيام بأدوار سطح المكتب مثل جنوم GNOME وكيدي KDE.

## ١,٣,٥: انواع تجارية أخرى من Unixes

بعض من انواع ينكس Unixes التجارية الأكثر شعبية هي:

• Oracle Solaris

• IBM AIX

• HP-UX

وكل منها يعمل بمنظور مخترعها أو مطوريها. وعادة يكون العتاد فيها hardware كبير وقوي، وتقوم بتقديم مميزات مثل التبديل الساخن للمعالج والذاكرة hot-swap CPU and memory، أو التكامل مع النظم المركزية القديمة legacy mainframe systems.

معظم المستخدمين يميلون إلى اختبار منتجات شركاتهم وفق ما يفضلون - إلا إذا كان هناك برنامج ما يتطلب أجهزة معينة أو ان تطبيق application يتطلب إضافة ملحقات إلى الأجهزة الموجودة. على سبيل المثال، يعمل IBM-AIX على مجموعة واسعة من عتاد IBM ويمكن مشاركة العتاد مع النظم المركزية mainframe. وهكذا، تجد AIX في الشركات التي لديها بالفعل بصمة كبيرة لـ IBM، أو التي تستخدم برنامج IBM مثل WebSphere.

## ١,٣,٦ لينكس

جانب واحد في لينكس يختلف عن غيره من البدائل وهو بعد اختيار المسؤول administrator لنظام لينكس لا يزال عليه اختيار التوزيعات distribution كما ذكر في الموضوع الأول عن التوزيعات distribution والنواة kernel والمرافق utilities وأدوات الإدارة management tools فيقوم المسؤول بتثبيت الحزم package وتحميل التحديثات update بعد التثبيت الأولي initial installation.

بعض أنظمة التشغيل متوفرة من خلال مورد واحد فقط مثل OS-X وWindows، ويكون دعمها من خلال نفس المورد. لكن في لينكس Linux هناك خيارات عديدة؛ وتبدأ من العروض التجارية المقدمة للخادم Server أو لجهاز مكتبي Desktop، إلى ان نصل إلى التوزيعات المعدلة والتي تحول جهاز حاسب قديم إلى جدار حماية للشبكة firewall network.

في كثير من الأحيان يختار المنتجين مجموعة فرعية من التوزيعات للدعم. وبما ان للتوزيعات إصدارات مختلفة من المكتبات الرئيسية فإنه من الصعب على الشركة تقديم الدعم لجميع تلك الإصدارات.

الحكومات والشركات الكبيرة أيضا تحد من خياراتها للتوزيعات لتختار ما يقدم له الدعم التجاري. وهذا أمر شائع في الشركات الكبيرة حيث الدفع للحصول على الدعم أفضل من المخاطرة بانقطاع واسع النطاق.

التوزيعات أيضا لديها دورات اصدار release cycles وتكون أحيانا كل ستة أشهر. فاذا كانت الترقية غير مطلوبة فان كل إصدار يمكن ان يكون معتمد لمدة معقولة من الزمن. لذلك بعض اصدارات لينكس لديها دعم طويل المدى (LTS) من ٥ سنوات أو أكثر في حين سيكون دعم الآخرين فقط لمدة عامين أو أقل.

بعض التوزيعات تميز الإصدار حسب المرحلة من الاستقرار والاختبار أو الغير مستقر. والفرق بينهم أن الاصدار الغير مستقر يقاوض الثقة بالميزات. فعندما يتم دمج الميزات في نظام لفترة طويلة وتظهر الكثير من المشاكل والاعطال؛ ينتقل البرنامج من خلال ذلك الاختبار إلى مرحلة الإصدار المستقر. توزيعة ديبان Debian distribution تحذر المستخدمين عن استخدام اصدار "sid" بالتحذير التالي:

" sid " يخضع لتغييرات واسعة النطاق وتحديث للمكتبات. هذا يمكن أن يؤدي إلى "عدم استقرار" النظام؛ والذي يحتوي على حزم لا يمكن تثبيتها بسبب النقص في المكتبات، لذلك فالنتائج قد لا تحقق المطلوب. لذلك استخدمه على مسؤوليتك الخاصة!

في الجهة الأخرى، يوجد توزيعات أخرى تعتمد على توزيعات بيتا Beta. على سبيل المثال توزيع فيدورا Fedora distribution أطلقت نسخة بيتا Beta أو نسخة ما قبل الاصدار pre-releases لبرنامجها قبل الاصدار الكامل للحد من الخلل. غالبا ما يعتبر المستخدمون فيدورا على أنها إصدار بيتا Beta لنظام ريدهات RedHat. وبعد إضافة أو تعديل المميزات في فيدورا تصبح جاهزة لتوزيع ريدهات RedHat distribution.

## ١,٣,٧ نظام اندرويد Android

يعتبر نظام اندرويد Android - برعاية جوجل - هو توزيعة لينكس الأكثر شعبية في العالم. وهو يختلف جوهريا عن نظرائه. لينكس هو النواة والعديد من الأوامر التي سيتم تغطيتها في هذه الدورة هي في الواقع جزء من حزمة جنو (GNU وليس يونكس). وهذا هو السبب في أن بعض الناس يصرون على استخدام مصطلح جنو / لينكس بدلا من لينكس فقط.

يستخدم اندرويد Dalvik virtual machine (الجهاز الظاهري دالفيك) مع لينكس، ويوفر منصة قوية للأجهزة المحمولة mobile devices مثل الهواتف phones والاجهزة اللوحية tablets. ومع ذلك، يفتقر اندرويد إلى الحزم التقليدية التي غالبا ما تأتي مع توزيعات لينكس (مثل جنو GNU وXorg)، اندرويد لا يتفق عموما مع توزيعات لينكس لسطح المكتب.

هذا التناقض يعني أن مستخدمي ريدهات RedHat أو أوبونتو Ubuntu لا يمكنهم تحميل البرنامج من متجر Google Play. وبالمثل؛ فإن محطة المحاكاة terminal emulator في اندرويد تفتقر إلى العديد من الأوامر عن نظرائه في لينكس. ومع ذلك فمن الممكن استخدام BUSYBOX مع اندرويد لتمكين معظم تلك الأوامر للعمل.

## الفصل الثاني – تراخيص وتطبيقات المصدر المفتوح

# Open Source Applications and Licenses

## ٢,١ مقدمة

في هذا الفصل سوف تصبح العديد من التطبيقات المفتوحة المصدر وادواته مألوفة لديك. أيضا سوف نناقش برمجيات المصدر المفتوح والتراخيص.

هل تعلم ان جميع المواقع التالية تعمل بنظام لينكس:

- amazon
- NETFLIX
- Tweeter
- facebook
- Google



مواقع تعمل على بيئة لينكس

## ٢,٢ اهم تطبيقات المصدر المفتوح

نواة لينكس Linux kernel تستطيع تشغيل مجموعة واسعة من البرامج عبر العديد من منصات العتاد hardware platforms. الحاسب يمكن أن يعمل كخادم server مما يعني أنه يعالج البيانات نيابة عن الآخرين، أو يمكن أن يعمل كجهاز مكتبي desktop للاستخدام المباشر مع المستخدم. الحاسب يستطيع تشغيل برامج أو ان يكون جهاز مطور development machine لإنشاء برامج. أيضا يمكنه لعب أدوار متعددة حيث ان لينكس Linux لا يميز دور الجهاز بل ولكن الأهم هي الإعداد والتي تحدد التطبيقات applications المراد تشغيلها.

الميزة هنا هو أنه يمكنك محاكاة simulate تقريبا جميع جوانب بيئة الإنتاج production environment، من تطوير development واختبار testing وتحقق verification وذلك بهدف تقليص الأجهزة وبالتالي توفير الوقت والتكاليف. لشخص يتعلم نظام لينكس Linux؛ يمكنك تشغيل تطبيقات الخادم على جهاز مكتبي أو على الخادم الافتراضي virtual server الذي يعمل على مزود خدمة الانترنت Internet Service Provider – (ISP). وبطبيعة الحال لن تكون قادراً على التعامل مع مزود كبير الحجم والذي لديه أجهزة ذات تكلفة مرتفعة. ولكن يمكنك محاكاة أي تكوين configuration دون الحاجة إلى أجهزة قوية أو ترخيص الخادم server licensing.

تندرج برامج لينكس عموماً ضمن الفئات الثلاث التالية:

- ❖ برنامج الخادم Server software – وهي البرامج التي لا تتفاعل مباشرة مع الشاشة ولوحة المفاتيح للجهاز الذي تعمل عليه. والغرض منها هو خدمة المعلومات في أجهزة الحاسب الأخرى وهم العملاء clients. أحيانا برنامج الخادم لا يتحدث إلى أجهزة الحاسب الأخرى ولكن فقط يتعامل مع البيانات.
- ❖ برامج مكتبية Desktop software – وهي التي يمكنك التفاعل معها مباشرة مثل متصفح الويب web browser ومحرر النصوص text editor ومشغل الموسيقى music player. في كثير من الحالات يتحدث البرنامج - مثل متصفح الانترنت - إلى الخادم server على الطرف الآخر ومن ثم يقوم بتفسير البيانات لك. في هذه الحالة فإن البرنامج المكتبي سيصبح هو العميل.

❖ أدوات Tools - وهي فئة واسعة من البرامج الغرض منها تسهيل إدارة النظام الخاص بك مثل الأداة التي تساعدك على تنسيق العرض، أو التي توفر لك قشرة لينكس Linux shell، أو حتى الأدوات الأكثر تطوراً والتي تحول كود المصدر source code إلى مادة يمكن تنفيذها على الحاسب.

أما بالنسبة إلى تطبيقات الجوال mobile applications فهي مشابهة لتطبيقات الجهاز المكتبي ولكن يتم تشغيلها على الهاتف الجوال mobile أو الحاسب اللوحي tablet بدلاً من جهاز الحاسب المكتبي desktop computer.

عادةً تستطيع تنفيذ أي عمل تريده في لينكس بواسطة عدد من التطبيقات applications. فهناك العديد من متصفحات الويب web browsers والعديد من خوادم الويب web servers والعديد من برامج تحرير النصوص text editors. وهذا لا يختلف عن المصدر المغلق closed source. ولكن الميزة في المصادر المفتوحة open source هي أنه إذا كان شخص ما لا تعجبه الطريقة التي يعمل بها خادم الويب web servers؛ فإنه يستطيع بناؤه بنفسه. وكلما تقدمت مع لينكس فستتعلم أيضاً كيفية تقييم البرامج المستخدمة.

## ٢,٢,١ تطبيقات الخادم

لينكس تتفوق في تشغيل تطبيقات الخادم server applications بسبب موثوقيتها reliability وكفاءتها efficiency. وعند النظر في برنامج الخادم server software؛ فإن السؤال الأكثر أهمية هو "ماهي الخدمة التي سأقوم بإدارتها؟" فمثلاً إذا كنت تريد خدمة صفحات الويب web pages؛ فسوف تحتاج إلى برنامج خادم الويب web server software وليس لخادم البريد mail server!

العمل كخادم ويب web servers هو أحد الاستخدامات المبكرة لنظام لينكس Linux والذي يستضيف فيه صفحات الويب web pages، ليتم بعد ذلك استعراضها بواسطة متصفح الانترنت من خلال بروتوكول نقل النص التشعبي (HTTP) أو نسخته المشفرة HTTPS. صفحة الويب web page نفسها يمكن أن تكون إحدى نوعين الأول هو ان تكون ثابتة static مما يعني أنه عندما يطلب متصفح الويب web browser الصفحة من خادم الويب web server فإن الخادم يرسل الملف كما يظهر على القرص disk. أما الثاني فهي المحتوى الديناميكي وهذا يعني أن يتم إرسال الطلب من قبل خادم الويب web server إلى التطبيق application

والذي سيقوم بتوليد المحتوى. برنامج وورد برس WordPress هو أحد الأمثلة المعروفة. حيث يمكن للمستخدمين تطوير المحتوى من خلال متصفح الانترنت الخاص بهم في تطبيق وورد برس WordPress والبرنامج يحوله إلى موقع تفاعلي functional website على شبكة الانترنت بشكل كامل. في كل مرة تتسوق عبر الانترنت فانت تبحث في موقع ديناميكي.

اباتشي Apache هو خادم الويب web server المهيمن حتى اليوم. كان أباتشي في الأصل مشروع مستقل ولكن المجموعة شكلت من قبل Apache Software Foundation وهم الآن لديهم أكثر من مئة مشروع برمجي مفتوح المصدر.

نجين إكس nginx هو خادم ويب web server آخر؛ وجاء من دولة روسيا. وهو يركز على الأداء performance من خلال الاستفادة من المزيد من انوية يونيكس الحديثة modern UNIX kernels أما على مستوى المهام، فإنه يقوم بعمل مهام فرعية subset مقارنة بالمهام التي يمكن لأباتشي فعلها. الجدير بالذكر ان أكثر من ٦٥٪ من المواقع تعمل إما بخادم الويب نجن إكس nginx أو أباتشي Apache.

ولنتقل الآن من خوادم الويب إلى البريد الإلكتروني، غالبا ما تستخدم خوادم لينكس Linux servers في خدمة البريد الإلكتروني. عند مناقشة هذه خدمة فإنه من المفيد دائما أن ننظر إلى ثلاثة أدوار مختلفة؛ لأنها حيوية للحصول على البريد الإلكتروني بين المستخدمين:

١ - وكيل نقل البريد (MTA) – Mail Transfer Agent – ليقوم بمعرفة أي خادم server يحتاج

لاستقبال البريد الإلكتروني e-mail، ويستخدم بروتوكول نقل البريد البسيط Simple Mail

Transfer Protocol – (SMTP) لنقل البريد الإلكتروني إلى ذلك الخادم server. ليس من

الغريب للبريد الإلكتروني e-mail ان يستغرق عدة قفزات hops (بين الموجهات Routers)

للوصول إلى وجهته النهائية، وذلك لأن المنشأة قد يكون لها عدة وكلاء نقل بريد MTAs.

٢ - وكيل تسليم البريد (MDA) – Mail Delivery Agent وايضا يسمى وكيل تسليم محلي Local

Delivery Agent وهو يقوم بتخزين البريد الإلكتروني e-mail في علبة البريد الخاصة بالمستخدم

user's mailbox. وعادة ما يتم استخدامه في اخر وكيل نقل بريد MTA في السلسلة.

٣- خادما POP وIMAP – وهما بروتوكول مكتب البريد Post Office Protocol، والثاني هو

بروتوكول الوصول إلى الرسائل عبر الانترنت Internet Message Access Protocol -

وهما بروتوكولا اتصال حيث يسمحان للبريد الإلكتروني الخاص بالعميل email client والذي يعمل

على جهازه من مخاطبة الخادم البعيد remote server وذلك لالتقاط البريد الإلكتروني.

أحيانا جزء من برنامج ما ينفذ بمكونات متعددة. في البيئة المغلقة المصدر؛ Microsoft Exchange يقوم بإعطاء كافة المكونات بحيث لا يكون هناك أي خيار آخر. اما في عالم المصدر المفتوح فهناك العديد من الخيارات، فبعض خوادم POP / IMAP تنفذ لتنسيق قاعدة بيانات database البريد الخاصة بهم للحصول على ميزة الأداء، أو أيضا ستحتوي على وكيل تسليم البريد MDA إذا كان ذلك رغبة في قاعدة البيانات المخصصة custom database. بالإضافة إلى ذلك؛ يستطيع من يستخدم صيغ الملفات القياسية (مثل رسائل البريد الإلكتروني كاملة في ملف نصي واحد) ان يختار أي وكيل تسليم بريد MDA.

يعتبر sendmail أكثر وكيل نقل بريد MTA معروف. كما يوجد أيضاً وكيل مشهور آخر وهو Postfix حيث وجد ليكون مشابه لوكيل sendmail ولكن أكثر أمانا.

إذا كنت تستخدم صيغ الملفات القياسية standard file formats لتخزين رسائل البريد الإلكتروني e-mail فيمكن لوكيل نقل البريد MTA الخاص بك أيضا ارسال البريد. هناك ايضا procmail؛ والذي يتيح لك تصفية filter مخصصة لمعالجة البريد وتنظيمه.

Dovecot هو خادم POP / IMAP مشهور بسبب سهولة استخدامه وصيانتته المنخفضة. Cyrus IMAP هو خيار آخر.

بعد ذلك مشاركة الملف، في مجال مشاركة الملفات فإن الفائز بلا منازع هو سامبا Samba. حيث يسمح لجهاز لينكس Linux machine ليظهر كأنه جهاز ويندوز Windows machine وبذلك تتم مشاركة الملفات share files في مجال Windows. سامبا Samba ينفذ يتم دور مكونات الخادم، مثل جعل الملفات متاحة للمشاركة وكذلك بعض مهام الأخرى لخادم ويندوز Windows server roles، نقطة أخيرة، يقوم سامبا بعملية انهاء العميل client end حتى لا يستهلك جهاز لينكس قد حصة نظام ويندوز الخاصة بمشاركة الملفات.

إذا كان لديك أجهزة أبل على الشبكة فان مشروع Netatalk يجعل جهاز لينكس الخاص بك يتصرف كما انه خادم ملف أبل Apple file server.

يطلق على بروتوكول مشاركة الملفات الأصلي لنظام UNIX اسم نظام ملفات الشبكة Network File System (NFS) - System، وعادة ما يكون نظام NFS جزءا من النواة kernel، مما يعني أن نظام الملفات عن بعد remote file system يمكن تركيبه مثل القرص العادي، مما يجعل الوصول إلى الملفات مباشر ومتاح لتطبيقات أخرى.

كلما كبرت شبكة الحاسب الخاصة بك كلما احتجت إلى وجود مرجع directory. ويطلق على أقدم مرجع directory نظام أسماء المجال Domain Name System- (DNS)؛ والذي يستخدم لتحويل اسم مثل <http://www.linux.com> إلى عنوان IP كالتالي 100,100,168,192، وهو معرف فريد لهذا الجهاز على شبكة الانترنت. كما يحتوي نظام DNS على معلومات عالمية مثل عنوان وكيل نقل البريد MTA لاسم مجال معين. وهناك منظمات تدير خادم DNS خاص بها، بغرض استضافة الأسماء وأيضا ليكون بمثابة دليل داخلي internal directory للخدمات. (ISC)-The Internet Software Consortium تحفظ تقريبا جميع خوادم DNS الأكثر شعبية، وتسمى bind. وذلك على اسم العملية التي تقوم بتشغيل الخدمة.

نظام أسماء المجال DNS يركز بشكل كبير على أسماء أجهزة الحاسب وعناوين IP والتي ليس من السهل البحث عنها. ظهرت مراجع directories أخرى تقوم بتخزين المعلومات الأخرى مثل حسابات المستخدمين والأدوار الأمنية. بروتوكول الوصول الخفيف إلى الدليل Lightweight Directory Access Protocol - (LDAP) هو الدليل الأكثر شيوعا والذي يدير أيضا دليل مايكروسوفت النشط Microsoft's ActiveDirectory. في LDAP يتم تخزين أي كائن object في شجرة، ومكان هذا الكائن يمكن أن يستخدم لاستخلاص المعلومات حول الكائن بالإضافة إلى ما تم تخزينه مع الكائن نفسه. على سبيل المثال؛ يمكن تخزين مسؤول لينكس في فرع من شجرة يسمى "قسم تقنية المعلومات"، الفرع السابق موجود تحت فرع يسمى "العمليات". وبهذا التسلسل نستطيع ايجاد كل أعضاء الجهاز الفني من خلال البحث ضمن فرع قسم تقنية المعلومات. هنا OpenLDAP هو اللاعب المهيمن.

جزء اخير من البنية التحتية للشبكة هو بروتوكول التكوين الديناميكي للمضيف Dynamic Host Configuration Protocol – (DHCP). عندما يقلع boots up جهاز الحاسب فإنه يحتاج إلى عنوان IP للشبكة المحلية Local Network بحيث يمكن تحديده بشكل فريد. وظيفة DHCP هي الاستماع للطلبات وتعيين عناوين شاغرة من تجمع البروتوكول DHCP pool. ايضا تحتفظ The Internet Software Consortium - (ISC) بخادم DHCP ISC وهو اللاعب الأكثر شيوعا هنا.

قاعدة البيانات database تخزن المعلومات stores information وتسمح باسترجاعها والاستعلام عنها بكل سهولة. قواعد البيانات الأكثر شعبية هنا هي MySQL و PostgreSQL . تستخدم لغة تسمى لغة الاستعلام الهيكلية (SQL) – Structured Query Language على قاعدة بيانات بها أرقام مبيعات أولية وذلك من أجل إعداد تقرير بالمبيعات الإجمالية حسب ما هو مطلوب كمبيعات منتج معين أو مبيعات لتاريخ محدد.

## ٢,٢,٢ تطبيقات سطح المكتب

نظام لينكس يحتوي على مجموعة كبيرة من تطبيقات سطح المكتب. يمكنك أن تجد ألعاب تسلية وتطبيقات إنتاجية وأدوات إبداعية وأكثر من ذلك. هذا القسم مجرد عرض عام لما يوجد في الساحة؛ مع التركيز على البرامج الأكثر أهمية على حسب وجهة نظر معهد لينكس المهني Linux Professional Institute LPI.

أولا سنستعرض بيئة الجهاز المكتبي قبل النظر في التطبيقات الفردية، يعمل نظام لينكس المكتبي Linux desktop بنظام يسمى نافذة X (X-Window) والمعروف أيضا باسم X11. خادم لينكس X11 هو X.org يوفر وسيلة للبرامج تمكنها من العمل في الوضع الرسومي graphical mode واستقبال المدخلات من لوحة المفاتيح keyboard والفأرة mouse. يتم التعامل مع النوافذ والأيقونات من خلال جزء برمجي اخر يسمى مدير النوافذ window manager أو بيئة الجهاز المكتبي desktop environment. مدير النافذة window manager هو برنامج يقدم كود لعمل القوائم والتحكم في نوافذ التطبيقات الموجودة على شاشة سطح المكتب. بيئة سطح المكتب لها طبقات؛ وكل طبقة تقدم مميزات مثل شاشة تسجيل الدخول والجلسات ومدير الملفات والمرافق utilities الأخرى. وخلاصة القول انه يمكن تحويل محطة عمل لينكس النصية إلى جهاز مكتبي رسومي وذلك بإضافة نافذة X ثم اختيار إما بيئة مكتبية أو مدير النافذة.

والآن نتوسع في مدير النافذة، يتضمن مدير النافذة على التطبيقات Compiz و FVWM والتتوير Enlightenment. بيئات الجهاز المكتبي الرئيسية هي KDE و GNOME وكلاهما له مدير نافذة خاصة به. كل من KDE و GNOME مشاريع كاملة مع كمية هائلة من البرامج التي بنيت لأجلهما، والاختيار عادة يكون حسب التفضيل الشخصي للمستخدم.

ننتقل بعد ذلك إلى التطبيقات الإنتاجية الأساسية basic productivity applications مثل معالج النصوص وجدول البيانات وعارض الشرائح فإنها موجودة على شكل مجموعة كاملة وتسمى بجناح المكتب office suite، لكن يبقى مايكروسوفت أوفيس Microsoft Office اللاعب المهيمن في السوق.

يقدم برنامجي أوبن أوفيس OpenOffice (ويسمى أحيانا OpenOffice.org) وبرنامج LibreOffice مجموعة مكتب كاملة بما في ذلك أداة للرسم والمتوافقة مع Microsoft Office سواء من حيث الشكل والمميزات. هاذان المشروعان هما مثالين عظيمين لتأثير سياسة المصدر المفتوح.

والآن لنستعرض مزيد من التفاصيل عن برنامج ليبر أوفيس LibreOffice. في عام ١٩٩٩ شركة صن مايكروسيستمز Sun Microsystems قامت بشراء شركة المانية غامضة نسبيا كانت تعمل على مجموعة مكتب office suite لنظام لينكس تحت مسمى ستار أوفيس StarOffice. بعد فترة وجيزة اعاد الشركة التسمية واصدرت أوبن أوفيس OpenOffice وبرخصة مفتوحة المصدر، ايضا بقي ستار أوفيس StarOffice منتج شخصي للشركة. وفي عام ٢٠١٠ اشترت شركة أوراكل Oracle شركة صن مايكروسيستمز ثم حول المشروع إلى مؤسسة أباتشي Apache Foundation.

شركة أوراكل Oracle ليس لها في دعم مشاريع المصدر المفتوح التي تملكها، لذلك وبعد فترة وجيزة من الاستحواذ تغير المشروع ليصبح ليبر أوفيس LibreOffice. في تلك المرحلة أصبح هناك مجموعتان من الناس يطورون نفس البرمجيات. ذهب أكثر الدعم لمشروع ليبر أوفيس LibreOffice ولهذا السبب تم تضمينه افتراضيا في العديد من توزيعات لينكس.

والآن ننتقل إلى تصفح شبكة الانترنت، يوجد في عالم لينكس متنافسين رئيسيين هما فايرفوكس Firefox وجوجل كروم Google Chrome. كلاهما مفتوح المصدر ويمتازان بسرعة العمل وكثرة المميزات ووجود دعم ممتاز

من المطورين. المستعرضان السابقان هما مثال جيد على التنوع في المصدر المفتوح. فإدخال التحسينات على أحدهما؛ يحفز الفريق الآخر لتقديم الأفضل. ونتيجة لذلك فإن شبكة الانترنت لديها متصفحين ممتازين للعمل على شبكة الانترنت بأفضل شكل، والتي تعمل عبر مجموعة متنوعة من المنصات.

والآن نتحدث عن البريد الإلكتروني، جاء مشروع موزيلا Mozilla من ثندربيرد Thunderbird، وهو نسخة كاملة لعميل البريد الإلكتروني على سطح المكتب. ثندربيرد يتصل بخادم POP أو IMAP، وهو يقوم بعرض البريد الإلكتروني email محليا، ويرسل البريد الإلكتروني من خلال خادم SMTP خارجي.

بالإضافة إلى موزيلا Mozilla فهناك عميلي البريد الإلكتروني في جنوم وكيدي GNOME and KDE و هما Evolution و KMail . المعيار القياسي Standardization في بروتوكول POP وبروتوكول IMAP وفي شكل البريد الإلكتروني المحلي يعني أنه من السهل التبدل بين عملاء البريد الإلكتروني دون فقدان البيانات. البريد الإلكتروني بواسطة الويب Web based email هو أيضا خيار آخر.

أخيرا الأنواع الإبداعية. في هذه الفئة نجد Blender و GIMP و Audacity والتي تقوم بإنشاء أفلام ثلاثية البعد 3D-movie ومعالجة الصور ثنائية البعد 2D-image ، وتحرير الصوت على الترتيب. تلك الأنواع لديها درجات متفاوتة من النجاح في الأسواق المهنية. لكن نستطيع القول أن Blender يُستخدم في كل شيء بداية من الأفلام المستقلة إلى افلام هوليوود السينمائية.

### ٢,٢,٣ أدوات وحدة التحكم

يظهر في تاريخ تطور يونيكس UNIX تداخل كبير بين مهارات تطوير البرمجيات software development skills وإدارة النظم systems administration. الأدوات التي تمكنك من إدارة النظام لها مميزات لغات الحاسب مثل الحلقات loops، وتستخدم بعض لغات الحاسب على نطاق واسع في انجاز مهام ادارة النظم بشكل تلقائي automating systems administration tasks؛ وبالتالي ينبغي أن تأخذ هذه المهارات المكملة بعين الاعتبار.

على المستوى الأساسي، يمكنك التفاعل مع نظام لينكس من خلال القشرة shell حيث لا يهم إذا كنت تحاول الاتصال بالنظام عن بعد أو من خلال لوحة المفاتيح المرفقة. وظيفة القشرة shell هي قبول الأوامر commands مثل تعديل الملفات file manipulations وفتح التطبيقات ثم بعد ذلك تمريرها إلى نواة لينكس لتقوم بتنفيذها Linux kernel. وهنا مثال يبين لنا تفاعل نموذجي مع قشرة لينكس Linux shell:

```
sysadmin@localhost:~$ ls -l /tmp/*.gz
-rw-r--r-- 1 sean root 246841 Mar 5 2013 /tmp/fdboot.img.gz
sysadmin@localhost:~$ rm /tmp/fdboot.img.gz
```

مثال على أوامر موجهة لقشرة لينكس

يقوم المستخدم بالتعامل مع موجه الأوامر prompt، والذي ينتهي عادة بعلامة الدولار (\$) للإشارة إلى الحساب العادي والغير مميز unprivileged account. أما ما يسبق كتابة الأوامر وهو في هذا المثال sysadmin@localhost:~ فهي اعدادات الموجه والتي توفر معلومات إضافية للمستخدم. في الشكل أعلاه؛ sysadmin هو اسم المستخدم، أما localhost فهو اسم الخادم، أما علامة تيلدا ~ فهي الدليل الحالي directory (في يونيكس، الرمز تيلدا tilde هو رمز مصغر لدليل المستخدم الرئيسي). وسوف نتعرف على أوامر لينكس بمزيد من التفصيل في الفصول القادمة، ولكن ولأنهاء شرح المثال السابق، فإن الأمر ls يعرض قائمة بالملفات؛ حيث يستقبل بعض المعلومات حول الملف؛ وفي السطر الثالث من الشكل تتم إزالة هذا الملف مع امر الحذف rm.

توفر قشرة لينكس Linux shell لغة غنية بالتكرار على الملفات iterating over files، وميزة تخصيص بيئة العمل customizing the environment، إضافة إلى ذلك فأنا نجد جُمْل برمجية syntax مدمجة built-in scripting language. على سبيل المثال؛ يمكن كتابة سطر واحد لإيجاد الملفات التي تحتوي على تطابق لنمط معين؛ واستخراج المعلومات المفيدة من ملف، واخيراً نسخها إلى ملف جديد.

لينكس يوفر مجموعة متنوعة من القشور shells للاختيار منها، والاختلاف في الغالب يكون في ماهية الأشياء التي يمكن تخصيصها وكيف، بالإضافة إلى نوعية الجمل المضمنة في اللغة. هناك فئتان رئيسيتان هما قشرة بورن Bourne shell وقشرة C. سميت قشرة بورن Bourne على اسم مصممها، أما قشرة C فبسبب

تسميتها ان الجُمْل البرمجية فيها اخذت الكثير من لغة C. الجدير بالذكر ان كلا القشرتين اخترعتا عام ١٩٧٠. في الوقت الحالي توجد إصدارات جديدة لهما هما قشرة الباش (Bash) Bourne Again Shell وقشرة tcsh (tee-cee-shell). الباش Bash هو القشرة الافتراضية التي تأتي مع اغلب النظم، ولكنك تستطيع استخدام قشرة tcsh إذا كانت هي المفضلة لديك.

بعض المستخدمون أخذوا المميزات المفضلة لهم من قشرتي Bash و tcsh وقاموا بصنع قشور اخرى، مثل قشرة كورن Korn shell (ksh) وايضاً قشرة zsh. اختيار القشرة shell في الغالب مسألة شخصية. إذا بدأت مع قشرة الباش Bash وتمكنت من التأقلم معها؛ فيمكنك العمل بفعالية على معظم أنظمة لينكس. بعد ذلك يمكنك التغيير وتجربة قشور اخرى لمعرفة ما إذا كانت تساعد انتاجيتك.

بعد ذلك يأتي اختيار برامج تحرير النصوص text editors. يستخدم محرر النصوص في وحدة التحكم console لتعديل ملفات التكوين configuration files. وبصفة عامة يوجد نوعين رئيسيين من محرري النصوص؛ الأول هو فاي vi أو فايم vim والثاني هو إيماكس emacs. كلاهما يعتبران أداتين قويتين لتحرير الملفات النصية، لكنهما يختلفان في شكل الأوامر وكيفية كتابة الإضافات plugins لهما. الإضافات يمكن أن تكون أي شيء من الجمل الواضحة syntax highlighting لمشاريع البرمجيات software projects إلى التقويمات السنوية الكاملة integrated calendars.

التعامل مع فايم vim و إيماكس emacs صعب وله مستويات تعلم مرهقة. لذلك فهما غير مفيدان إذا كان كل ما تحتاجه هو تحرير بسيط من ملف نصي صغير. لذلك هنالك برنامجا بيكو pico و نانو nano والموجودان في معظم الانظمة (وهذا الأخير هو مشتق من السابق)، ويوفران تحرير نص أساسي فقط.

حتى إذا اخترت عدم استخدام محرر فاي vi؛ فإنه يجب عليك ان تتعرف عليه لأنه موجود في كل انظمة لينكس. الجدير بالذكر انه إذا كنت تريد استعادة نظام لينكس لإصلاحه عن طريق تشغيل أداة استرداد النظام الموزع distribution's recovery فيجب أن يكون لديك المحرر فاي vi.

في انظمة التشغيل عادة ما نقوم بإضافة البرامج install وتحديثها update أو حتى إزالتها remove. وهذا يعني تحميل شفرة المصدر source code ثم بعد ذلك تنصيبها والبناء عليها واخيرا نسخ الملفات على كل

نظام. لحسن الحظ هناك توزيعات distributions صنعت حزم packages والتي هي عبارة عن نسخ مضغوطة من التطبيق. مدير الحزم package manager يعتني بحفظ مسار الملفات التي تنتمي لحزمة ما وايضا تنزيل التحديثات من خادم بعيد يسمى المستودع repository. أنظمة دبيان Debian تشتمل على الأدوات dpkg و apt-get و apt-cache. اما الانظمة المشتقة من ريد هات Red Hat؛ فيوجد فيها rpm و yum.

## ٢,٢,٤ أدوات التطوير

ينبغي ألا تكون متفاجئا عندما تعلم أن مثل هذه البرمجيات (أدوات التطوير) مبنية على إسهامات من المبرمجين، وان نظام لينكس Linux لديه دعم ممتاز لتطوير تلك البرمجيات. يتم بناء القشور shells لتكون قابلة للبرمجة programmable بالإضافة إلى وجود برامج تحرير editors قوية في كل نظام. هناك أيضا العديد من أدوات التطوير development tools المتاحة، بالإضافة إلى ان وجود العديد من اللغات الحديثة والتي تتعامل مع لينكس على انه أول اهتماماتها.

توفر لغات الحاسب Computer languages وسيلة للمبرمج programmer لإدخال التعليمات في شكل مقروء للإنسان human readable format، ثم بعد ذلك يتم ترجمتها translated إلى مادة يفهمها الحاسب. اللغات تقع ضمن مجموعتين: تفسير interpreted أو تجميع للتنفيذ compiled. interpreted. تترجم الكود code المكتوب إلى كود حاسب يعمل كبرنامج، واما compiled فيقوم بترجمة اللغة بالكامل بغرض التنفيذ الفوري.

سنحدث الآن عن عدد من لغات البرمجة، الأولى هي لغة الترجمة C (C compiled language) حيث قد تم كتابة نظام التشغيل لينكس Linux بهذه اللغة. الفائدة الرئيسية من C أن مفرداتها وخرائطها مرتبطة ارتباطا وثيقا بكود الالة، بحيث ان أي مبرمج ماهر يستطيع كتابة كود code صغير وفعال. وكان ذلك مهم جدا عندما كانت ذاكرة الحاسب computer memory تقاس بالكيلو بايت، حتى أيضا مع أحجام الذاكرة الكبيرة اليوم؛ لغة C لا تزال مفيدة لكتابة الاكواد التي يجب أن تعمل بسرعة مثل نظام التشغيل.

وقد تم توسيع لغة C على مر السنين؛ والدليل على ذلك هو لغة C++ الذي تقوم بدعم الكائنات object support وإضافتها إلى لغة C وهو نمط مختلف من البرمجة، دليل اخر هو لغة Objective C والتي أخذت اتجاه آخر وتستخدم بشكل كبير في منتجات أبل Apple.

اللغة الثانية وهي أيضا مهمة هي لغة جافا Java والتي تأخذ ادوار مختلفة في نهج الترجمة. فبدلا من الترجمة إلى كود الآلة، فإن جافا Java تتخيل وجود معالج افتراضي hypothetical CPU وتسمية آلة جافا الافتراضية Java Virtual Machine (JVM). وبالتالي تتم ترجمة كل الكود لذلك المعالج. بعد ذلك يقوم كل حاسب مضيف host computer بتشغيل برنامج JVM ليقوم بترجمة تعليمات JVM وتسمى بايت كود bytecode إلى تعليمات رئيسية native instructions.

قد يظن البعض أن الترجمة الإضافية السابقة تجعل JAVA بطيء. ولكن في الحقيقة JVM بسيط بحيث يتم تنفيذه بشكل سريع وموثوق على أي آلة، ابتداء من جهاز حاسب قوي وانتهاء بجهاز منخفض الطاقة موصول بالتلفاز. الجدير بالذكر انه يمكن تشغيل ملف مترجم جافا compiled Java file على أي جهاز حاسب يحوي JVM.

فائدة أخرى من الترجمة compiling إلى مكان وسطي intermediate target هي أن JVM يمكن أن يوفر خدمات للتطبيقات التي عادة لا تكون متاحة على المعالج. إن حجز ذاكرة واسنادها إلى برنامج هي عملية معقدة، والحل هنا يكمن في استخدام JVM. وهذا يعني أيضا أن واضعي JVM ركزوا على تحسينات JVM ككل، لذلك أي تقدم يُصنع يصبح متوفر على الفور للتطبيقات.

من ناحية أخرى، لغة التفسير interpreted يتم تحويلها إلى كود الآلة للتنفيذ. الحاسب القوي جدا ينهك من هذا ولكن يمكن ان يعوضه من خلال زيادة الإنتاجية وتحقيق مكاسب من خلال عدم الحاجة للتوقف عن العمل للترجمة. تميل لغات التفسير Interpreted languages أيضا لتقديم المزيد من المزايا عن لغات الترجمة compiled languages، وهذا يعني أن تقليل الكود مطلب رئيسي. مترجم اللغة نفسها عادة ما يكون مكتوب بلغة أخرى مثل C وأحيانا جافا. وهذا يعني أن لغة التفسير يتم تشغيلها على آلة جافا الافتراضية JVM والتي تترجم إلى كود الآلة في وقت التشغيل.

ثالثاً لغة بيرل Perl هي لغة تفسير وُضعت في الاصل لمعالجة النص. على مر السنين اكتسبت هذه اللغة استحسان مسؤولي الانظمة لأنها في تحسن مستمر. تستخدم لغة بيرل في كل شيء من التشغيل الآلي إلى بناء تطبيقات الويب.

رابعاً PHP وهي لغة تم بناؤها في الأصل لإنشاء صفحات الويب الديناميكية. تتم قراءة ملف PHP من قبل خوادم الويب web servers مثل أبانتشي Apache. توجد في ملف PHP علامات خاصة؛ تشير إلى أن أجزاء من الكود code يجب أن تفسر على أنها تعليمات instructions. ليقوم بعد ذلك خادم الويب بجمع أجزاء الملف مع بعضها البعض وإرساله إلى متصفح الويب web browser. الميزة الرئيسية في لغة PHP هي أنها سهلة التعلم ومتوفرة تقريباً على أي نظام. وبسبب هذا بُنيت العديد من المشاريع بواسطة PHP. ومن الأمثلة البارزة مدونة ورد برس (WordPress (blogging وايضا cacti (monitoring) وأجزاء من موقع التواصل الاجتماعي الشهير الفيسبوك facebook.

خامساً روبي Ruby وهي لغة أخرى تأثرت بلغة بيرل Perl والقشرة Shell بجانب العديد من اللغات الأخرى. لكن هذه اللغة تمتاز بمقدرتها على تحويل مهام البرمجة المعقدة إلى سهلة نسبياً، ومع تضمين نافذة Rails framework؛ تصبح الاختيار الأمثل لبناء تطبيقات الويب المعقدة. روبي Ruby هي أيضاً اللغة المحركة لكثير من أدوات التشغيل الآلي الرائدة مثل Chef و Puppet، ميزة أخرى لهذه اللغة انها تسهل إدارة الاعداد الكبير من أنظمة لينكس.

سادساً بايثون Python وهي لغة برمجة شائعة الاستخدام أيضاً. مثل لغة روبي Ruby فإن بايثون تجعل المهام المعقدة أسهل، ولها نافذة تسمى Django والتي تجعل بناء تطبيقات الويب امراً سهلاً للغاية. بايثون Python لديها قدرات معالجة إحصائية ممتازة وهي المفضلة في الأوساط الأكاديمية.

خلاصة القول ان اللغة هي مجرد أداة تجعل من السهل أن نقول للحاسب ما نريد فعله. وتحتوي اللغات على مجموعة من المكتبات، ويمكن تلخيص دور المكتبات على انها تقوم بجمع المهام المشتركة library bundles common tasks في حزمة واحدة تستخدم من قبل المطور. ImageMagick هي إحدى هذه المكتبات، والتي تتيح للمبرمجين تعديل الصور في الكود code. وتأتي مع بعض أدوات سطر الأوامر command line tools والتي تُمكنك من معالجة الصور من القشرة shell والاستفادة من قدرات البرمجة هناك.

OpenSSL هي مكتبة تشفير التي يتم استخدامها في كل شيء من خوادم الشبكة إلى سطر الأوامر. وهي توفر

واجهة موحدة بحيث يمكنك إضافة الترميز cryptography في لغة بيرل Perl.

وفي المستويات الأخيرة نجد مكتبة C. وهي توفر مجموعة أساسية من الوظائف مثل العرض والقراءة والكتابة من وعلى الملفات والتي تستخدم من قبل التطبيقات واللغات الأخرى على حد سواء.

## ٢,٣ فهم البرمجيات المفتوحة المصدر والتراخيص

عندما نتحدث عن شراء البرمجيات هناك ثلاثة عناصر متميزة:

❖ الملكية Ownership: من يملك الملكية الفكرية وراء البرنامج؟

❖ تحويل الأموال Money transfer: الكيفية؟

❖ الترخيص Licensing: على ماذا ستحصل؟ ماذا يمكنك أن تفعل بالبرنامج؟ يمكنك استخدامه على

حاسب واحد فقط أو يمكنك إعطائه لشخص آخر؟

في معظم الحالات، ملكية البرنامج تبقى مع الشخص أو الشركة التي صنعتها، أما المستخدمون فيتم منحهم ترخيص لاستخدام البرنامج وهذا حق مكفول في قانون حق الملكية. اما النقطة الثانية وهي تحويل الأموال فهي تعتمد على نموذج العمل للصانع. وأخيرا تراخيص البرنامج وهي التي تميز حقا بين البرمجيات مفتوحة المصدر والبرمجيات مغلقة المصدر.

هنا مثالين مختلفين!

شركة مايكروسوفت ويندوز Microsoft Windows تمتلك الملكية الفكرية. اتفاقية ترخيص المستخدم End User License Agreement – (AULE) هي وثيقة قانونية خاصة يجب عليك الموافقة عليها بالنقر عليها عند التثبيت install للإشارة إلى فهمك وموافقتك. مايكروسوفت Microsoft تحافظ على كود المصدر source code وتقوم بتوزيع نسخ ثنائية binary copies من خلال القنوات المصرح بها authorized channels. بالنسبة لمعظم المنتجات الاستهلاكية consumer products فإنه يسمح لك بتثبيت البرنامج على جهاز حاسب واحد فقط؛ وأيضا لا يُسمح بعمل نسخ أخرى غير النسخة الاحتياطية. وأخيرا فلا يسمح لك

بتغيير هندسة البرنامج. لذلك فأنت تدفع لنسخة واحدة من البرنامج فقط ولن تحصل الا على تحديثات بسيطة له من غير أي ترقية كاملة.

في الجهة الأخرى، نجد ان لينكس Linux مملوكة من قبل لينوس تورفالدس Linus Torvalds والذي وضع الكود الذي قام ببرمجته تحت ما يسمى برخصة جنو العمومية الإصدار الثاني GNU Public License (GPLv2) - version 2. يعني هذا الترخيص إن كود المصدر source code يجب أن يكون متاحا لأي شخص، كما يسمح له بإجراء أي تغييرات أو تعديلات يريد ادخالها عليه. التحذير الوحيد أنه إذا قمت بعمل تغييرات ثم قمت بتوزيعها فإنه يجب عليك وضعها تحت نفس الترخيص حتى يتسنى للآخرين الاستفادة منها. تنص رخصة GPLv2 أيضا على أنه لا يُسمح لك بأخذ تكاليف لتوزيع كود المصدر source code أكثر من تكاليفك الفعلية للقيام بذلك (مثل نسخها إلى الوسائط القابلة للإزالة).

بشكل عام عند إنشاء create عمل ما يمكنك الحصول على الحق في تقرير كيفية استخدام ذلك العمل وتوزيعه used and distributed كيف تشاء. البرمجيات الحرة والمفتوحة المصدر Free and Open Source Software - (FOSS) تشير إلى التخلي given up عن هذا الحق والسماح بعرض كود الرئيسي للبرنامج وهو كود المصدر source code وأيضا القيام بعملية إعادة التوزيع redistribute لذلك العمل. وهذا ما فعله لينوس تورفالدس Linus Torvalds مع لينكس، على الرغم من أنه أنشأ created لينكس الا انه يمكن لأي كان استخدامه بلا مقابل، وذلك بسبب تخليه عن حقه فيه right given up من خلال وضعه تحت ترخيص GPLv2.

تراخيص البرامج هي قضية سياسية political issue، وينبغي ألا تكون متفاجئا بوجود العديد من الآراء المختلفة حولها. وهو ما سبب ظهور منظمات قامت بطرح رخص خاصة بها والتي تتضمن عرض لوجهات نظرها، ومع وجود ذلك العدد من التراخيص، أصبح اختيار ترخيص موجود لبرنامج ما أسهل من انشاء ترخيص جديد. على سبيل المثال، أوجدت الجامعات مثل معهد ماساتشوستس للتكنولوجيا (MIT)، وجامعة كاليفورنيا University of California تراخيص لمشاريعهم مثل مؤسسة أباتشي Apache Foundation. الجدير بالذكر أن مجموعات مثل مؤسسة البرمجيات الحرة Free Software Foundation قد صنعت تراخيص خاصة بها لتعزيز أجندتها أيضا.

## ٢,٣,١ مؤسسة البرمجيات الحرة والمصدر المفتوح

في عالم المصادر المفتوحة هناك مجموعتين يمكن اعتبارها القوى والأكثر تأثيراً، الأولى هي مؤسسة البرمجيات الحرة (FSF) – Free Software Foundation والثانية (OSI) – Open Source Initiative وهي مبادرة المصدر المفتوح.

تأسست مؤسسة البرمجيات الحرة عام ١٩٨٥، وذلك من قبل ريتشارد ستالمان Richard Stallman (RMS). هدف هذه المؤسسة هو تعزيز البرمجيات الحرة Free Software. البرمجيات الحرة لا تشير إلى سعر البرنامج بل إلى الحرية في تبادل ودراسة وتعديل التعليمات البرمجية للكود المصدر underlying source code. ومن وجهة نظرها (FSF) أن البرمجيات الاحتكارية proprietary software (التي توزع بموجب رخصة مغلقة المصدر) سيئة. كما تطالب مؤسسة (FSF) أن تكون البرمجيات المرخصة قابلة للتعديل modifications. وبعد ذلك تأتي مشاركة هذه التعديلات أو التغييرات وهو ما يسمى بفلسفة النسخة المتروكة copyleft.

تعارض مؤسسة البرمجيات الحرة (FSF) براءة اختراع البرمجيات software patents، وتعمل كمراقب عام للمنظمات المعيارية standards organizations، وتقوم بنشر أي انتهاك لمعايير البرمجيات الحرة بواسطة مواد مثل إدارة الحقوق الرقمية Digital Rights Management – (DRM) التي يمكن أن تقيد ما تقدمه من خدمات.

وضعت (FSF) مجموعة من التراخيص الخاصة بها مثل GPLv2 و GPLv3 وترخيص GPL الصغير Lesser GPL licenses الإصدارات ٢ و ٣ (LGPLv2 و LGPLv3). الترخيص الصغير lesser licenses مثل الترخيص العادي إلا أنه يتضمن أحكاماً ضد البرامج غير المجانية non-Free Software. على سبيل المثال، تحت مظلة الترخيص GPLv2 فإنه لا يمكنك إعادة توزيع البرمجيات التي تستخدم مكتبة ذات مصدر مغلق مثل برنامج تشغيل الأجهزة non-Free Software ولكن الترخيص الصغير يسمح بذلك.

التغييرات بين الإصدار ٢ و ٣ تتمحور- إلى حد كبير حول استخدام البرمجيات الحرة في جهاز مغلق closed hardware device والذي صيغ باسم Tivoization. تيفو TiVo هي الشركة التي بنت مسجل الفيديو

الرقمي التلفزيوني television digital video recorder على أجهزتهم واستخدموا لينكس كأساس لبرامجهم. بموجب ترخيص GPLv2 نشرت تيفو TiVo كود المصدر source code لنسخة من لينكس، ولكن أجهزتهم لم تقم بتشغيل أي تعديل ثنائي binaries. وهذا بنظر مؤسسة البرمجيات الحرة (FSF) يتعارض مع روح GPLv2 لذلك أضافوا بندا محددًا إلى الإصدار ٣ من الرخصة. وتوصل لينوس تورفالدس Linus Torvalds إلى حل مع تيفو بشأن هذه المسألة واختارت البقاء مع ترخيص GPLv2.

والآن لنتحدث عن المجموعة الثانية في عالم المصادر المفتوحة وهي مبادرة المصدر المفتوح، تأسست مبادرة المصدر المفتوح Open Source Initiative في عام ١٩٩٨ من قبل بروس بيرينز Bruce Perens وإريك ريموند Eric Raymond (ESR). وهم يعتقدون أن البرمجيات الحرة Free Software هي مشحونة جدا بالسياسة، وأنها أقل صرامة مما يجب أن تكون، وتحديدًا حول نقطة الحقوق المتروكة copyleft في تراخيص البرمجيات الحرة (FSF). ويرى أعضاء (OSI) انه ينبغي أن يكون المصدر source متاح بحرية، وانه يجب الغاء أي قيود موضوعة على استخدام البرمجيات بغض النظر عن الاستخدام المقصود. وبعكس (FSF) فان (OSI) لا يوجد لديها مجموعة تراخيص مختلفة، ولكنها عوضًا عن ذلك تحتوي على قائمة بها مجموعة من المبادئ واضيفت لها تراخيص أخرى توافقت مع مبادئهم، وسميت بتراخيص المصدر المفتوح Open Source licenses. لذا البرامج التي تتفق مع رخص المصدر المفتوح Open Source license هي برمجيات مصدر مفتوح Open Source Software.

بعض رخص المصدر المفتوح تأتي من تراخيص عائلة BSD، وهي أبسط بكثير من تراخيص GPL. حيث أنها تسمح بإعادة توزيع المصدر والنظام الثنائي source and binaries بشروطين، الأول هو أن تبقى إشعارات حقوق النشر copyright. والثاني ألا تُلحَ أن منشئ الإصدار الأصلي في النسخة التي عدلتها. وبعبارة أخرى اعمل ما تريد في هذا البرنامج، فقط لا تقول أنك كتبتّه. وتتفق رخص معهد ماساتشوستس للتكنولوجيا MIT مع هذا الفكر ولكنه مكتوب لديهم بصيغة أخرى.

تراخيص مؤسسة البرمجيات الحرة FSF مثل GPLv2 هي تراخيص مفتوحة المصدر. ومع ذلك، فإن العديد من التراخيص المفتوحة المصدر مثل BSD ومعهد ماساتشوستس للتكنولوجيا MIT لا تحتوي على أحكام الحقوق المتروكة copyleft، وبالتالي فهي مسألة غير مقبولة لمؤسسة البرمجيات الحرة FSF. ويطلق على هذه

الرخص المتساهلة permissive تراخيص للبرمجيات الحرة free software licenses وذلك لأنها متساهلة تجاه كيفية إعادة توزيع البرمجيات. لذلك تستطيع أخذ البرامج المرخصة من BSD وإدراجها في منتجات البرمجيات المغلقة طالما أنك اعطيت الإسناد الصحيح.

٢,٣,٢ مزيداً من الشروط لنفس الشيء

بدلاً من التدقيق على التفاصيل الصغيرة بين المصدر المفتوح OSI والبرمجيات الحرة FSF، بدأ المجتمع بالإشارة إلى كل ذلك بمسمى برمجيات المصدر الحر والمفتوح Free and Open Source software (FOSS). الكلمة الإنجليزية "مجانا" free والموجودة في المسمى FOSS يمكن أن تكون مربكة، حيث أنها تعني مجانا بلا مقابل كما هو الحال في وجبة غداء" (بدون أي تكلفة) أو تعني حر كما في الخطاب (دون أي قيود). وقد أدى هذا الغموض إلى إدراج كلمة ليبر libre للإشارة إلى التعريف الأخير. وبالتالي، وفي نهاية المطاف أصبح الاسم كما يلي البرمجيات المجانية / الحرة المفتوحة المصدر Free/Libre/Open Source Software - (FLOSS).

رغم أن هذه الشروط تبدو متألفة، إلا أنها تخفي الكثير من اختلافات الفكر بين المدرستين السابقتين. وخلاصة القول؛ عند استخدام برامج FLOSS، فأنتك غير ملزم بدفع ثمنها، كما ويمكنك إعادة توزيع تلك البرامج دون أي قيود.

٢,٣,٣ أنظمة التراخيص الأخرى

تراخيص FOSS تتعلق معظمها بالبرامج. وقد وضع الناس أعمال مثل الرسم والخطط ضمن تراخيص FOSS ولكن لم يكن هذا هو القصد.

عند وضع برنامج ما في المجال العام، فهذا يعني أن كاتب البرنامج قد تخلى عن جميع الحقوق، بما في ذلك حقوق الملكية copyright. ويعتبر هذا في بعض البلدان هو التصرف الطبيعي عندما يتم العمل من قبل وكالة حكومية. وفي بلدان أخرى تتحول حقوق الطبع والنشر والملكية إلى حق عام في حالة وفاة المؤلف وانقضاء فترة انتظار طويلة.

أنشأت منظمة المشاع الإبداعي (CC) Creative Commons - تراخيص المشاع الإبداعي Creative Commons Licenses التي تحاول معالجة النوايا وراء رخص البرمجيات الحرة المفتوحة المصدر FOSS للكيانات الغير برمجية non software entities. ويمكن أيضا ان تستخدم تراخيص CC لتقييد الاستخدام التجاري إذا كان هذا هو رغبة صاحب حقوق الطبع والنشر copyright holder. ويمكن تلخيص تراخيص CC كالتالي:

- ❖ **إسناد (CC-BY) Attribution -** مثل الكثير من رخص BSD يخولك CC-BY من استخدام اي محتوى مع ذكر صاحب حقوق الطبع والنشر.
- ❖ **إسناد المشاركة بالمثل (CC BY-SA) Attribution ShareAlike :** وهي نسخة متروكة الحقوق copyleft ولكن في ترخيص الإسناد. وهي تنص على أن العمل المشتق يجب أن يكون مشترك تحت نفس الترخيص وذلك يشبه إلى حد كبير افكار البرمجيات الحرة Free Software ideals.
- ❖ **إسناد لا- للاشتقاق (CC BY-ND) Attribution No-Derivs :** ويعني امكانية إعادة توزيع المحتوى تحت نفس شروط CC-BY ولكن بدون أي تغيير.
- ❖ **إسناد غير تجاري (CC BY-NC) Attribution-NonCommercial :** وهو مشابه تماما لإسناد CC BY، ولكن لا يحق فيه استخدام المحتوى لأي اغراض تجارية.
- ❖ **إسناد غير تجاري للوثائق الحرة (CC-BY-NC-SA) Attribution-NonCommercial ShareAlike -** وهو مبني على ترخيص CC BY-NC ولكن يستلزم أن تكون التغييرات متاحة للمشاركة تحت نفس الترخيص.
- ❖ **إسناد - غير تجاري - غير مشتق (CC-BY-NC-ND) Attribution-NonCommercial No-Derivs -** ويعني أنه يمكن مشاركة المحتوى ليتم استخدامه لأغراض غير تجارية، ولكن بدون تغيير المحتوى.
- ❖ **لا للحقوق المحفوظة (CC0) No Rights Reserved :** هذه هي نسخة ترخيص المشاع الإبداعي بالمجال العام.

الرخص أعلاه يمكن تلخيصها على النحو التالي بالمثل ShareAlike أو بدون قيود no restrictions، وهل يسمح فيها أو لا بالاستخدام التجاري أو الاشتاقات.

### ٢,٣,٤ النماذج التجارية للمصدر المفتوح

إذا كنت تعطي البرنامج مجاناً فكيف يمكنك كسب المال؟

إن أبسط طريقة لكسب المال هو عن طريق بيع الدعم support أو الضمان warranty للبرمجيات. ويمكن أيضاً كسب المال عن طريق تثبيت installing البرنامج للآخرين، والمساعدة في حل المشاكل problems، أو تحديد الخلل fixing bugs مقابل المال فتصبح استشاري consultant فعال.

كما يمكنك أيضاً فرض رسوم على خدمة service أو الاشتراك subscription بالبرنامج. مشروع مسجل الفيديو الرقمي ذو المصدر المفتوح Open Source MythTV digital video recorder project هو مثال ممتاز على ذلك. البرنامج نفسه مجاني؛ ولكن يتم الدفع لتوصيله بقائمة جهاز التلفاز لمعرفة أوقات البرامج المعروضة.

من وسائل كسب المال أيضاً إعطاء مجموعة أجهزة package hardware أو إضافة برامج مغلقة المصدر closed source لبيعها جنباً إلى جنب مع البرمجيات الحرة Free software. الأجهزة والأنظمة المدمجة Appliances and embedded systems التي تستخدم نظام التشغيل لينكس يمكن تطويرها وبيعها أيضاً. العديد من جدران الحماية firewalls الاستهلاكية وأجهزة الترفيه entertainment devices تتبع هذا النموذج.

يمكنك أيضاً تطوير develop برمجيات المصدر المفتوح open source software كجزء من وظيفتك. إذا قمت بإنشاء أداة لتسهيل عملك فستكون قادر على إقناع صاحب العمل بجعلها ذات مصدر مفتوح. أيضاً يمكن ان تعمل بوظيفة بمقابل مالي ولكنك تعمل على برنامج ذو ترخيص مفتوح المصدر، فعملية نشره للمستخدمين تسمح بالمساعدة في حل نفس مشكلة.

في عام ١٩٩٠، كان جيرالد كومز Gerald Combs يعمل على موفر خدمة الانترنت Internet service provider – (ISP) وبدأ بكتابة أداة تحليل الشبكة network analysis tool وذلك لأن الأدوات المماثلة في ذلك الوقت كانت مكلفة للغاية. وقد ساهم أكثر من ٦٠٠ شخص إلى الآن في المشروع المسمى Wireshark. والآن يعتبر ذلك المشروع أفضل حتى من العروض التجارية الموجودة، وهو ما أدى إلى قيام شركة تشكلت حول جيرالد لدعم Wireshark. وعند بيع المنتجات والدعم أصبح المشروع أكثر فائدة. واخيراً تم شراء هذه الشركة من قبل منتج شبكة معروف لي دعم تطور البرنامج.

شركات أخرى حصلت على قيمة هائلة من برامج المصدر المفتوح، حيث انهم وجدوا أنها تستحق توظيف أشخاص للعمل على عليها بدوام كامل. وكمثال على ذلك؛ فقد قامت شركة جوجل Google صاحبة محرك البحث search engine بتوظيف كاتب لغة البرمجة بايثون Python، حتى لينوس تورفالدس نفسه Linus Torvalds تم توظيفه من قبل مؤسسة لينكس Linux Foundation للعمل على مشروع لينكس. شركة الهاتف الأمريكية AT & T حصلت على قيمة مماثلة من خلال مشروع Ruby و Rails والخاص بدليل الهاتف الاصفر الخاص بهم Yellow Pages وقد وظفوا مجموعة من الموظفين فقط للعمل عليها.

طريقة نهائية لجني المال بشكل غير مباشر من خلال المصادر المفتوحة وهي أنها وسيلة ممتازة للحكم على المهارات. فهناك فرق بين ان تقول عن مميزات ادائك وبين ان تنشر عملك وتشاركه ليسمح للعالم بمعرفة نوعيته بمن فيهم رؤساء عمل محتملين. وبالمثل وجدت الشركات التي تفتح الأجزاء الغير الهامة من برامجها الداخلية انها تجذب اهتمام الجميع.

## الفصل الثالث – استخدام لينكس

# Using Linux

### ٣,١ مقدمة

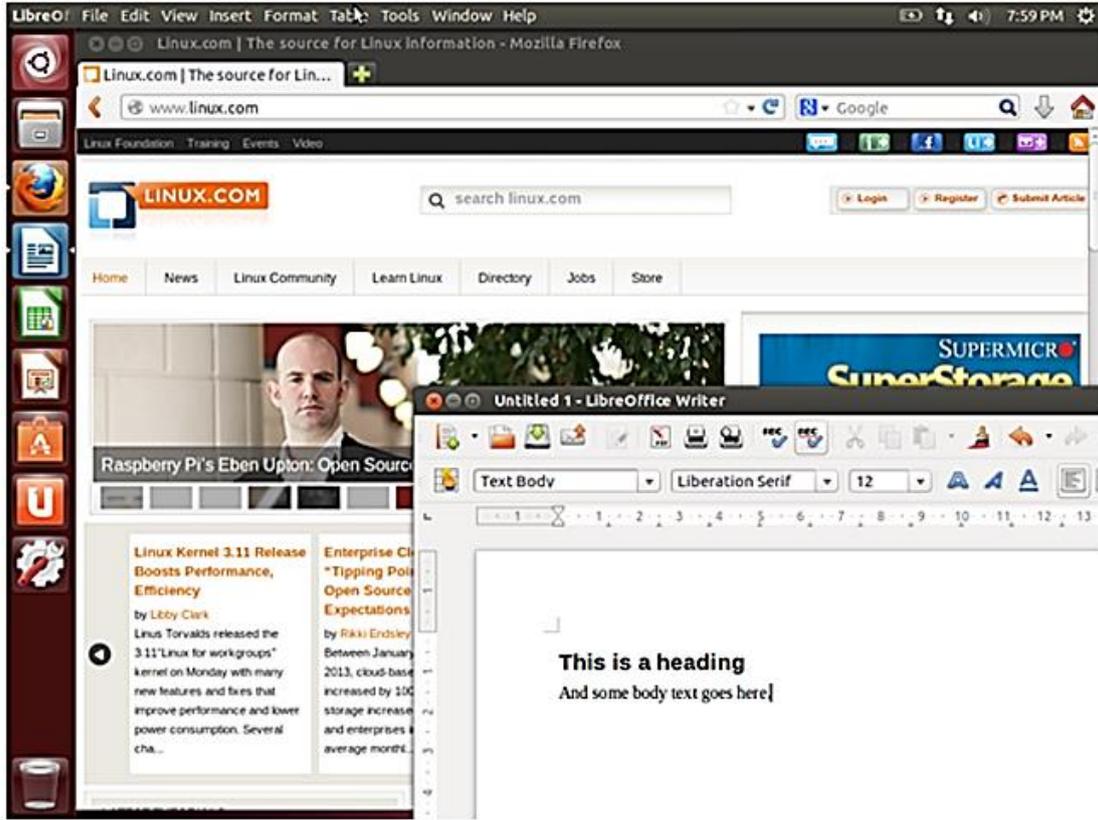
قبل أن تتمكن من أن تصبح مسؤول فعال لأنظمة لينكس effective Linux systems administrator، يجب أن تكون قادرا على استخدام جهاز لينكس المكتبي desktop، وان تكون محترف مع المعلومات الأساسية basic Information ومهارات تكنولوجيا الاتصالات (ICT) - Communication Technology، لأنها سوف تساعدك عند التعامل مع المستخدمين users، ومما لا شك فيه أن التعمق في استخدام لينكس سوف يساعدك على تحسين مهاراتك بسرعة أكبر. وعلاوة على ذلك، عمل مدير نظم systems administrator لا تقتصر على الإلمام بأعمال الخادم server work، بل يمتد ليشمل العمل على البريد الإلكتروني email والتوثيق documentation !

اعلى منصب للتوظيف في لينكس من قبل مدراء تقنية المعلومات IT هو مسؤول أنظمة systems administrators.

### ٣,٢ الوضع الرسومي مقابل غير الرسومي

نظام التشغيل لينكس Linux يمكن استخدامه بأحد طريقتين؛ الأولى هي الواجهة الرسومية graphically والثانية هي غير الرسومية non- graphically. تكون التطبيقات في الواجهة الرسومية على شكل نوافذ windows قابلة للتحريك move ولتغيير الحجم resize. القوائم menus والأدوات tools هي علامة فارقة في الواجهة الرسومية graphically mode حيث أنها تساعدك في العثور على ما تريد البحث عنه. كذلك ستجد متصفح ويب web browser وأدوات تحرير الرسومات graphics editing tools والبريد الإلكتروني email الخاص بك.

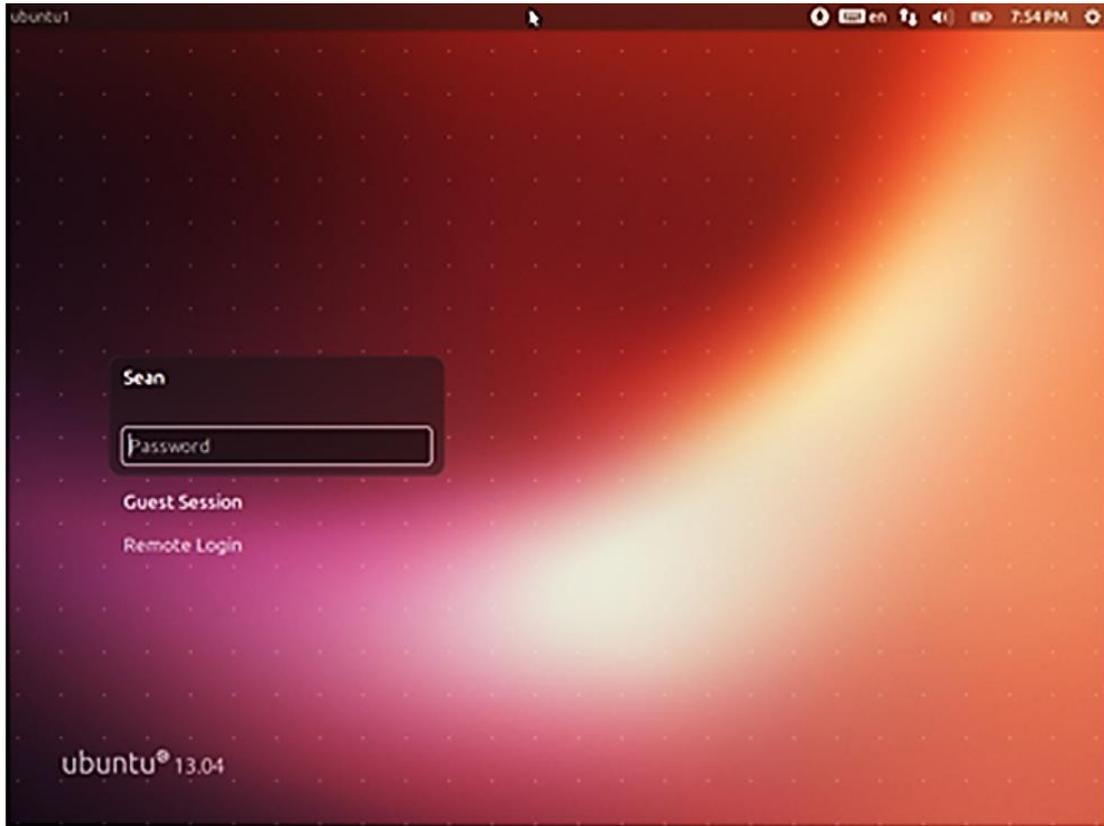
الشكل القادم هو مثال على جهاز سطح مكتب رسومي graphical desktop لأحد أنظمة التشغيل في لينكس، ونرى في يسار الشاشة شريط القوائم menu bar؛ وبه مجموعة من التطبيقات المشهورة popular applications، أما في الأسفل فنرى برنامج المستندات document من باقة LibreOffice؛ وكما نلاحظ يتم التعديل بالمستند، أخيرا وفي الخلفية نجد أحد مستعرضي الويب.



### الواجهة الرسومية لعدد من التطبيقات في لينكس

في الواجهة الرسومية graphical mode؛ يمكن أن نقوم بفتح عدة واجهات بنفس الوقت؛ وهو أمر مفيد للغاية عند أداء المهام tasks على أجهزة الحاسب البعيدة المتعددة multiple remote computers. حتى أنك تستطيع ان تقوم بتسجيل الدخول log in باستخدام اسم المستخدم username وكلمة المرور الخاصة بك password من خلال واجهة رسومية graphical interface.

الشكل أدناه هو مثال على شاشة تسجيل الدخول graphical login في البيئة الرسومية graphical interface.



شاشة تسجيل الدخول باستخدام اسم المستخدم وكلمة المرور

بعد تسجيل الدخول logging in، ستنتقل إلى مكتبي desktop حيث يمكنك تحميل التطبيقات load applications.

الواجهة غير الرسومية non-graphical mode تقوم فيها بتسجيل الدخول القائم على النص text-based login، كما هو موضح في الشكل أدناه. يُطلب منك اسم المستخدم الخاص بك username وبعد ذلك كلمة المرور password. إذا تمت عملية تسجيل الدخول login بشكل ناجح successful، يتم نقلك مباشرة إلى القشرة Shell.

```
Ubuntu 13.04 ubuntu1 tty2
ubuntu1 login:
```

### واجهة تسجيل الدخول غير الرسومية في لينكس

في الوضع الغير رسومي non-graphical mode لا توجد نوافذ windows للتحرك. على الرغم من وجود برامج تحرير النصوص text editors ومتصفحات الويب web browsers وعملاء البريد الإلكتروني email clients الا أنهم جميعا عبارة عن نصوص فقط text. وكانت هذه هي الطريقة التي بدأ بها يونيكس UNIX قبل وجود الواجهات الرسومية graphical environments. معظم الخوادم servers تعمل في هذا الوضع أيضا، لأن المستخدمين لا يقومون بتسجيل الدخول فيها بشكل مباشر، مما يجعل الواجهة الرسومية graphical interface مضيعة للموارد resources waste. المثال التالي هو لما يمكن أن تراه بعد أن تقوم بعملية تسجيل الدخول logging in.

```

Ubuntu 13.04 ubuntu1 tty2

ubuntu1 login: sean
Password:

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

Welcome to Ubuntu 13.04 (GNU/Linux 3.8.0-19-generic x86_64)

 * Documentation:  https://help.ubuntu.com/

212 packages can be updated.
91 updates are security updates.

sean@ubuntu1:~$ w
 20:08:35 up 14 min,  2 users,  load average: 0.45, 0.44, 0.32
USER      TTY      FROM            LOGIN@   IDLE   JCPU   PCPU WHAT
sean     tty2                20:08   14:35   0.05s  0.00s w
sean     tty7                19:54   14:35   1:08   0.16s gnome-session -
sean@ubuntu1:~$ _

```

### مثال على الشاشة بعد تسجيل الدخول

في الشاشة السابقة نرى موجة الدخول الأصلي original prompt to login في الجزء العلوي من النص. أثناء عملية تسجيل الدخول login، ربما تشاهد بعض الرسائل message، والتي تسمى رسالة اليوم (MOTD)، وهي وسيلة مناسبة لمدير النظام systems administrator للقيام بعملية تمرير المعلومات pass information إلى المستخدمين users. بعد MOTD ستجد موجة الأوامر command prompt. نلاحظ في الشاشة أعلاه، أن المستخدم ادخل الأمر w في موجة الأوامر prompt وهو يوضح حالات تسجيل الدخول على النظام. أيضا نرى أنه تم إدخال الأوامر الجديدة ومعالجتها وتظهر على النافذة الطرفية terminal، أما الأوامر الأقدم فتكون في أعلى الجلسة إلى أن تختفي. النافذة الطرفية terminal هي المسؤولة عن حفظ كل ما يظهر فيها history، مثل السماح للمستخدم بتحريك المؤشر لأعلى scroll up ورؤية الأوامر commands التي سبق إدخالها. بالنسبة للينكس ما يظهر في الشاشة هو الموجود بالنظام.

### ٣,٣ سطر الأوامر

سطر الأوامر command line هو إدخال نصي بسيط simple text input، يتيح لك الإدخال كما تريد، من الأوامر ككلمة واحدة إلى البرامج المعقدة complicated scripts. إذا قمت بتسجيل الدخول من خلال وضع النص text-mode (الواجهة الغير الرسومية) فانت في وحدة التحكم console. اما إذا قمت بتسجيل الدخول في الواجهة الرسومية graphical mode، فستحتاج للدخول إلى برنامج القشرة Shell والتي هي وحدة تحكم نصية text console مع نافذة window يمكن تحريكها move it around أو تغيير حجمها .resize.

كل جهاز مكتبي في لينكس Linux desktop مختلف عن نظرائه، لذلك يجب أن تبحث في القوائم menus الخاصة بك عن خيار نافذة طرفية terminal أو x-term. البرنامجين السابقين يمثلان القشرة ولكن بواجهة رسومية shells graphical، والاختلاف بينهما في الغالب هو في الشكل appearance لا في العمل function. إذا كان لديك أداة للبحث فيمكنك البحث عن كلمة terminal كما هو موجود في نظام التشغيل أوبونتو Ubuntu.



#### خاصية البحث

هذه الأدوات tools تتيح لك البحث search بسرعة عما تريده في نظام التشغيل، بدلا من البحث من خلال القوائم menus.

## ٣, ٤ المحاكاة الافتراضية والحوسبة السحابية

لينكس هو نظام تشغيل متعدد المستخدمين multiuser operating system، مما يعني أنه يمكن للعديد من المستخدمين users العمل على النظام بشكل متزامن simultaneously. في الغالب؛ المستخدم العادي لا يمكنه الحاق الضرر بالمستخدمين الآخرين. ومع ذلك، هناك بعض القيود limitations حيث أن بعض المستخدمين يقومون بالإستلاء على مساحة القرص disk أو إستخدام الكثير من الذاكرة memory أو إستهلاك موارد المعالج CPU resources مما يجعل النظام بطيء للجميع. أيضا كل مستخدم في وضع تعدد المستخدمين multiuser mode يمتلك فقط الصلاحيات الاساسية للمستخدم العادي unprivileged users، مما يجعل تشغيل خادم الويب web server الخاص بكل مستخدم صعب جدا.

الافتراضية تعمل لأن الخوادم servers تكون معظم الوقت في وضع الخمول idling وذلك لأنها لا تحتاج إلى موارد مادية physical resources مثل الشاشة monitor أو لوحة المفاتيح keyboard لذلك وباستخدام تلك التقنية فأنا نستطيع أخذ معالج قوي powerful CPU ونشره حول بين أجهزة ظاهرية متعددة multiple virtual machines مع الحفاظ على القسمة العادلة equitable sharing بين الضيوف في نظام لينكس. ولكن وفي هذه الطريقة يوجد قيد limitation وعادة هو الذاكرة memory. عدا ذلك ومع التقدم التكنولوجي لبرامج hypervisor وللمعالجات CPUs أيضا، أصبح من الممكن وضع آلات ظاهرية virtual machines على مضيف واحد host أكثر من أو وقت مضى.

في البيئة الافتراضية، يوجد حاسب فعلي physical واحد يسمى المضيف host، يستطيع تشغيل العشرات من الأجهزة، حيث يقوم بتشغيل عدة نسخ من نظام التشغيل لعدد من الأجهزة ويسمون ضيوف guests. يقوم جهاز host بتشغيل برنامج يسمى hypervisor، وظيفة ذلك البرنامج هي تبديل التحكم switche control بين مختلف الضيوف، مثلما يحدث في نواة لينكس عند العمل في العمليات الفردية.

وبدعم من المعالج نفسه، لن يعرف الضيوف أنهم على جهاز ظاهري. حيث يحصل كل ضيف على معالج افتراضي وذاكرة وصول عشوائي RAM، وقرص صلب HD، ويتواصلون في شبكتهم الخاصة. بل إنه ليس

من الضروري تشغيل نفس نظام التشغيل على جميع اجهزة الضيوف، مما يقلل كذلك عدد من الخوادم المادية اللازمة physical servers.

تقدم الافتراضية وسيلة لخفض استهلاك الطاقة lower power usage وتقليل استهلاك مراكز البيانات على reduce datacenter space الخوادم الفعلية physical servers. الضيوف هم الآن مجرد برامج معدة software configurations، لذلك فمن السهل اختبارها testing وإنهائها destroy عندما تنتهي الفائدة منها.

من الممكن تشغيل نظام تشغيل لحظي متعدد multiple instance على جهاز فعلي physical machine واحد وتوصيله connect عبر الشبكة network، عندها يكون موقع location الجهاز غير ذات أهمية. الحوسبة السحابية Cloud computing تأخذ هذا النهج، وتتيح لك الحصول على جهاز ظاهري virtual machine في مراكز البيانات البعيد remote datacenter التي لا تملكها، وتدفع فقط للموارد resources التي تستخدمها.

مزودي vendors خدمة الحوسبة السحابية Cloud computing يمكنهم الاستفادة اقتصاديا economy بتقديم مواردهم computing resources بأسعار أفضل من تكلفة cost شراء تلك الأجهزة كأجهزة الفضاء. الخوادم الظاهرية virtual servers هي الوجه الاخر للحوسبة السحابية. حيث يمكنك الحصول على تخزين الملفات file storage وعلى قواعد البيانات databases، أو حتى على مجموعة من البرمجيات software. المفتاح في معظم هذه المنتجات products هو أن تدفع مقابل ما تستخدم، مثل كمية معينة لكل جيجابايت من البيانات شهريا، بدلا من شراء الأجهزة والبرمجيات.

بعض الحالات some situations أكثر ملاءمة للسحابة cloud من غيرها. المخاوف الأمنية security والأداء performance عادة ما تكون العناصر الأولى في عملية الاختيار، ثم تليها التكاليف costs وفعالية الوظائف functionality.

يلعب لينكس دورا محوريا pivotal role في الحوسبة السحابية cloud computing. تعمل معظم الخوادم الظاهرية virtual servers على أحد انوية لينكس Linux Kernel وغالبا ما تستخدم لينكس لاستضافة host التطبيقات applications وراء خدمات الحوسبة السحابية cloud computing services.

### ٣,٥ استخدام لينكس للعمل:

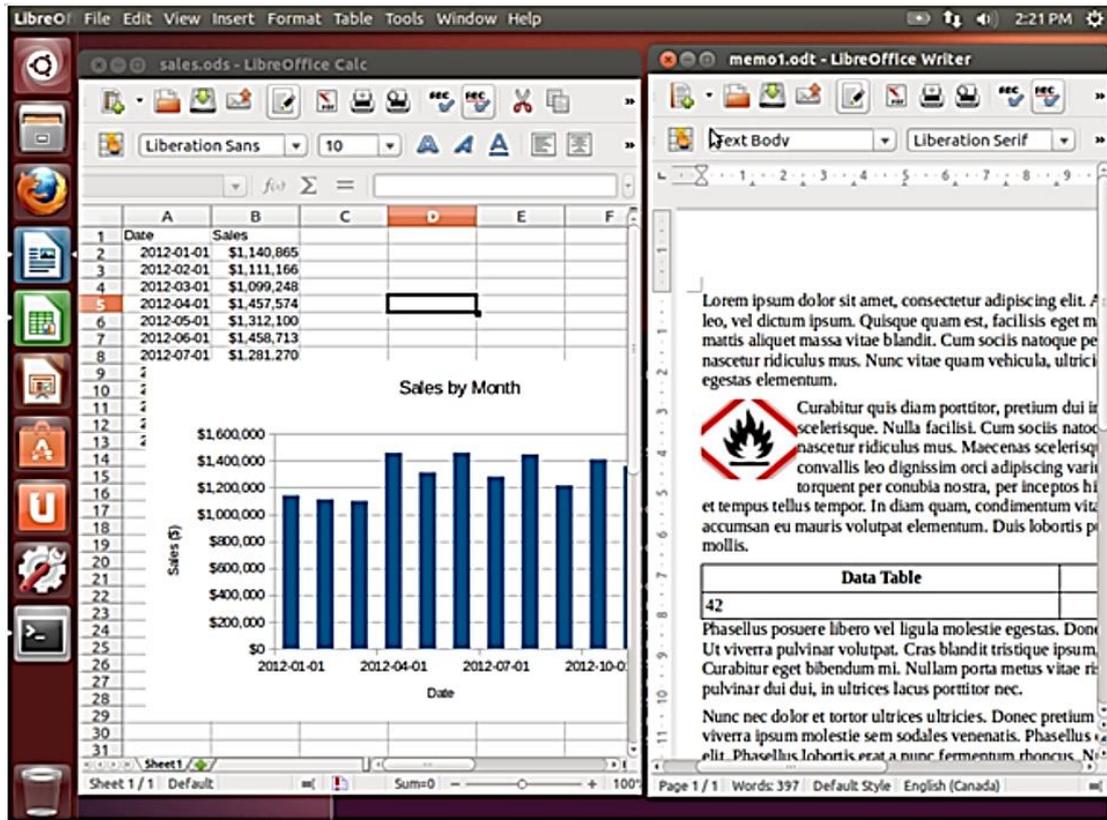
الأدوات الأساسية basic tools المستخدمة في معظم المكاتب offices هي:

- معالج النصوص Word processor
- الجداول الإلكترونية Spreadsheet
- حزمة العرض Presentation package
- متصفح الويب Web browser

برنامج OpenOffice وبرنامج LibreOffice يهتمان بالخدمات السابقة. حيث يستخدم معالج النصوص word processor لتحرير الوثائق، مثل التقارير والمذكرات. أما الجداول الإلكترونية Spreadsheets فهي مفيدة للعمل مع الأرقام كتلخيص المبيعات وعمل توقعات مستقبلية. أما حزمة العرض presentation فتستخدم لإنشاء الشرائح slides مع إضافة بعض من المكونات مثل النصوص text والرسومات graphics والفيديوهات المدمجة embedded video. الشرائح قد تكون مطبوعة printed أو معروضة displayed على شاشة screen أو على جهاز الإسقاط projector لمشاركتها مع الجمهور.

مبين أدناه برنامجي الجداول الإلكترونية spreadsheet ومحرر نصوص document editor التابعين لمجموعة LibreOffice. لاحظ أن spreadsheet ليس مقيد بعدد محدد من الصفوف rows والأعمدة columns. كما يمكن للأرقام أن تكون مصدرا source للرسم البياني graph، ونرى أن الصيغ formulas يمكن كتابتها لحساب القيم calculate values على ضوء المعلومات information المعطاة، مثل حساب نسبة الفائدة ومبالغ القروض معا وذلك للمساعدة في مقارنة جميع الخيارات.

باستخدام محرر LibreOffice يمكن أن يحتوي المستند document على نص text ورسومات graphics وجداول البيانات data tables وأكثر من ذلك. كما ويمكنك ربط المستندات documents وجداول البيانات spreadsheets معا، على سبيل المثال يمكنك القيام بعملية تلخيص summarize البيانات في شكل مكتوب written form، وبشكل تلقائي فإن أي تغييرات تتم على ذلك جدول spreadsheet ستظهر في المسند document المرتبط به.



مثالين من برامج حزمة LibreOffice

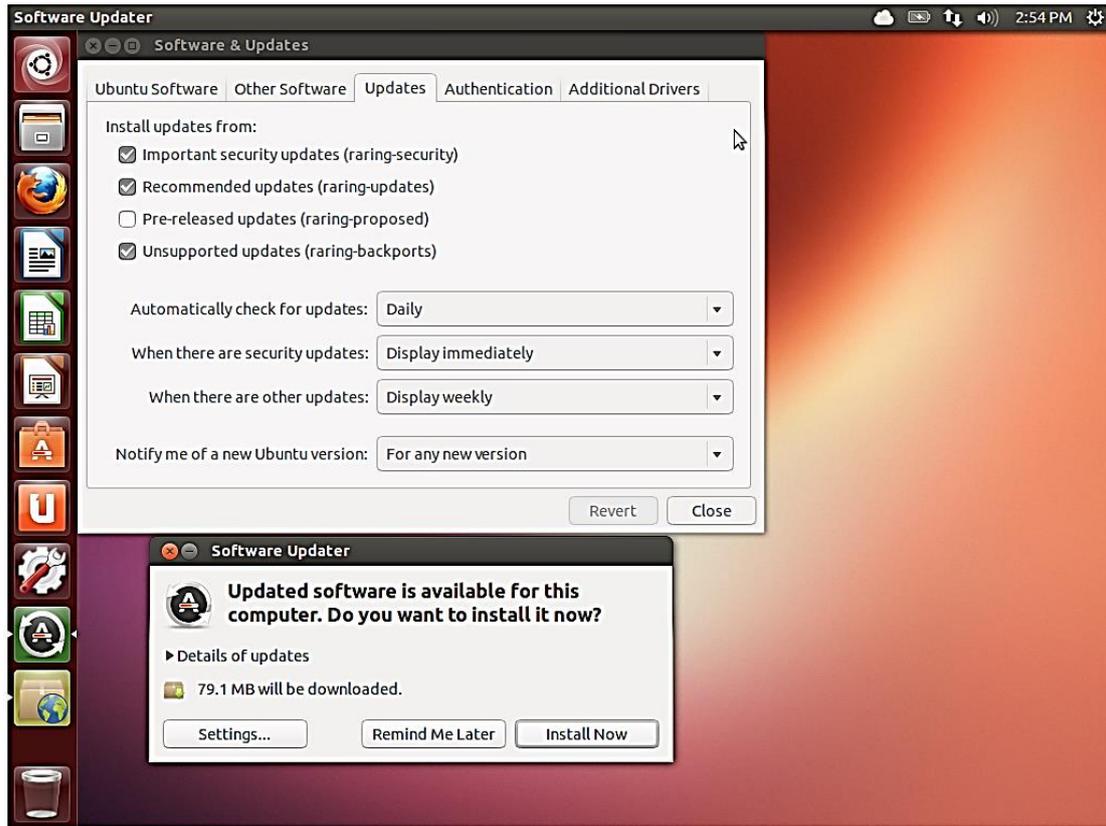
LibreOffice يمكن أن يقوم أيضا بعمل تنسيقات formats للملفات الأخرى، مثل Microsoft Office أو ملفات تنسيق المستندات المحمولة (PDF) Adobe Portable Document Format. بالإضافة إلى ذلك وباستخدام ملحقات extensions يمكن دمجها مع برامج ويكي Wiki software لإعطائك إنترانت قوي powerful intranet.

ينظر المتصفحان فايرفوكس Firefox وجوجل كروم Google Chrome لنظام التشغيل لينكس على انه العميل المهم لديهم. لذلك يمكنك أن تتوقع أن يكون لديك أحدث البرامج latest software المتاحة في نظامك للمتصفحين السابقين. والحصول على المميزات الجديدة new features وإصلاح الأخطاء bug fixes بأسرع وقت. بعض الإضافات plugins مثل أدوبي فلاش Adobe Flash، قد لا تعمل دائما بشكل صحيح لأنهم يهتمون بشركة أخرى غير Linux.

### ٦, ٣ حفظ حاسب لينكس الخاص بك بشكل آمن:

لينكس لا يهتم إذا كنت على لوحة المفاتيح keyboard لجهاز حاسب computer أو متصل عبر الانترنت، لذلك عليك أن تأخذ في الحسبان بعض الاحتياطات الأساسية للتأكد من ان بياناتك data آمنة ومحصنة. أسهل ما يمكنك القيام به هو استخدام كلمة مرور password فريدة unique وجيدة good في كل موقع ومكان نستعرضه، وخصوصا على جهازك المحلي local machine. كلمة السر الجيدة لا تقل عن ١٠ حروف وتحتوي على مزيج من الأرقام والحروف والرموز الخاصة. استخدم حزمة مثل KeePassX لتوليد كلمات السر، وبعد ذلك كل ما يجب عليك فعله هو تسجيل الدخول بكلمة المرور login password إلى جهازك ثم كتابة كلمة المرور لفتح ملف KeePassX الخاص بك.

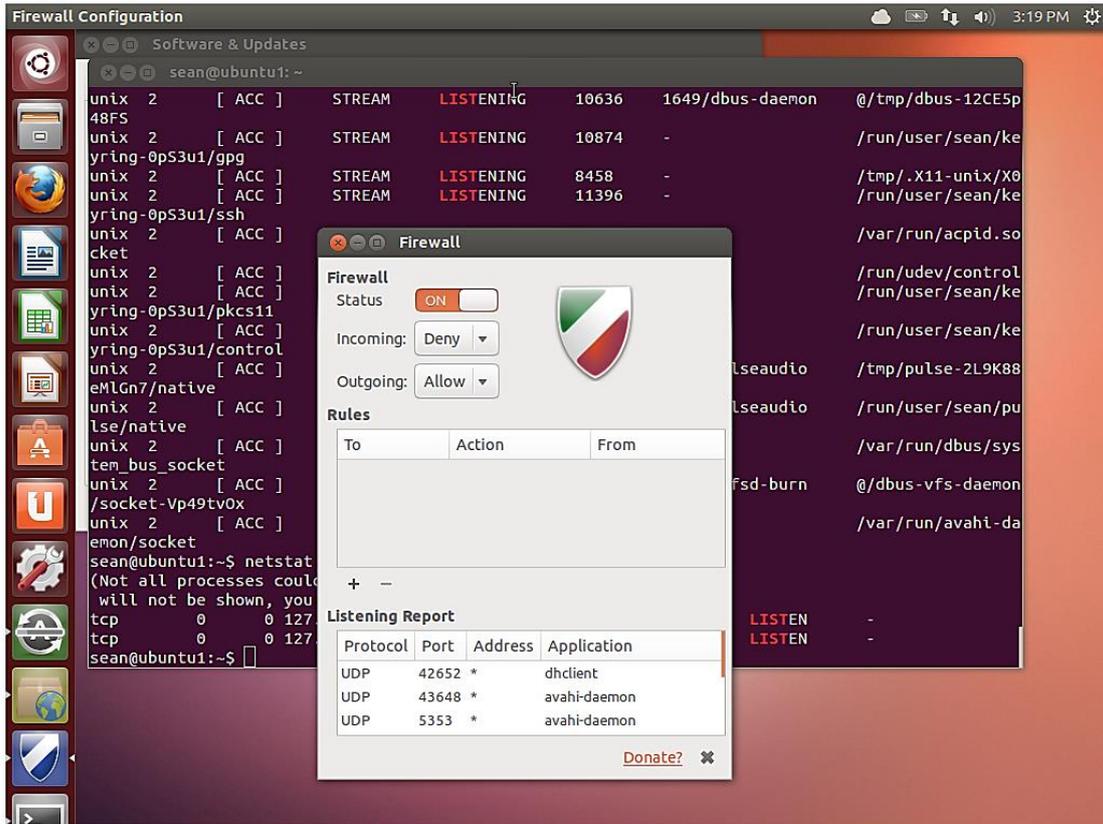
قم بعمل نقطة جدولة دورية للتحقق من وجود تحديثات updates. الشاشة التالية مثال على ضبط تحديث برامج أوبونتو Ubuntu software update configuration، والذي يتوفر من قائمة الإعدادات.



### ضبط تحديث برامج أوبونتو

في الجزء العلوي من الشاشة السابقة، نرى خاصية التحقق من وجود تحديثات بشكل يومي **daily basis**. إذا كانت هناك تحديثات تتعلق بالأمان **security updates**، سيتم على الفور سؤالك للقيام بتنصيبها. خلاف ذلك، سوف تحصل على التحديثات على شكل دفعات أسبوعية **batched up**. اما الجزء السفلي من الشاشة فهو نافذة حوار تظهر عند وجود تحديثات جديدة **updates**. كل ما عليك فعله هو الضغط على التنصيب الآن **Install Now** وسيتم التحديث!

وأخيراً، سوف تحتاج إلى حماية جهاز الحاسب الخاص بك من قبول الاتصالات الواردة **incoming connections**. جدار الحماية **firewall** هو جهاز تصفية **filter** حركة مرور الشبكة **network traffic**، ولينكس لديها واحد مدمج في نظامها. **gufw** هو واجهة رسومية **graphical interface** لجدار الحماية والغير معقد لنظام التشغيل أوبونتو **Ubuntu**.



### جدار حماية أوبونتو

ببساطة، عند تغيير حالة الإعدادات إلى فعال on، سوف تقوم بحجب block كل حركة المرور traffic والقادمة إلى جهاز الحاسب computer الخاص بك الا إذا قمت بإنشائها initiated بنفسك. يمكنك اختيار selective الاتصالات التي تريدها والسماح allow لها بالمرور من وإلى جهازك وذلك عن طريق النقر على علامة زائد plus sign.

الجدير بالذكر، انت تستخدم برنامج iptables والمدمج built-in في نظام system جدار الحماية firewall. وهو متوفر في واجهة المستخدم الرسومية (GUI) -Graphical User Interface بدلاً من كتابة اكواد البرنامج المعقدة؛ يتم استخدام الواجهة الرسومية. تلك الواجهة تسمح لك ببناء نظام فعال effective policy لجهاز لينكس المكتبي desktop، ولكنه لا يقارن بما يستطيع iptables أن يقوم به.

### ٣,٧ حماية نفسك

عندما تتصفح الويب web، فإنك تترك بصمة رقمية digital footprint خاصة بك. الكثير من معلومات information هذه البصمة يتم تجاهلها ignored، وبعضها يتم تجميعه لخدمة الإحصاءات collect statistics والمستخدمة للدعاية advertising، ولكن بعض تلك المعلومات يمكن استخدامها لأغراض خبيثة malicious purposes.

كقاعدة عامة يجب ألا تثق بالمواقع sites التي تتفاعل معها interact with. استخدام كلمات مرور passwords مختلفة في كل موقع بحيث إذا تم اختراق hacked هذا الموقع، فإن كلمة السر المخترقة لا يمكن استخدامها لدخول access إلى حساباتك في مواقع أخرى. قم باستخدام KeePassX هو أسهل طريقة لتوليد كلمات مرور وحفظها. وأيضاً قم بتقليل limit كمية المعلومات information التي تعطيها إلى المواقع إلى ما هو مطلوب فقط. فمثلاً اعطاء اسم والدتك وتاريخ الميلاد قد يساعد في استعادة حسابك في الشبكة الاجتماعية social network إذا فقدت كلمة المرور الخاصة بك، ولكنه في نفس الوقت قد يستخدم ضدك بغرض انتحال شخصيتك impersonate والوصول إلى حسابك البنكي.

المحفوظات أو الكوكيز Cookies هي الطريقة الرئيسية main mechanism التي تستخدمها المواقع websites لتتبعك track. في بعض الأحيان يكون التتبع جيد مثل تتبع عربة التسوق الخاصة بك أو في عملية التسجيل مرة أخرى عند العودة إلى نفس الموقع.

عندما تتصفح مواقع الانترنت، يقوم خادم الويب بإرسال ملفات Cookies (وهي عبارة عن جزء صغير من النص) مع صفحة الويب التي طلبتها. متصفحك يقوم بتخزين تلك الملفات ثم يقوم بإعادة إرسالها عند كل طلب جديد منك لذلك الموقع. لذلك فإن لكل موقع من مواقع الانترنت ملف Cookies خاص به، فمثلاً لا يتم إرسال المحفوظات من موقع example.com إلى موقع example.org.

العديد من المواقع لديها نص مدمج embedded scripts يأتي من طرف ثالث third parties، مثل لوحة إعلانية banner advertisement أو تحليلات تتبعه analytics pixel. تتبع البيكسل tracking pixel هو تتبع وصول المستخدم إلى محتوى ما على شبكة الانترنت. فإذا كان موقع example.com وموقع

example.org يحتويان على بكسل تتبع tracking pixel مثلا من المعلن، فانه سيتم ارسال ملف المحفوظات Cookies نفسه عند تصفح كلا الموقعين. عندها سيعلم المعلن أنك قمت بزيارة example.com و example.org.

عند الضغط على زر الإعجاب "Like" في عالم الشبكات الاجتماعية social network، ومع الوصول الكافي للمعلومات broad enough reach فإنه يمكن أن يحصل موقع website على شبكة الانترنت على معرفة الصفحات التي تقوم عادة بتكرار زيارتها؛ وبالتالي معرفة اهتماماتك الشخصية interests وشريحتك السكانية demographics.

هناك استراتيجيات strategies مختلفة للتعامل مع هذا الأمر. منها مثلا أن تقوم بتجاهل ignore موضوع التتبع. اما الأخرى فهي الحد limit من تتبع البيكسل tracking pixels الذي تقوم بقبوله، إما عن طريق منع التتبع تماما blocking أو عن طريق القيام بعملية حذف بيانات التتبع clearing them out بشكل دوري periodically.

في الشكل أدناه نرى الإعدادات settings المتعلقة بالمحفوظات cookie في متصفح الفايرفوكس Firefox. في الجزء العلوي نشاهد أن المستخدم user اختار لا لعملية التتبع no to track. وهي في الحقيقة عبارة عن علامة اختيارية voluntary tag تصاحب أي طلب request ترسله لأي موقع. هناك بعض المواقع ستحترم رغبتك. أما في الجزء السفلي فنرى اختيار لا لتذكر never remember المحفوظات cookie من طرف ثالث third party cookies، كما نرى أيضا الاختيار حذف remove المحفوظات العادية regular cookies مثل تصفح موقع ما وذلك بعد إغلاق البرنامج.

التغيير في إعدادات الخصوصية privacy settings يجعلك مجهول anonymous على الانترنت Internet، ولكن يمكنه أيضا أن يسبب مشاكل مع بعض المواقع sites التي تعتمد على ملفات المحفوظات من طرف ثالث third party cookies. إذا حدث هذا، فسيكون لديك المقدرة للسماح لبعض ملفات المحفوظات ليتم حفظها.



### خيارات المحفوظات في متصفح Firefox

هنا أيضا لديك خيار لنسيان forget تاريخ بحث search history أو عدم التتبع not track على الإطلاق. مع إزالة عمليات البحث السابقة search history removed، لن يكون هناك سجل record على جهاز الحاسب المحلي الخاص بك من المواقع sites التي قمت بزيارتها.

إذا كنت تشعر بقلق بالغ بان تكون مجهول anonymous على شبكة الانترنت، فيمكنك تحميل download واستخدام متصفح Tor Browser. المتصفح السابق هو اختصار لما يسمى Onion Router والذي هو عبارة عن شبكة عامة network of publically تدير خوادم servers تعمل على إخفاء حركة المرور الخاصة بك. المتصفح الذي يأتي مع تلك الحزمة لا يحتوي على مزايا stripped down version، وعبارة أخرى، فان Tor Browser لا يقوم بتنفيذ أي برنامج نصي scripts، لذلك السبب قد لا تعمل بعض المواقع sites بشكل صحيح. ومع ذلك، فإن هذا المتصفح هو أفضل وسيلة لإخفاء الهوية identity الخاصة بك إذا كنت ترغب بذلك.

## المعمل الرابع – أساسيات سطر الأوامر

# Command Line Skills

## ٤,١ مقدمة

هذا هو المعمل الرابع: أساسيات سطر الأوامر. عن طريق أداء هذا المعمل، سوف تتعلم كيفية استخدام القشرة .shell

في هذا التمرين، سوف تقوم بالمهام التالية:

- استكشاف مميزات Bash.
- استخدام المتغيرات في shell.
- فهم كيفية استخدام globbing.
- القدرة على الاستفادة من الاقتباس.

## ٤,٢ الملفات والمراجع

في هذه المهمة، فإننا سوف نعمل على واجهة سطر الأوامر (CLI) – Command Line Interface في نظام التشغيل لينكس وذلك لتعلم واستكشاف كيفية كتابة الأوامر الأساسية ومن ثم تنفيذها. معظم المستخدمين على دراية بكيفية تنفيذ الأوامر باستخدام واجهة المستخدم الرسومية Graphical User Interface (GUI). لذلك، فإن هذه المهمة ستقدم بعض المفاهيم الجديدة إذا كانت هذه هي المرة الأولى التي تتعامل فيها مع CLI. لاستخدام CLI فسوف تحتاج إلى كتابة الأمر الذي تريد تشغيله .run. في نظام النوافذ window سوف تكتب الأمر كتطبيق محاكي terminal emulator application؛ داخل النافذة الطرفية Terminal وسيعرض النظام system موجه سطر الأوامر prompt، والذي يتبعه مؤشر وامض، كما هو واضح في الشكل ادناه:

```
sysadmin@localhost:~$
```

موجه الأوامر prompt

قد تحتاج إلى الضغط على مفتاح إدخال Enter وذلك لعرض موجه الأوامر.

موجه الأوامر `prompt` يزودك باسم المستخدم الحالي للنظام، وهو في الحالة السابقة `sysadmin`. كما يخبرك أيضا باسم المضيف أو جهاز الحاسب الذي تستخدمه وهو هنا `localhost`. وأخيرا الدليل المحلي `directory` ويرمز له بعلامة تيلدا `~`.

عند كتابة أمر فسوف يظهر ذلك الأمر في مؤشر النص `text cursor`. لتحرير ما تقوم بكتابته من الأوامر، يمكنك استخدام مفاتيح التحكم مثل البداية `home` ونهاية `end` ومسافة للخلف `backspace` ومفاتيح الأسهم `arrow keys` بغرض التحرير.

وبمجرد الانتهاء من كتابة الأمر بشكل صحيح، اضغط `Enter` لتنفيذ ذلك.

١,٢,٤ الخطوة ١

الأمر التالي يعرض نفس المعلومات التي تراها في الجزء الأول من الموجه `prompt` وهو اسم المستخدم الحالي للنظام. تأكد من أنك قمت بالنقر على نافذة `Terminal` أولا ثم اكتب الأمر التالي يليه مفتاح `Enter`:

```
whoami
```

وينبغي أن يكون التنفيذ مشابه لما يلي:

```
sysadmin@localhost:~$ whoami
sysadmin
sysadmin@localhost:~$
```

تنفيذ الأمر `whoami`

نتيجة تنفيذ الأمر `whoami` في المثال السابق هو `sysadmin`، وهو يعرض اسم المستخدم الحالي. على الرغم من أن اسم المستخدم الخاص بك معروض في الموجه ولكن وفي حالة عدم وجوده فيمكن أن تستخدم الأمر السابق للحصول على تلك المعلومة.

## ٤،٢،٢ الخطوة ٢

يعرض الأمر التالي معلومات حول النظام الحالي. لتتمكن من رؤية اسم النواة kernel الذي تستخدمها، اكتب الأمر التالي في terminal:

```
uname
```

وينبغي أن يكون التنفيذ مشابه لما يلي:

```
sysadmin@localhost:~$ uname
Linux
```

### تنفيذ الأمر uname

العديد من الأوامر التي يتم تنفيذها تخرج على بشكل نصي. يمكنك تغيير التنفيذ التي تنتجها الأوامر باستخدام خيارات options بعد الأمر.

الخيارات لأمر ما يمكن ان تحدد بعدة طرق. تقليديا في يونيكس UNIX، تكتب الخيارات بواسطة شرطة يليها حرف آخر مثل n- .

اما في لينكس Linux، الخيارات في بعض الأحيان يمكن أن تكتب بواسطة شرطتان تليها كلمة واحدة أو كلمة موصولة مثل --nodename.

نفذ الأمر uname مرة أخرى في terminal، مرة مع الخيار n- ومرة أخرى مع الخيار --nodename. وذلك لعرض اسم مضيف الشبكة (كما هو موجود في الموجه prompt).

```
uname -n
```

```
uname --nodename
```

وينبغي أن يكون التنفيذ مشابه لما يلي:

```
sysadmin@localhost:~$ uname -n
localhost
sysadmin@localhost:~$ uname --nodename
localhost
```

تنفيذ الأمرين «uname -n ،uname --nodename

٣،٢،٤ الخطوة ٣

يستخدم الأمر pwd لعرض الموقع الحالي أو الدليل الحالي directory. اكتب الأمر التالي لعرض دليل العمل directory:

```
pwd
```

وينبغي أن يكون التنفيذ مشابه لما يلي:

```
sysadmin@localhost:~$ pwd
/home/sysadmin
sysadmin@localhost:~$
```

تنفيذ الأمر pwd

الدليل directory الحالي في المثال أعلاه هو /home/sysadmin. ويشار إليه أيضا بأنه دليل البداية your home directory ، وهو مكان خاص للمستخدم، حيث يستطيع التحكم بالملفات files الموجودة به، وعادة لا يستطيع المستخدمين الآخرين الوصول access إليه. افتراضيا، تتم تسمية هذا الدليل directory بنفس اسم المستخدم ويقع تحت دليل /home. يستخدم لينكس الخط المائل (/) لفصل الدلائل directories لعمل ما يسمى بالمسار، وهو ظاهر في نتيجة تنفيذ الأمر السابق /home/sysadmin. الخط المائل الأول يمثل دليل المستوى الأعلى top level directory والمعروف باسم الدليل الجذر root directory. الحرف تيلدا (~) الظاهر في الموجه prompt يشير أيضا إلى الدليل الحالي. هذا الحرف هو اختصار لتمثيل دليل البداية .home.

لمعلوماتك: `pwd` اختصار لطباعة دليل العمل `print working directory`. حيث أن آلات يونيكس القديمة لم يكن فيها شاشات `monitors`، وإخراج الأوامر كان بالطباعة ومن هنا جاء هذا الاسم. الجدير بالذكر أنه لا تحدث طباعة فعلية في الإصدارات الحديثة ولكن الأمر بقي كما هو.

### ٤,٣ متغيرات القشرة

تستخدم متغيرات القشرة `Shell variables` لتخزين البيانات في لينكس. وتستخدم هذه البيانات عن طريق القشرة نفسها وكذلك البرامج والمستخدمين. يركز هذا القسم على معرفة كيفية عرض قيم المتغيرات القشرة.

#### ٤,٣,١ الخطوة ١

يمكن استخدام الأمر `echo` لطباعة النص `text` وقيمة المتغير `value of a variable` ولإظهار التطابقات `metacharacters` في بيئة القشرة (سيتم الحديث عنه لاحقاً في هذا المعمل). اكتب الأمر التالي للحصول على النص المدخل كما يلي:

```
echo Hello Student
```

وينبغي أن يكون التنفيذ مشابه لما يلي:

```
sysadmin@localhost:~$ echo Hello Student
Hello Student
sysadmin@localhost:~$
```

تنفيذ الأمر `echo Hello Student`

#### ٤,٣,٢ الخطوة ٢

اكتب الأمر التالي لعرض قيمة المتغير `PATH`:

echo \$PATH

وينبغي أن يكون التنفيذ مشابه لما يلي:

```
sysadmin@localhost:~$ echo $PATH
/home/sysadmin/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/s
bin:/bin:/usr/games
sysadmin@localhost:~$
```

تنفيذ الأمر echo \$PATH

يتم عرض المتغير عن طريق وضع علامة الدولار \$ أمام اسم المتغير.

يستخدم هذا المتغير للعثور على اماكن تخزين الأوامر. كل المراجع directories المذكورة أعلاه يتم البحث فيها عند تنفيذ الأمر. على سبيل المثال إذا حاولت تشغيل الأمر date فستظهر القشرة في البداية كما يلي

directory /home/sysadmin/bin directory . ثم في directory /usr/local/sbin . وحالما يتم العثور على

الأمر فان القشرة ستقوم بتشغيله .run

٣,٣,٤ الخطوة ٣

يستخدم الأمر which لتحديد ما إذا كان هناك ملف اسمه تاريخ date قابل للتنفيذ وموجود في الدليل directory والمدرج في قيمة PATH:

which date

وينبغي أن يكون التنفيذ مشابه لما يلي:

```
sysadmin@localhost:~$ which date
/bin/date
sysadmin@localhost:~$
```

تنفيذ الأمر which date

نتيجة الأمر which تخبرك أنه عند تنفيذ الأمر date فان النظام سيشغل الأمر ./bin/date. وكما ذكرنا يستخدم الأمر which لتحديد موقع الأمر .date.

## ٤,٤ Globbing

أحرف glob في لينكس Linux مشابه لما يعرف في العديد من أنظمة التشغيل باسم "wildcard". فباستخدام أحرف glob تستطيع العثور على أسماء الملفات بواسطة نمط من الأنماط.

أحرف glob هي من مميزات القشرة وليست مرتبطة بأمر معين. وتبعاً لذلك يمكنك استخدام أحرف glob مع أي أمر من أوامر لينكس Linux command.

عندما يتم استخدام أحرف glob تقوم القشرة shell بتوسيع كامل النمط لمطابقة match كل الملفات files في الدليل directory المحدد والتي تطابق نمط أحرف glob.

للتوضيح سوف نستخدم الأمر echo لعرض عملية التوسع.

### ١,٤,٤ الخطوة ١

استخدم الأمر echo لعرض كافة أسماء الملفات في الدليل الحالي والتي تطابق النمط " \* ":

```
echo *
```

وينبغي أن يكون التنفيذ مشابه لما يلي:

```
sysadmin@localhost:~$ echo *
Desktop Documents Downloads Music Pictures Public Templates Videos
sysadmin@localhost:~$
```

تنفيذ الأمر echo \*

النجمة asterisk " \* " تطابق صفر أو أكثر من الأحرف في اسم الملف. في المثال أعلاه فان الناتج مطابق لجميع أسماء الملفات filenames في الدليل الحالي directory.  
اما الأمر echo فهو الذي يقوم بإخراج أسماء الملفات على الشاشة.

٢, ٤, ٤, ٢ الخطوة

الأوامر التالية ستعرض كافة الملفات في الدليل الحالي التي تبدأ بحرفي D و P:

echo D\*

echo P\*

وينبغي أن يكون التنفيذ مشابه لما يلي:

```
sysadmin@localhost:~$ echo D*
Desktop Documents Downloads
sysadmin@localhost:~$ echo P*
Pictures Public
sysadmin@localhost:~$
```

تنفيذ الأمرين echo P\*، echo D\*

في المثال \* D تم عرض جميع أسماء الملفات في الدليل الحالي التي تبدأ بحرف D.

٣, ٤, ٤, ٣ الخطوة

علامة النجمة asterisk " \* " يمكن استخدامها مقابل أي حرف من حروف الكلمة أو الاسم. مثلاً الأمر التالي يعرض كافة الملفات files في الدليل directory الحالي التي تنتهي بحرف s:

echo \*s

وينبغي أن يكون التنفيذ مشابه لما يلي:

```
sysadmin@localhost:~$ echo *s
Documents Downloads Pictures Templates Videos
sysadmin@localhost:~$
```

تنفيذ الأمر `echo *s`

٤,٤,٤ الخطوة ٤

لاحظ أن أيضا ان النجمة asterisk أن تكتب عدة مرات في كلمة أو في عدة كلمات أيضا:

```
echo D*n*s
```

وينبغي أن يكون التنفيذ مشابه لما يلي:

```
sysadmin@localhost:~$ echo D*n*s
Documents Downloads
sysadmin@localhost:~$
```

تنفيذ الأمر `echo D*n*s`

سوف ندرس في احرف glob علامة الاستفهام question mark " ؟ " . علامة الاستفهام تطابق تماما حرف واحد فقط.

علامة الاستفهام يمكن استخدامها مثل النجمة في أي مكان في الكلمة ويمكن أن تكتب أيضا عدة مرات.

٥,٤,٤ الخطوة ٥

بما أن كل علامة استفهام تطابق حرف واحدة اذن فان كتابة ستة علامات استفهام سوف تطابق ستة احرف. اكتب ما يلي لعرض أسماء الملفات المكونة من ستة أحرف:

echo ??????

وينبغي أن يكون التنفيذ مشابه لما يلي:

```
sysadmin@localhost:~$ echo ??????  
Public Videos  
sysadmin@localhost:~$
```

تنفيذ الأمر echo ??????

هام: كل علامة استفهام ؟ يجب أن تتطابق مع حرف واحد فقط في اسم الملف، لا أكثر ولا اقل.

٦, ٤, ٤, ٦ الخطوة

تستطيع استخدام علامة الاستفهام مع أحرف glob الأخرى في عملية المطابقة. اكتب ما يلي لعرض أسماء الملفات التي تبدأ بحرف D وتتكون من تسعة أحرف:

echo D?????????

وينبغي أن يكون التنفيذ مشابه لما يلي:

```
sysadmin@localhost:~$ echo D?????????  
Documents Downloads  
sysadmin@localhost:~$
```

تنفيذ الأمر echo D?????????

٧, ٤, ٤, ٧ الخطوة

يمكن الجمع بين حروف glob في نفس الأمر. يعرض الأمر التالي أسماء الملفات التي لا يقل عن ستة أحرف وتنتهي بالحرف s:

```
echo ?????*s
```

وينبغي أن يكون التنفيذ مشابه لما يلي:

```
sysadmin@localhost:~$ echo ?????*s
Documents Downloads Pictures Templates Videos
sysadmin@localhost:~$
```

تنفيذ الأمر `echo ?????*s`

النمط `?????*s` يطابق أسماء الملفات التي تبدأ بخمسة أحرف، ثم يكون هناك صفر أو أكثر من الأحرف لمطابقة علامة النجمة، وأخيرا الأسماء التي تنتهي بحرف `s`.

٨, ٤, ٤, ٨ الخطوة ٨

الحرف التالي من `glob` مشابه لحرف علامة الاستفهام والتي تطابق حرف واحد.

تستخدم الأقواس المربعة `square brackets` `[ ]` " لتحديد التوافق، حيث ان الحرف يمكن أن يكون قائمة `list` أو مدى `range` محدد والذي يعرف باسم فئة الحرف `character class`.

ويمكن أيضا اختيار النفي مع علامة التعجب `exclamation point` `!` " كحرف للاستثناء.

فنجذ في المثال `echo [DP]*` ان الحرف الأول من اسم الملف يمكن أن يكون إما `D` أو `P`. بينما في مثال `echo [!DP]*` ان الحرف الأول يمكن أن يكون أي حرف باستثناء `D` أو `P`:

```
echo [DP]*
```

```
echo [!DP]*
```

وينبغي أن يكون التنفيذ مشابه لما يلي:

```
sysadmin@localhost:~$ echo [DP]*
Desktop Documents Downloads Pictures Public
sysadmin@localhost:~$ echo [!DP]*
Music Templates Videos
sysadmin@localhost:~$
```

تنفيذ الأمرين `echo [DP]*`، `echo [!DP]*`

٩، ٤، ٤، ٩ الخطوة

في الأمثلة القادمة، سيتم تحديد مدى `range` معين من الأحرف. في المثال `echo [D-P]*` الحرف الأول من اسم الملف يمكن أن يكون أي حرف ابتداء من `D` ونهاية بـ `P`. اما المثال `echo [!D-P]*` فهو يخرج عكس المثال الأول حيث يظهر الملفات التي لا تكون واقعة في المدى المحدد اعلاه:

```
echo [D-P]*
echo [!D-P]*
```

وينبغي أن يكون التنفيذ مشابه لما يلي:

```
sysadmin@localhost:~$ echo [D-P]*
Desktop Documents Downloads Music Pictures Public
sysadmin@localhost:~$ echo [!D-P]*
Templates Videos
sysadmin@localhost:~$
```

تنفيذ الأمرين `echo [D-P]*`، `echo [!D-P]*`

إذن الملفات التي ظهرت بدايتها أحد الحروف التالية تبعا للمدى المحدد وهي (E، F، G، H، I، J، K، L، M، N و O)، ولكن ماذا لو كان المدى من رقم ١ إلى حرف A [A-1]؟ الجواب هو استخدام جدول نص للغة اسكي كود ASCII والتي تحدد المدى من وإلى. يمكنك عرض هذا الجدول عن طريق البحث عنه على الانترنت أو كتابة الأمر التالي `ascii`.

وإجابة على السؤال السابق فان جدول نص لغة ASCII سيعرض الملفات التي تبدأ بـ 1 ، 2 ، 3 ، 4 ، 5 ، 6 ، 7 ، 8 ، 9 ، : ، ؛ ، > ، = ، < ، ؟ ، @ و A .

## ٤,٥ الاقتباس

هناك ثلاثة أنواع من الاقتباسات التي تستخدمها قشرة الباش Bash shell: الأول هو علامة الاقتباس المفردة single quote ( ' ) والثاني علامة الاقتباس المزدوجة double quote ( " ) والأخير هو علامة الاقتباس الخلفية back quote ( ` ) . العلامات السابقة لها مميزات خاصة في قشرة الباش Bash shell .

لفهم علامة الاقتباس المفردة والمزدوجة فكر في المثال التالي، علامة النجمة asterisk ( \* ) تستخدم في القشرة كحرف مميز glob لكن كيف أقوم بإدخالها على انها نص فقط؟ هنا تأتي فائدة علامات الاقتباس.

علامة الاقتباس المفردة single quote تمنع القشرة من تفسير interpreting أو توسيع expanding كافة الأحرف الخاصة. وغالبا ما تستخدم علامات الاقتباس المفردة لحماية سلسلة string من التغيير بواسطة القشرة، بحيث يمكن تفسير سلسلة من أمر ما كمعامل parameter .

علامة الاقتباس المزدوجة double quote توقف القشرة من توسيع الأحرف glob مثل النجمة asterisk ( \* ) وعلامة الاستفهام question mark ( ؟ ) وبين القوسين المربعين square brackets ( [ ] ) . اذن علامة الاقتباس المزدوجة لا تسمح بتوسع المتغيرات variable expansion واستبدال الأوامر command substitution .

أخيرا علامة الاقتباس الخلفية back quote وهي لاستبدال الأوامر، بمعنى السماح للأمر بأن ينفذ ضمن خط أمر آخر line of another command .

لاستخدام الاقتباس يجب ادخال زوجين من علامات الاقتباس في البداية وفي النهاية، وإلا فإن القشرة ستتجاهل الأمر بالكامل.

علامات الاقتباس مفيدة لمنع blocking القشرة shell من تفسير interpreting الحروف ، أيضا فان القشرة توفر وسيلة لمنع تفسير حرف وتسمى بالهروب escaping . رمز الهروب escaping هو "\" ويستخدم في بداية الحرف.

## ١,٥,٤ الخطوة

الأمر التالي يظهر استخدام علامة الاقتباس الخلفية back quote ( ` ). مثال لتنفيذ الأمر date داخل سطر الأمر echo:

```
echo Today is `date`
```

وينبغي أن يكون التنفيذ مشابه لما يلي:

```
sysadmin@localhost:~$ echo Today is `date`
Today is Tue Jan 19 15:48:57 UTC 2016
sysadmin@localhost:~$
```

تنفيذ الأمر `echo Today is `date`

## ٢,٥,٤ الخطوة

يمكنك وضع علامة الدولار \$ قبل وبعد الأمر لإنجاز استبدال الأوامر command substitution:

```
echo Today is $(date)
```

وينبغي أن يكون التنفيذ مشابه لما يلي:

```
sysadmin@localhost:~$ echo Today is $(date)
Today is Tue Jan 19 15:51:09 UTC 2016
sysadmin@localhost:~$
```

تنفيذ الأمر echo Today is \$(date)

لماذا اذن هناك طريقتين لعمل نفس الشيء؟ فعلاصة الاقتباس الخلفية Backquote تبدو مشابهة جدا لعلامة الاقتباس المفردة single quote ( ' )، مما يجعل من الصعب إدراك ما يفترض بهما للقيام به. في الأصل

استخدمت القشرة `backquotes`. لكن في الإصدار الحديث من قشرة الباش `Bash shell` تم إضافة تنسيق للأمر `$(command)` لجعل الموضوع أكثر وضوحا.

٣,٥,٤ الخطوة ٣

إذا كنت لا تريد استخدام علامة الاقتباس الخلفية `Backquote` لتنفيذ الأوامر، ضع علامة الاقتباس المفردة `single quote` كما يلي:

```
echo This is the command `date`
```

وينبغي أن يكون التنفيذ مشابه لما يلي:

```
sysadmin@localhost:~$ echo This is the command `date`
This is the command `date`
sysadmin@localhost:~$
```

تنفيذ الأمر `echo This is the command `date``

٤,٥,٤ الخطوة ٤

لاحظ أنه يمكنك أيضا وضع حرف مائل أمام كل حرف `backquote`. تنفيذ ما يلي:

```
echo This is the command `date`
```

وينبغي أن يكون التنفيذ مشابه لما يلي:

```
sysadmin@localhost:~$ echo This is the command `date`
This is the command `date`
sysadmin@localhost:~$
```

تنفيذ الأمر `echo This is the command `date``

## ٥,٥,٤ الخطوة ٥

علامة الاقتباس المزدوجة Double quote لا يكون له أي تأثير على حروف علامة الاقتباس الخلفية .backquote. وتستخدمها القشرة shell كأوامر استبدال command substitution كما يلي:

```
echo This is the command "`date`"
```

وينبغي أن يكون التنفيذ مشابه لما يلي:

```
sysadmin@localhost:~$ echo This is the command "`date`"
This is the command Tue Jan 19 16:05:41 UTC 2016
sysadmin@localhost:~$
```

تنفيذ الأمر " `date` " echo This is the command

## ٦,٥,٤ الخطوة ٦

علامة الاقتباس المزدوجة double quote ( " ) تأثر على أحرف glob وتعطل المعنى خاص بهم. نفذ التالي:

```
echo D*
```

```
echo "D*"
```

وينبغي أن يكون التنفيذ مشابه لما يلي:

```
sysadmin@localhost:~$ echo D*
Desktop Documents Downloads
sysadmin@localhost:~$ echo "D*"
D*
sysadmin@localhost:~$
```

تنفيذ الأمرين "D\*" ،echo D\*

هام: قد تبدو علامات الاقتباس تافهة وغريبة في الوقت الراهن، ولكن كما اكتسبت المزيد من الخبرة في التعامل مع الأوامر، فسوف تكتشف أن فهم كيفية الاقتباس هو أمر بالغ الأهمية عند التعامل مع القشرة.

## ٤,٦ جمل التحكم

عادة عند كتابة أمر ما فإنه ينفذ بالضغط على مفتاح Enter. تقدم قشرة الباش Bash shell ثلاثة جمل مختلفة يمكن استخدامها لفصل عدة أوامر multiple commands مكتوبة معا.

أبسط فاصل بين الأوامر هو الفاصلة المنقوطة semicolon ( ; ). استخدام الفاصلة المنقوطة بين عدة أوامر تسمح بتنفيذها واحدا تلو الآخر، بالتسلسل من اليسار إلى اليمين.

الأحرف && تستخدم لعمل جمل منطقية بحيث يتم تنفيذ أوامر مشروطة ومفصولة باستخدامها. إذا كان الأمر على يسار الحرفين && ناجحا سيتم تنفيذ باقي الجملة. والعكس صحيح فإذا فشل الأمر إلى يسار && فلا يتم تنفيذ تلك الجملة.

نرى بعد ذلك الحرف || وايضا يستخدم لعمل جمل منطقية، وهو حرف تنفيذ شرطي وهذا التنفيذ يكون بشكل مفصول كل امر على حدة، فإذا فشل الأمر في اليسار يتم التنفيذ. وإذا كان الأمر في يسار الحرف || غير متحقق سوف يتم التنفيذ.

لمعرفة كيفية عمل هذه جمل التحكم سنستخدم التالي: الصواب true والخطأ false. الصواب دائما قابل للتنفيذ بينما الخطأ يفشل دائما. قد لا تتوفر أمثلة واقعية لكيفية عمل كلاً من جمل التحكم && و || من دون إدخال أوامر جديدة.

### ٤,٦,١ الخطوة ١

هنا سيتم تنفيذ الأوامر الثلاثة التالية ولكن بشكل مفصول بواسطة الفاصلة المنقوطة semicolons:

```
echo Hello; echo Linux; echo Student
```

كما ترون تظهر النتائج لتنفيذ الأوامر الثلاثة بالتتابع:

```
sysadmin@localhost:~$ echo Hello; echo Linux; echo Student
Hello
Linux
Student
sysadmin@localhost:~$
```

تنفيذ الأمر `echo Hello; echo Linux; echo Student`

٢, ٦, ٤ الخطوة

الآن نضع ثلاث أوامر معا ونفصلها بفواصل منقوطة semicolons، حيث ينفذ الأمر الأول مع نتيجة الخطأ:

```
false; echo Not; echo Conditional
```

وينبغي أن يكون التنفيذ مشابه لما يلي:

```
sysadmin@localhost:~$ false; echo Not; echo Conditional
Not
Conditional
sysadmin@localhost:~$
```

تنفيذ الأمر `false; echo Not; echo Conditional`

لاحظ أنه في المثال السابق كل الأوامر الثلاثة تم تنفيذها على الرغم من أن الأمر الأول فشل حيث لا تظهر نتيجة الأمر الخاطئ `false`. لذلك فان فصل الأوامر بالفواصل المنقوطة semicolons ( ; ) يجعلها مستقلة تماما عن بعضها البعض.

٣, ٦, ٤ الخطوة

هنا سنستخدم جملة (و) `and` للفصل بين الأوامر:

echo Start && echo Going && echo Gone

وينبغي أن يكون التنفيذ مشابه لما يلي:

```
sysadmin@localhost:~$ echo Start && echo Going && echo Gone
Start
Going
Gone
sysadmin@localhost:~$
```

تنفيذ الأمر echo Start && echo Going && echo Gone

عندما تنفيذ جملة echo بشكل صحيح فسيتم إرجاع القيمة value of success، مما يسمح للعبارة التالية بان يتم تنفيذها أيضا.

٤,٦,٤ الخطوة ٤

استخدم (و) and مع الأوامر التي تفشل كما هو موضح أدناه:

echo Success && false && echo Bye

وينبغي أن يكون التنفيذ مشابه لما يلي:

```
sysadmin@localhost:~$ echo Success && false && echo Bye
Success
sysadmin@localhost:~$
```

تنفيذ الأمر echo Success && false && echo Bye

أول أمر echo نجح ويمكن رؤية نتيجته. أمر الخطأ false نفذ والنتيجة هي الفشل وبالتالي لن يتم تنفيذ أمر echo الاخير.

## ٥, ٦, ٤ الخطوة

حرف (أو) or يفصل بين الأوامر التالية وهو الذي يوضح الفرق قبل وبعد استخدامه، ايضا إذا كانت أول عملية ناجحة فإنها تسبب عدم التنفيذ:

```
false || echo Fail Or
```

```
true || echo Nothing to see here
```

وينبغي أن يكون التنفيذ مشابه لما يلي:

```
sysadmin@localhost:~$ false || echo Fail Or
Fail Or
sysadmin@localhost:~$ true || echo Nothing to see here
sysadmin@localhost:~$
```

تنفيذ الأمرين 'false || echo Fail Or' و 'true || echo Nothing to see here'

## ٧, ٤ سجل القشرة

تحتفظ قشرة الباش Bash shell على سجل تاريخي history للأوامر التي كتبت. تلك الأوامر Previous commands يمكن الوصول إليها بسهولة وبعده طرق.

أول وأسهل طريقة لتذكر أمر سابق هي استخدام مفتاح السهم للأعلى up arrow key. حيث ان كل مرة تضغط عليه تظهر أمر واحد، استمر بالضغط إلى ان تصل للأمر الذي تريده. وعكسه مفتاح السهم لأسفل down arrow key حيث سوف يذهب إلى الأمام عبر السجل التاريخي history للأمر الذي تريده.

عندما تجد الأمر الذي تريد تنفيذه يمكنك استخدام مفاتيح الأسهم لليسار left arrow keys ومفاتيح الأسهم لليمين right arrow keys لوضع المؤشر عليه وتعديله. مفاتيح أخرى مفيدة للتعديل مثل مفتاح البداية Home والنهاية End ومسافة للخلف Backspace وحذف Delete.

وهناك طريقة أخرى لاستخدام خاصية السجل هو كتابة امر السجل التاريخي `history` لعرض قائمة بالأوامر السابقة مع الترتيم. الارقام الموجودة على يمين الأمر يمكن استخدامها لتنفيذ ذلك الأمر مرة أخرى. ايضا أمر السجل `history` له عدد من الخيارات، وباستخدام تلك الخيارات يمكن التعديل بحيث تظهر الأوامر التي سيتم عرضها أو التي سيتم تخزينها.

١,٧,٤ الخطوة

نفذ الأوامر التالية ثم اكتب امر السجل `history`:

`date`

`clear`

`echo Hi`

`history`

تذكر: الأمر `date` سيطبع الوقت والتاريخ على النظام. اما امر `clear` فهو يزيل الأوامر من الشاشة.

وينبغي أن يكون التنفيذ مشابه لما يلي:

```
sysadmin@localhost:~$ echo Hi
Hi
sysadmin@localhost:~$ history
 1 date
 2 clear
 3 echo Hi
 4 history
sysadmin@localhost:~$
```

تنفيذ الأوامر `date`، `clear`، `echo Hi`، `history`

عند التنفيذ؛ فمن المحتمل جداً أن تختلف أرقام الأوامر عن الأرقام المذكورة أعلاه. هذا لأنك على الأرجح قد نفذت عددا مختلفا من الأوامر.

٢,٧,٤ الخطوة ٢

يستخدم الأمر التالي لعرض عدد محدود من الأوامر، حيث يمكن لأمر السجل `history` ان يعتبر عددا ما كعامل `parameter` ليقوم بعرضه. اكتب الأمر التالي لعرض الأوامر الخمس الماضية من السجل:

```
history 5
```

وينبغي أن يكون التنفيذ مشابه لما يلي:

```
sysadmin@localhost:~$ history 5
185 false || Fail Or
186 false || echo Fail Or
187 true || echo Nothing to see here
188 history
189 history 5
sysadmin@localhost:~$
```

تنفيذ الأمر `history 5`

٣,٧,٤ الخطوة ٣

لتنفيذ الأمر مرة أخرى، اكتب علامة التعجب `!` exclamation point ورقم ذلك الأمر. على سبيل المثال نفذ الأمر ٩٤:

```
!94
```

التنفيذ هنا كما يلي:

```
sysadmin@localhost:~$ ! 94
-bash: 94: command not found
sysadmin@localhost:~$
```

تنفيذ الأمر 94 !

لاحظ أنه لم يظهر اي أمر لأننا لم نكتب 94 أمر في القشرة.

٤,٧,٤ الخطوة ٤

بعد ذلك، جرب الوصول إلى سجل تاريخ الأوامر `history` باستخدام مفاتيح الأسهم صعوداً `up arrow keys` ومفاتيح الاسهم نزولاً `down arrow keys`. استمر بالضغط على مفتاح السهم للأعلى حتى تجد الأمر الذي تريد تنفيذه. عند الحاجة استخدم مفاتيح أخرى لتحرير الأوامر ثم اضغط على مفتاح الإدخال `Enter` لتنفيذها.

## المعمل الخامس – الحصول على المساعدة

## Getting Help

## ٥,١ مقدمة

هذا هو المعمل الخامس: الحصول على المساعدة. عن طريق أداء هذا التمرين، سوف تتعلم كيفية الحصول على المساعدة في الأوامر وفي البحث عن الملفات.

في هذا التمرين، سوف تقوم بالمهام التالية:

- استخدام عدة أنظمة مساندة للحصول على المساعدة في الأوامر.
- تعلم كيفية تحديد الأوامر.

## ٥,٢ الحصول على المساعدة

في هذه المهمة `task` سوف نستكشف طريقة وكيفية الحصول على المساعدة `help` في نظام لينكس. وهو امر مهم عندما تجد أنك عالق `stuck` في نقطة ما؛ أو عندما لا تستطيع أن تتذكر `remember` كيف يعمل أحد الأوامر `command`.

بالإضافة إلى عمليات البحث `searches` على الانترنت `Internet`، يوفر نظام التشغيل لينكس مجموعة متنوعة من التقنيات `techniques` لمعرفة تفاصيل أكثر عن أمر ما `command` أو ميزة معينة `feature`. ومما لا شك فيه ان معرفة هذه التقنيات المختلفة `different` تسمح لك بالعثور `find` على المعلومة التي تريدها بكل سرعة وسهولة `easily and quickly`.

### ٥,٢,١ الخطوة ١

تنفيذ الأوامر في وعاء `Bash` عن طريق كتابة الأمر ثم الضغط على مفتاح `Enter`. على سبيل المثال، اكتب الأمر التالي لعرض تاريخ اليوم:

date

يجب أن يكون التنفيذ مشابه لما يلي:

```
sysadmin@localhost:~$ date
Tue Jan 19 17:27:20 UTC 2016
sysadmin@localhost:~$
```

تنفيذ الأمر data

٢,٢,٥ الخطوة ٢

لمعرفة المزيد حول الأوامر، ادخل على صفحات دليل المستخدم manual page بواسطة الأمر .man على سبيل المثال نفذ الأمر التالي لمعرفة المزيد عن الأمر :date

```
man date
```

ويظهر الأمر في الموجه كما يلي:

```
sysadmin@localhost:~$ man date
```

الأمر man date في الموجه

وينبغي أن يكون التنفيذ مشابه لما يلي:

DATE (1) User Commands DATE (1)

#### NAME

date - print or set the system date and time

#### SYNOPSIS

date [OPTION]... [+FORMAT]

date [-u|--utc|--universal] [MMDDhhmm[[CC]YY][.ss]]

#### DESCRIPTION

Display the current time in the given FORMAT, or set the system date.

**-d, --date=STRING**

display time described by STRING, not `now'

**-f, --file=DATEFILE**

like --date once for each line of DATEFILE

**-r, --reference=FILE**

display the last modification time of FILE

**-R, --rfc-2822**

output date and time in RFC 2822 format. Example:

Mon, 07 Aug

Manual page date(1) line 1 (press h for help or q to quit)

تنفيذ الأمر man date

ملاحظة: الوثائق التي يتم عرضها بالأمر man يطلق عليهم اسم صفحات دليل المستخدم manual page.

إذا استطاع الأمر man العثور في دليل المستخدم manual page لمعطيات argument، عندها سيتم عرض النتائج باستخدام أمر يسمى أقل less.

يصف الجدول table التالي مفاتيح مفيدة useful keys تستخدم مع امر أقل less ذلك للتحكم control في إخراج العرض display output كما يلي:

المفتاح	وظيفته
الحرف h أو H	عرض المساعدة
الحرف q أو Q	الخروج من المساعدة أو الصفحات دليل المستخدم
مفتاح المسافة Spacebar أو مفتاح صفحة للأسفل PageDown	تحريك الشاشة للأمام
الحرف b أو مفتاح صفحة للأعلى PageUp	تحريك الشاشة إلى الخلف
مفتاح ادخال Enter أو سهم للأسفل down arrow	تحرك سطر واحد للأسفل
سهم للأعلى Up arrow	تحرك سطر واحد لأعلى
حرف I متبوع بنص للبحث text to search	البحث من البداية إلى الأمام
حرف؟ متبوع بنص للبحث text to search	بدء البحث من الخلف
حرف n	الانتقال إلى النص التالي الذي يطابق البحث
حرف N	الانتقال إلى النص السابق الذي يطابق البحث

جدول للمفاتيح المستخدمة مع امر less أقل للتحكم في إخراج العرض

٣, ٢, ٥ الخطوة ٣

اكتب حرف h لترى قائمة من أوامر الحركة. بعد قراءة أوامر الحركة اكتب حرف q للعودة إلى وثيقة. كما يظهر في الشكل التالي:

## SUMMARY OF LESS COMMANDS

Commands marked with \* may be preceded by a number, N.

Notes in parentheses indicate the behavior if N is given.

A key preceded by a caret indicates the Ctrl key; thus ^K is ctrl-K.

```
h H          Display this help.
q :q Q :Q ZZ  Exit.
```

### MOVING

```
e ^E j ^N CR * Forward one line (or N lines).
y ^Y k ^K ^P * Backward one line (or N lines).
f ^F ^V SPACE * Forward one window (or N lines).
b ^B ESC-v    * Backward one window (or N lines).
z              * Forward one window (and set window to N).
w              * Backward one window (and set window to N).
ESC-SPACE     * Forward one window, but don't stop at
                end-of-file.
d ^D          * Forward one half-window (and set half-window to
                N)
u ^U          * Backward one half-window (and set half-window to
                N)
ESC-) RightArrow * Left one half screen width (or N positions).
HELP -- Press RETURN for more, or q when done
```

### تنفيذ الأمر h

كلما تعلمت المزيد عن نظام التشغيل لينكس ستجد أن صفحات دليل المستخدم `man pages` تعبر مصدر قيم للغاية للمعلومات.

## ٤, ٢, ٥ الخطوة ٤

عمليات البحث ليست حساسة لحالة الاحرف case sensitive ولا تلتف من الاسفل للأعلى. ابدأ البحث إلى الأمام لكلمة ملف file بكتابة:

```
/file
```

لاحظ أن ما ستكتبه سيظهر في الجزء الأيسر السفلي من الشاشة.

```
-r, --reference=FILE
        display the last modification time of FILE
-R, --rfc-2822
        output date and time in RFC 2822 format. Example:
        Mon, 07 Aug
/file
```

تنفيذ الأمر /file

## ٥, ٢, ٥ الخطوة ٥

لاحظ أن النص المطابق لعملية البحث تم تمييزه باللون الابيض. يمكنك البحث عن التالي بالضغط على حرف n. أو الانتقال إلى النتائج السابقة بالضغط على حرف N:

**-f, --file=DATEFILE**

like `--date` once for each line of `DATEFILE`

**-r, --reference=FILE**

display the last modification time of `FILE`

**-R, --rfc-2822**

output date and time in RFC 2822 format. Example:

Mon, 07 Aug

2006 12:34:56 -0600

**--rfc-3339=TIMESPEC**

output date and time in RFC 3339 format.

`TIMESPEC='date', 'seconds', or 'ns'` for date and time to the indicated precision.

Date and time components are separated by a single space:

2006-08-07 12:34:56-06:00

**-s, --set=STRING**

set time described by `STRING`

**-u, --utc, --universal**

print or set Coordinated Universal Time

**--help** display this help and exit

Manual page date(1) line 18/204 24% (press h for help or q to quit)

نتيجة البحث

## ٥,٢,٦ الخطوة ٦

تستخدم اوامر الحركة movement commands كما هو موضح سابقا (مثل استخدام مفتاح المسافة spacebar للانتقال move down إلى الشاشة التالية one screen) لقراءة صفحة دليل المستخدم man page مع الأمر date. عند الانتهاء من القراءة، اكتب الحرف q للخروج exit من صفحة دليل المستخدم.

## ٥,٢,٧ الخطوة ٧

في بعض الحالات some cases قد لا تتذكر remember الاسم الدقيق exact name لأمر ما command. عندها يمكن استخدام الخيار -k مع الأمر man في صفحات الدليل manual page والحصول على كلمة رئيسية keyword لذلك الأمر. على سبيل المثال تنفيذ الأمر التالي يعرض ملخص لجميع الصفحات دليل المستخدم man pages التي لديها كلمة الرئيسية keyword وهي كلمة "password"، كما هو ظاهر في المثال التالي:

```
man -k password
```

فيكون التنفيذ مشابه لما يلي:

```

sysadmin@localhost:~$ man -k password
chage (1) - change user password expiry information
chgpasswd (8) - update group passwords in batch mode
chpasswd (8) - update passwords in batch mode
cpgr (8) - copy with locking the given file to the
password or gr...
cppw (8) - copy with locking the given file to the
password or gr...
expiry (1) - check and enforce password expiration policy
login.defs (5) - shadow password suite configuration
pam_pwhistory (8) - PAM module to remember last passwords
pam_unix (8) - Module for traditional password authentication
passwd (1) - change user password
passwd (1ssl) - compute password hashes
passwd (5) - the password file
pwck (8) - verify integrity of password files
pwconv (8) - convert to and from shadow passwords and
groups
shadow (5) - shadowed password file
shadowconfig (8) - toggle shadow passwords on and off
unix_chkpwd (8) - Helper binary that verifies the password of
the curren...
unix_update (8) - Helper binary that updates the password of a
given user
vipw (8) - edit the password, group, shadow-password or
shadow-gr...

sysadmin@localhost:~$

```

تنفيذ الأمر man -k password

خيار k- المصاحب للأمر man غالباً ما يكون له كمية كبيرة من النتائج huge amount of output. لذلك سوف نتحدث في معمل لاحق عن تقنيات technique وذلك للحد من تلك النتائج limit this output أو على الأقل السماح بالمرور السريع easily scroll على تلك البيانات though the data. في الوقت الراهن استخدام شريط التمرير scrollbar على الجانب الأيمن on the right hand من نافذة العرض terminal window لتحريك الشاشة to move the display صعوداً ونزولاً up and down حسب الحاجة .as needed

٨, ٢, ٥ الخطوة ٨

لاحظ أن الأمر apropos هو طريقة أخرى another way لعرض view معلومات ملخصة summaries عن كلمة رئيسية keyword في صفحة دليل المستخدم man page، اكتب الأمر التالي في موجهه الأوامر prompt:

```
apropos password
```

فيكون التنفيذ مشابه لما يلي:

```

sysadmin@localhost:~$ apropos password
chage (1) - change user password expiry information
chgpasswd (8) - update group passwords in batch mode
chpasswd (8) - update passwords in batch mode
cpgr (8) - copy with locking the given file to the
password or gr...
cppw (8) - copy with locking the given file to the
password or gr...
expiry (1) - check and enforce password expiration policy
login.defs (5) - shadow password suite configuration
pam_pwhistory (8) - PAM module to remember last passwords
pam_unix (8) - Module for traditional password authentication
passwd (1) - change user password
passwd (1ssl) - compute password hashes
passwd (5) - the password file
pwck (8) - verify integrity of password files
pwconv (8) - convert to and from shadow passwords and
groups
shadow (5) - shadowed password file
shadowconfig (8) - toggle shadow passwords on and off
unix_chkpwd (8) - Helper binary that verifies the password of
the curren...
unix_update (8) - Helper binary that updates the password of a
given user
vipw (8) - edit the password, group, shadow-password or
shadow-gr...

sysadmin@localhost:~$

```

تنفيذ الأمر apropos password

ملاحظة: ليس هناك فرق بين الأمر man -k والأمر apropos حيث أنهما يؤديان نفس الوظيفة.

٩,٢,٥ الخطوة ٩

غالبا ما تكون صفحات دليل المستخدم `man pages` متعددة `multiple` وبنفس الاسم `same name`. على سبيل المثال، الأمر السابق أظهر ثلاث صفحات لكلمة "`passwd`". نفذ `Execute` الأمر التالي `following command` لعرض `view` كلمة "`passwd`" بكلمة كاملة في صفحات دليل المستخدم `man pages` وليس كشرح لها داخل الصفحات:

```
man -f passwd
```

فيكون التنفيذ مشابه لما يلي:

```
sysadmin@localhost:~$ man -f passwd
passwd (5)          - the password file
passwd (1)          - change user password
passwd (1ssl)       - compute password hashes
sysadmin@localhost:~$
```

تنفيذ الأمر `man -f passwd`

إن وجود صفحات لدليل المستخدم `man pages` مختلفة `different` لنفس الاسم `same name` تكون مربكة `confusing` لكثير من مستخدمي `users` لينكس الحديثين. صفحات دليل المستخدم `man pages` ليست فقط لأوامر لينكس ولكن أيضا لملفات النظام `system files` وغيرها من المميزات `features` الموجودة بنظام التشغيل `Operating System`. بالإضافة إلى ذلك يوجد في بعض الأحيان أمران يتشاركان نفس الاسم، كما في المثال المذكور أعلاه.

صفحات دليل المستخدم المختلفة تتميز بوجود الأقسام sections. افتراضيا هناك تسعة أقسام لصفحات دليل المستخدم man pages:

١. برامج قابلة للتنفيذ Executable programs أو أوامر القشرة shell commands.
٢. استدعاءات النظام System calls (الوظائف functions التي توفرها النواة kernel).
٣. استدعاءات المكتبة Library calls (وظائف functions داخل مكتبات البرنامج).
٤. ملفات خاصة Special files (وعادة ما توجد في /dev).
٥. صيغ واتفاقيات الملف File formats and conventions، على سبيل المثال  
/etc/passwd/
٦. ألعاب Games.
٧. متفرقات Miscellaneous (بما في ذلك الحزم الدقيقة macro packages والاتفاقيات conventions)، على سبيل المثال، man(7) و groff(7).
٨. أوامر إدارة النظام System administration commands (عادة تكون للجذر root).
٩. روتين النواة Kernel routines [غير القياسية Non standard].

عند كتابة أمر مثل man passwd يتم البحث في القسم الأول، وإذا تم العثور على تطابق يتم عرض صفحة النتائج. الأمر man -f passwd يُظهر أن في القسم الأول section 1 هناك صفحة دليل المستخدم man page وتظهر كالتالي: passwd: passwd (1) ونتيجة لذلك يتم عرضها بشكل افتراضي.

١٠، ٢، ٥ الخطوة ١٠

لعرض صفحات دليل المستخدم لقسم section آخر، يجب توفر رقم ذلك القسم. على سبيل المثال:

```
man 5 passwd
```

فيكون التنفيذ مشابه لما يلي:

```
PASSWD(5) File Formats and Conversions  
PASSWD(5)
```

**NAME**

```
passwd - the password file
```

**DESCRIPTION**

```
/etc/passwd contains one line for each user account, with  
seven fields
```

```
delimited by colons (":"). These fields are:
```

- o login name
- o optional encrypted password
- o numerical user ID
- o numerical group ID
- o user name or comment field
- o user home directory
- o optional user command interpreter

```
Manual page passwd(5) line 1 (press h for help or q to quit)
```

تنفيذ الأمر `man 5 passwd`

## ١١, ٢, ٥ الخطوة ١١

بدلا من استخدام الأمر `man -f` لعرض كافة أقسام صفحة الدليل للاسم، يمكنك استخدام الأمر `whatis` وذلك لعرض ارقام الأقسام التي توجد بها الكلمة التي تبحث عنها، كما هو موجود بالمثال التالي:

```
whatis passwd
```

فيكون التنفيذ مشابه لما يلي:

```
sysadmin@localhost:~$ whatis passwd
passwd (5)          - the password file
passwd (1)          - change user password
passwd (1ssl)       - compute password hashes
sysadmin@localhost:~$
```

تنفيذ الأمر `whatis passwd`

ملاحظة: ليس هناك فرق بين الأمر `man -f` والأمر `whatis` حيث انهما يقومان بنفس العمل.

## ١٢, ٢, ٥ الخطوة ١٢

تقريبا كل مميزات النظام `system features` مثل الأوامر `commands` ونظام الملفات `system files` لديها صفحات دليل الاستخدام `man pages`. بعض هذه المميزات `features` لها مزايا أكثر تقدما تسمى صفحات المعلومات `info pages`. لرؤية ذلك نفذ الأمر التالي:

```
info date
```

فيكون التنفيذ مشابه لما يلي:

```
File: coreutils.info, Node: date invocation, Next: arch invocation,
Up: Syst\em context
```

```
21.1 `date': Print or set system date and time
```

```
=====  
Synopsis:
```

```
date [OPTION]... [+FORMAT]
```

```
date [-u|--utc|--universal] [ MMDDhhmm[[CC]YY][.ss] ]
```

Invoking `date' with no FORMAT argument is equivalent to invoking it with a default format that depends on the `LC\_TIME' locale category. In the default C locale, this format is `+%a %b %e %H:%M:%S %Z %Y', so the output looks like `Thu Mar 3 13:47:51 PST 2005'.

Normally, `date' uses the time zone rules indicated by the `TZ' environment variable, or the system default rules if `TZ' is not set.  
\*Note Specifying the Time Zone with `TZ': (libc)TZ Variable.

If given an argument that starts with a `+', `date' prints the current date and time (or the date and time specified by the `--date'

```
--zz-Info: (coreutils.info.gz)date invocation, 41 lines --Top-----
```

```
Welcome to Info version 4.13. Type h for help, m for menu item.
```

### تنفيذ الأمر info date

يرى العديد من مستخدمي لينكس ان صفحات المعلومات info pages أسهل في القراءة. وهي غالبا ما تكون مكتوبة على دروس تعليمية lessons، بينما تتم كتابة صفحات دليل المستخدم بصيغة وثائق للنظام.

### ١٣, ٢, ٥ الخطوة ١٣

أثناء عرض صفحة معلومات info page من الخطوة السابقة، اكتب حرف h لترى قائمة من أوامر الحركة movement commands. لاحظ أنها مختلفة عن أوامر الحركة المستخدمة في man pages. بعد قراءة أوامر الحركة، اكتب حرف ل (l) للعودة إلى عرض الوثيقة.

### ١٤, ٢, ٥ الخطوة ١٤

استخدم أوامر الحركة movement commands لقراءة صفحة المعلومات info page الخاصة بالأمر date. عند الانتهاء ضع المؤشر cursor بأي مكان على السطر "Examples of date:" ثم اضغط على مفتاح الإدخال Enter. سيتم عرض وثيقة جديدة تبيّن أمثلة من امر التاريخ date.

### ١٥, ٢, ٥ الخطوة ١٥

اضغط على حرف ل (l) للعودة إلى الشاشة السابقة. عند الانتهاء من القراءة، اكتب حرف q للخروج من صفحة معلومات info page.

### ١٦, ٢, ٥ الخطوة ١٦

وهناك طريقة أخرى للحصول على مساعدة عن طريق اسناد الخيار --help إلى أمر ما. معظم الأوامر تسمح لك بتمرير الخيار help - - وذلك لعرض كيفية استخدام الأوامر الأساسية basic command، كما يظهر في المثال التالي:

```
date --help
```

فيكون التنفيذ مشابه لما يلي:

```

sysadmin@localhost:~$ date --help
Usage: date [OPTION]... [+FORMAT]

  or:  date [-u|--utc|--universal] [MMDDhhmm[[CC]YY][.ss]]

Display the current time in the given FORMAT, or set the system date.

-d, --date=STRING      display time described by STRING, not
                        `now'
-f, --file=DATEFILE    like --date once for each line of
                        DATEFILE
-r, --reference=FILE    display the last modification time of
                        FILE
-R, --rfc-2822          output date and time in RFC 2822 format.
                        Example: Mon, 07 Aug 2006 12:34:56 -0600
--rfc-3339=TIMESPEC    output date and time in RFC 3339 format.
                        TIMESPEC=`date', `seconds', or `ns' for
                        date and time to the indicated precision.
                        Date and time components are separated by
                        a single space: 2006-08-07 12:34:56-06:00
-s, --set=STRING        set time described by STRING
-u, --utc, --universal  print or set Coordinated Universal Time
--help                  display this help and exit
--version               output version information and exit

```

تنفيذ الأمر `date --help`

١٧، ٢، ٥ الخطوة ١٧

بعض مميزات النظام `system features` لديها وثائق مساعدة `help documents` أكثر تفصيلاً وموجودة في المرجع `directory` التالي `/usr/share/doc`. نفذ الأمر التالي لعرض محتويات هذه الوثيقة

:document

```
ls /usr/share/doc
```

فيكون التنفيذ مشابه لما يلي:

```
sysadmin@localhost:~$ ls /usr/share/doc
adduser          libdrm2          libx11-data
apt              libedit2         libxau6
ascii            libelf1          libxcb1
base-files       libffi6          libxdmcp6
base-passwd      libgcc1          libxext6
bash             libgcrypt11      libxml2
bind9            libgdbm3         libxmuu1
bind9-host       libgeoip1        locales
bind9utils       libgettextpo0    login
bsdmainutils     libglib2.0-0     logrotate
bsdutils         libgnutls26      lsb-base
busybox-initramfs libgomp1          makedev
bzip2            libgpg-error0    man-db
ca-certificates libgpm2           mawk
coreutils        libgssapi-krb5-2 mc
cpio             libgssapi3-heimdal mc-data
cron             libhcrypto4-heimdal mime-support
curl             libheimbase1-heimdal mlocate
dash             libheimntlm0-heimdal module-init-tools
```

تنفيذ الأمر `ls /usr/share/doc`

لاحظ أنه في جميع الحالات تقريبا، توفر صفحات دليل المستخدم `man pages` وصفحات المعلومات `info pages` المعلومات التي تحتاجها. ومع ذلك إذا كنت بحاجة إلى مزيد من المعلومات المتعمقة كاحتياجات خاصة لمسؤولي النظام `administrators sometimes` فإنك قد تجد هذه المعلومات في الملفات الموجودة في المرجع `directory` كالتالي `/usr/share/doc`.

## ٥,٣ ايجاد الملفات

في هذه المهمة، سوف نستكشف كيفية البحث search عن الملفات files في النظام system. وهي خاصية مفيد في حالة عدم التمكن من العثور على ملف في النظام، سواءً قمت بإنشاء هذا الملف أو تم إنشاؤه من قبل شخص آخر.

### ٥,٣,١ الخطوة ١

طريقة سهلة للبحث عن ملف باستخدام امر الموقع locate command . على سبيل المثال يمكنك العثور على موقع الملف crontab عن طريق تنفيذ الأمر التالي:

```
locate crontab
```

فيكون التنفيذ مشابه لما يلي:

```
sysadmin@localhost:~$ locate crontab
/etc/crontab
/usr/bin/crontab
/usr/share/doc/cron/examples/crontab2english.pl
/usr/share/man/man1/crontab.1.gz
/usr/share/man/man5/crontab.5.gz
sysadmin@localhost:~$
```

تنفيذ الأمر locate crontab

### ٥,٣,٢ الخطوة ٢

لاحظ أن التنفيذ السابق يتضمن كل الملفات التي تحتوي على كلمة crontab كجزء من اسمها. للبحث عن الملفات التي اسمها فقط "crontab"، استخدم الأمر التالي:

```
locate -b "\crontab"
```

فيكون التنفيذ مشابه لما يلي:

```
sysadmin@localhost:~$ locate -b "\crontab" /etc/crontab
/usr/bin/crontab
sysadmin@localhost:~$
```

تنفيذ الأمر "locate -b "\crontab"

ملاحظة: الأمر موقع locate يستخدم قاعدة بيانات database يتم تحديثها updated عادة مرة واحدة يوميا (عادة عند منتصف الليل). تحتوي قاعدة البيانات تلك database على قائمة بجميع الملفات المحدثة التي على النظام.

ونتيجة لذلك فإن أي ملفات جديدة تم إنشائها اليوم لن تكون موجودة عند البحث مع الأمر موقع locate. إذا كان لديك وصول access إلى النظام كمستخدم اساسي root user (حساب مسؤول النظام the system administrator account)، فيمكنك تحديث هذا الملف يدويا manually عن طريق تشغيل الأمر updated. عادة لا يمكن للمستخدمين العاديين regular users تحديث ملف قاعدة البيانات database file.

طريقة أخرى للبحث عن الملفات الجديدة newer files عن طريق كتابة الأمر find. يبحث هذا الأمر عن الملفات المباشرة live filesystem بدلا من قاعدة البيانات الثابتة static database. وسنرى هذا الأمر في المعامل القادمة.

٣,٣,٥ الخطوة ٣

إذا كنت ترغب فقط في معرفة أين موقع أمر ما فيمكنك تحقيق ذلك مع الأمر whereis:

```
whereis passwd
```

فيكون التنفيذ مشابه لما يلي:

```
sysadmin@localhost:~$ whereis passwd
passwd: /usr/bin/passwd /etc/passwd /usr/share/man/man1/passwd.1.gz /
usr/share/man/man1/passwd.1ssl.gz /usr/share/man/man5/passwd.5.gz
sysadmin@localhost:~$
```

تنفيذ الأمر whereis passwd

الأمر whereis يبحث فقط عن الأوامر commands وصفحات دليل المستخدم man pages وليس عن الملفات .files

تذكر أن هناك أكثر من passwd man page على النظام. لذلك وعند تنفيذ الأمر السابق نرى أسماء ملفات متعددة multiple file names في صفحات دليل المستخدم man pages (الملفات التي تنتهي في .gz هي صفحات دليل المستخدم).

## المعمل السادس – العمل مع الملفات والمراجع

## Working with Files and Directories

## ٦,١ مقدمة

هذا هو المعمل السادس: قائمة الملفات والدلائل. عن طريق أداء هذا التمرين، سوف تتعلم كيفية التنقل وإدارة الملفات والدلائل.

في هذا التمرين، سوف تقوم بالمهام التالية:

- قائمة الملفات والدلائل.
- نسخ ونقل وحذف الملفات والدلائل.

## ٦,٢ الملفات والمراجع

في هذه المهمة سوف تستكشف مفاهيم الملفات `files` والمراجع أو الدلائل `directories`.

في نظام التشغيل لينكس Linux OS يتم تخزين البيانات `data` في ملفات `files` وتخزين الملفات في الدلائل `directories`. يمكنك أن تستخدم المصطلح المجلدات `folders` لوصف الدلائل `directories`.

الدلائل `directories` هي في الواقع ملفات `files` أيضاً؛ والبيانات `data` التي تحتويها هي أسماء للملفات التي تم إدخالها فيها، وهي توجد جنباً إلى جنب مع `inode number` (وهو رقم معرف فريد يخصص لكل ملف) لمعرفة مكان وجود البيانات لهذا الملف على القرص.

كمستخدم لنظام لينكس سوف تحتاج إلى معرفة كيفية التعامل مع هذه الملفات والدلائل، بما في ذلك كيفية عرض قائمة `list` بالملفات `files` في دليل `directory` والقيام بالتحكم فيها من نسخ `copy` وحذف `delete` ونقل `move` لتلك الملفات.

تحذير: أسماء الملفات `files` والدلائل `directories` في لينكس حساسة لحالة الأحرف `case sensitive`.

وهذا يعني أن ملف يسمى "ABC" لا يمكن ان نطلق عليه "abc".

## ١,٢,٦ الخطوة ١

اكتب الأمر التالي لطباعة دليل العمل `working directory`:

```
pwd
```

فيكون التنفيذ مشابه لما يلي:

```
sysadmin@localhost:~$ pwd
/home/sysadmin
sysadmin@localhost:~$
```

تنفيذ الأمر `pwd`

دليل العمل `working directory` هو الدليل الذي تعمل عليه حالياً، أي ان النافذة الخاص به وتسمى `terminal` تعمل حالياً. وهذا ما يسمى أيضاً بالدليل الحالي `current directory`. هذه المعلومة تعتبر مهمة وذلك لأنه عندما تقوم بتشغيل الأوامر مستقبلاً سوف تستجيب لتلك الأوامر بشكل مختلف؛ اعتماداً على الدليل الذي تتواجد فيه حالياً.

تنفيذ الأمر `pwd` في المثال أعلاه هو المسار `/home/sysadmin`. ويمثل الخط المائل الأول الدليل الجذر `root directory` وهو أعلى مستوى في بنية الدليل `directory structure`.

في التنفيذ أعلاه `home` هو دليل `directory` تحت الدليل الجذر `root directory`، ودليل `sysadmin` تحت دليل `home`.

عند فتح نافذة `terminal`؛ ستضعك في دليلك الرئيسي `your home directory`. وهو الدليل الذي لديك الحق في الوصول الكامل `full access` له اما بقية المستخدمين عادة لا يصلون اليه. لرؤية المسار `path` للدليل الرئيسي الخاص بك؛ يمكنك تنفيذ الأمر التالي لعرض قيمة `HOME variable`:

```
echo $HOME
```

ويكون التنفيذ مشابه لما يلي:

```
sysadmin@localhost:~$ echo $HOME
/home/sysadmin
sysadmin@localhost:~$
```

تنفيذ الأمر echo \$HOME

٢,٢,٢ الخطوة ٢

يمكنك استخدام الأمر cd مع المسار path ( / ) لتغيير الدليل الحالي current directory. اكتب الأمر التالي لجعل الدليل الجذر root directory دليل العمل الحالي current working directory الخاص بك، وتحقق مع الأمر pwd:

```
cd /
pwd
```

ويكون التنفيذ مشابه لما يلي:

```
sysadmin@localhost:~$ cd /
sysadmin@localhost:/$ pwd
/
sysadmin@localhost:/$
```

تنفيذ الأمرين / cd ، pwd

٣,٢,٣ الخطوة ٣

للرجوع إلى دليلك your home directory استخدم الأمر cd والذي يمكن تنفيذه بدون مسار ( / ). التغيير إلى دليلك الرئيسي your home directory والتحقق عن طريق كتابة الأوامر التالية:

```
cd
```

```
pwd
```

ويكون التنفيذ مشابه لما يلي:

```
sysadmin@localhost:/$ cd
sysadmin@localhost:~$ pwd
/home/sysadmin
sysadmin@localhost:~$
```

تنفيذ الأمرين cd ، pwd

لاحظ التغيير في الموجه prompt. يمثل الحرف تيلدا (~) دليلك الرئيسي your home directory. ويقوم الموجه prompt باطلاعك على الدليل الذي تتواجد فيه حالياً.

٤, ٢, ٦ الخطوة ٤

الأمر cd مع المسار path ( / ) يأخذك إلى دليل محدد. تنفيذ الأمر cd مع دليل /home عن طريق كتابة الأمر التالي:

```
cd /home
```

```
pwd
```

ويكون التنفيذ مشابه لما يلي:

```
sysadmin@localhost:~$ cd /home
sysadmin@localhost:/home$ pwd
/home
sysadmin@localhost:/home$
```

تنفيذ الأمرين cd / home ، pwd

المسار المطلق absolute path هو المسار الذي يُكتب كاملاً مع الأمر cd وذلك بإضافة علامة المسار وهي مستقيم مائل للأمام forward slash (/). المسارات المطلقة هي دائماً مسارات كاملة من الدليل الجذر root directory إلى الدليل الفرعي sub-directory أو الملف file الذي تريد العمل عليه.

٥, ٢, ٦ الخطوة

تتم العودة إلى دليلك الرئيسي your home directory باستخدام الأمر cd مع تيلدا (~):

```
cd ~
```

```
pwd
```

ويكون التنفيذ مشابه لما يلي:

```
sysadmin@localhost:~/home$ cd ~
sysadmin@localhost:~$ pwd
/home/sysadmin
sysadmin@localhost:~$
```

تنفيذ الأمرين ~ cd ، pwd

عند كتابة الأمر cd مع حرف تيلدا tilde (~)، تقوم النافذة الطرفية terminal بالانتقال إلى الدليل الرئيسي home directory لحساب المستخدم على النظام.

اما عند عدم وجود علامة (/) ، سيتم التوسع والذهاب إلى المسار المكتوب للمستخدم النشط حالياً current user active في القشرة shell.

وإذا كان اسم المستخدم user name يتبعه مباشرة حرف تيلدا tilde (~) ، فان القشرة ستوسع منطقة تيلدا (~) واسم المستخدم user name إلى الدليل الرئيسي home directory لهذا المستخدم. على سبيل المثال، ~bob ستوسع إلى /home/bob.

وتعتبر المسارات paths التي تبدأ مع تيلدا tilde (~) مسارات مطلقة absolute paths لأنه بعد ان تقوم القشرة shell بتوسيع مسار تيلدا tilde path، يتم تشكيل المسار المطلق absolute path.

٦,٢,٦ الخطوة ٦

استخدم الأمر echo أدناه لعرض بعض الأمثلة لإستخدام تيلدا tilde (~) كجزء من مسار path :

```
echo ~ ~sysadmin ~root ~mail ~nobody
```

ويكون التنفيذ مشابه لما يلي:

```
sysadmin@localhost:~$ echo ~ ~sysadmin ~root ~mail ~nobody
/home/sysadmin /home/sysadmin /root /var/mail /nonexistent
sysadmin@localhost:~$
```

تنفيذ الأمر echo ~ ~sysadmin ~root ~mail ~nobody

٦,٢,٧ الخطوة ٧

التغيير إلى الدليل الرئيسي home directory للمستخدم الجذر root user عن طريق كتابة الأمر التالي:

```
cd ~root
```

ويكون التنفيذ مشابه لما يلي:

```
sysadmin@localhost:~$ cd ~root
-bash: cd: /root: Permission denied
sysadmin@localhost:~$
```

تنفيذ الأمر cd ~root

لاحظ رسالة الخطأ `error message`. حيث تشير إلى أن القشرة حاولت تنفيذ الأمر `cd` ثم فشلت بسبب رفض الإذن `permission being denied`؛ وذلك لأنك في حساب المستخدم `sysadmin`. وسوف تتعلم المزيد عن وأذونات الملف والدليل في معمل لاحق.

٨, ٢, ٦ الخطوة ٨

باستخدام المسار المطلق `absolute path`، اذهب إلى الدليل `/usr/bin` ثم اعرض دليل العمل `working directory`، استخدام الأوامر التالية لتنفيذ ذلك:

```
cd /usr/bin

pwd
```

ويكون التنفيذ مشابه لما يلي:

```
sysadmin@localhost:~$ cd /usr/bin
sysadmin@localhost:/usr/bin$ pwd
/usr/bin
sysadmin@localhost:/usr/bin$
```

تنفيذ الأمرين `cd /usr/bin`، `pwd`

٩, ٢, ٦ الخطوة ٩

استخدم المسار المطلق `absolute path` لتغيير الدليل `/usr` وعرض دليل العمل `working directory` وذلك عن طريق الأوامر التالية:

```
cd /usr

pwd
```

ويكون التنفيذ مشابه لما يلي:

```
sysadmin@localhost:~/usr/bin$ cd /usr
sysadmin@localhost:~/usr$ pwd
/usr
sysadmin@localhost:~/usr$
```

تنفيذ الأمرين cd /usr ، pwd

١٠، ٢، ٦ الخطوة

استخدم المسار المطلق absolute path للذهاب إلى الدليل /usr/share/doc وعرض دليل العمل working directory عن طريق الأوامر التالية:

```
cd /usr/share/doc
pwd
```

ويكون التنفيذ مشابه لما يلي:

```
sysadmin@localhost:~/usr$ cd /usr/share/doc
sysadmin@localhost:~/usr/share/doc$ pwd
/usr/share/doc
sysadmin@localhost:~/usr/share/doc$
```

تنفيذ الأمرين cd /usr/share/doc ، pwd

المسار المطلق Absolute مقابل المسارات النسبية Relative pathnames:

لنفترض أنك في الدليل /usr/share/doc وتريد أن تذهب إلى الدليل /usr/share/doc/bash. اكتب الأمر cd /usr/share/doc/bash، في مثل هذه الحالة، استخدام المسار النسبي relative pathnames للذهاب إلى ذلك الدليل أو الملف.

مع المسارات النسبية relative pathnames يتم كتابة الموقع الذي تريد الذهاب اليه من الموقع الحالي وليس من بداية المسار مثل المسار المطلق. وتوضح الأمثلة التالية كيفية استخدام المسارات النسبية relative pathnames.

١١، ٢، ٦ الخطوة ١١

باستخدام المسار النسبي relative path، اذهب إلى الدليل /usr/share/doc/bash وقم بعرض دليل العمل working directory عن طريق الأوامر التالية:

```
cd bash
pwd
```

ويكون التنفيذ مشابه لما يلي:

```
sysadmin@localhost:/usr/share/doc$ cd bash
sysadmin@localhost:/usr/share/doc/bash$ pwd
/usr/share/doc/bash
sysadmin@localhost:/usr/share/doc/bash$
```

تنفيذ الأمرين cd bash ، pwd

ملاحظة: إذا لم يكن هناك دليل باش bash directory تحت الدليل الحالي current directory، فإن الأمر السابق سوف يفشل.

١٢، ٢، ٦ الخطوة ١٢

استخدم المسار النسبي relative path للتغيير من الدليل أعلاه والعودة للخلف خطوة واحدة فقط:

```
cd ..
```

```
pwd
```

ويكون التنفيذ مشابه لما يلي:

```
sysadmin@localhost:/usr/share/doc/bash$ cd ..
sysadmin@localhost:/usr/share/doc$ pwd
/usr/share/doc
sysadmin@localhost:/usr/share/doc$
```

تنفيذ الأمرين `cd ..` ، `pwd`

تمثل النقاط ( .. ) مستوى واحد أعلى عن موقع الدليل الحالي، وبالتالي بدلا من ان تكون في الدليل

`/usr/share/doc/bash` ستذهب إلى الدليل `/usr/share/doc`.

١٣، ٢، ٦ الخطوة

استخدم المسار النسبي `relative path` للذهاب إلى مستوى واحد للأعلى `one level up` من الدليل الحالي

`current directory`، ثم قم بالذهاب إلى الدليل `dict` عن طريق المسار النسبي:

```
cd ../dict
```

```
pwd
```

ويكون التنفيذ مشابه لما يلي:

```
sysadmin@localhost:/usr/share/doc$ cd ../dict
sysadmin@localhost:/usr/share/dict$ pwd
/usr/share/dict
sysadmin@localhost:/usr/share/dict$
```

تنفيذ الأمرين `cd ../dict` ، `pwd`

وهنا نلاحظ انه تم القيام بعملين في نفس سطر الأوامر وهما العودة مستوى واحد للأعلى ثم الذهاب عن طريق المسار النسبي إلى الدليل dict.

### ٦,٣ سرد الملفات والأدلة

في هذه المهمة سوف تتعلم كيفية سرد الملفات files والدلائل directories.

#### ٦,٣,١ الخطوة ١

لسرد محتويات الدليل الحالي current directory، استخدم الأمر ls:

```
cd
```

```
ls
```

ويكون التنفيذ مشابه لما يلي:

```
sysadmin@localhost:/usr/share/dict$ cd
sysadmin@localhost:~$ ls
Desktop Documents Downloads Music Pictures Public Templates Videos
sysadmin@localhost:~$
```

تنفيذ الأمرين cd، ls

عند تنفيذ الأمر ls تم وضع أسماء الدلائل باللون الأزرق الفاتح. وهي ميزة تلقائية توفرها العديد من توزيعات distributions لينكس من خلال ميزة تدعى الاسم المستعار alias. وهو ما سنراه في معاملة لاحقة.

يدل اللون على نوع هذا العنصر item. يصف الجدول التالي بعض الألوان الأكثر شيوعا في أنظمة لينكس:

نوع الملف	اللون
ملف عادي Regular file	اسود أو ابيض Black or White
ملف دليل Directory file	ازرق Blue
ملف ارتباط رمزي Symbolic link file (ملف يشير إلى ملف آخر)	سماوي Cyan
ملف قابل للتنفيذ Executable file مثل AKA program	اخضر Green

جدول يوضح أشهر الالوان في توزيعات لينكس

٢,٣,٢ الخطوة ٢

لا يتم عرض جميع الملفات files بشكل افتراضي by default. هناك ملفات تسمى الملفات المخفية hidden files، والتي لا يتم عرضها بشكل افتراضي. لعرض كافة الملفات بما في ذلك الملفات المخفية، استخدم الخيار -a مع الأمر ls:

```
ls -a
```

ويكون التنفيذ مشابه لما يلي:

```
sysadmin@localhost:~$ ls -a
.          .bashrc   .selected_editor  Downloads  Public
..         .cache    Desktop           Music      Templates
.bash_logout .profile  Documents         Pictures   Videos
sysadmin@localhost:~$
```

تنفيذ الأمر ls -a

الملفات المخفية hidden files تبدأ بنقطة dot ( . ) وعادة هذه الملفات والدلائل يتم إخفائها لأنها ملفات لا تُطلب عادة.

على سبيل المثال، ملف `bashrc`. المُبين في المثال أعلاه يحتوي على معلومات التكوين `configuration` `information` لقشرة الباش `bash shell`. فهذا الملف لا نحتاج إلى عرضه بشكل منظم.

هناك ملفين مهمين من الملفات ذات النقط `dot files` موجودة في كل دليل `directory`، الأول ذا نقطة واحدة وهو `( . )` ويمثل الدليل الحالي `current directory` اما الثاني فهو ذا نقطتين `( .. )` ويمثل الدليل الاعلى من الدليل الحالي.

٣,٣,٦ الخطوة ٣

الأمر `ls` في حد ذاته يوفر فقط أسماء الملفات والدلائل `files and directories names` داخل الدليل المحدد أو الحالي. نفذ الخيار التالي `l`- لمعرفة كيف يوفر مزيد من المعلومات حول الملف:

```
ls -l /etc/hosts
```

ويكون التنفيذ مشابه لما يلي:

```
sysadmin@localhost:~$ ls -l /etc/hosts
-rw-r--r-- 1 root root 150 Jan 22 15:18 /etc/hosts
sysadmin@localhost:~$
```

تنفيذ الأمر `ls -l /etc/hosts`

إنناً ما معنى كل المخرجات `output` السابقة؟

يقدم الجدول التالي وصفاً موجزاً لكل جزء من مخرجات الأمر `ls` مع الخيار `l`-:

معنى التنفيذ	تنفيذ ls -l
علامة -d تشير إلى العنصر هو ملف عادي plain file، أما إذا رأينا حرف d فإنه يشير إلى ان العنصر هو دليل directory.	علامة -d
هذا يمثل أذونات الملف permissions of the file.	rw-r--r--
وهذا يمثل ما يسمى عدد الارتباط الثابت hard link count.	1
المالك المستخدم للملف user owner of the file.	root
مجموعة المالكين للملف group owner of the file.	root
حجم الملف بالبايت size of the file in bytes.	189
تاريخ / وقت لأخر تعديل في ملف The date/time of last modified.	Dec 12 16:14

جدول يوضح تنفيذ ls -l

٤,٣,٤ الخطوة ٤

أحيانا كنت تريد أن ترى محتويات الدلائل الفرعية subdirectories وليس فقط محتويات الدليل. يمكنك استخدام الخيار -R لعمل ذلك كما يلي:

```
ls -R /etc/udev
```

ويكون التنفيذ مشابه لما يلي:

```
sysadmin@localhost:~$ ls -R /etc/udev
/etc/udev:
rules.d udev.conf
/etc/udev/rules.d:
70-persistent-cd.rules README
sysadmin@localhost:~$
```

تنفيذ الأمر ls -R /etc/udev

الخيار -R يشير إلى التكرار recursive. وبالتالي سيتم عرض كل الملفات في الدليل /etc/udev وكذلك كافة الملفات في الادلة الفرعية subdirectory، هنا الدليل الفرعي هو rules.d .

كن حذرا عند التعامل مع الخيار -R فبعض الدلائل كبيرة جدا!

٥,٣,٦ الخطوة

يمكنك استخدام رموز globbing (wildcards) للحد من عدد الملفات أو الدلائل. على سبيل المثال، يمكن علامة النجمة (\*) مطابقة صفر أو أكثر من أحرف اسم الملف. نفذ الأمر التالي لعرض الملفات التي تبدأ بالحرف s فقط في دليل etc:

```
ls -d /etc/s*
```

ويكون التنفيذ مشابه لما يلي:

```
sysadmin@localhost:~$ ls -d /etc/s*
/etc/securetty /etc/sgml /etc/shells /etc/ssl /etc/sysctl.conf
/etc/security /etc/shadow /etc/skel /etc/sudoers /etc/sysctl.d
/etc/services /etc/shadow- /etc/ssh /etc/sudoers.d /etc/systemd
sysadmin@localhost:~$
```

تنفيذ الأمر ls -d /etc/s\*

لاحظ أن الخيار -d يمنع عرض الملفات من الدلائل الفرعية subdirectories. وينبغي دائما أن تستخدم الأمر ls عند استخدام رموز globbing.

## ٦,٣,٦ الخطوة ٦

علامة الاستفهام ( ؟ ) يمكن استخدامها لمطابقة حرف واحد من اسم الملف. نفذ الأمر التالي لعرض كافة الملفات في الدليل /etc والمكونة من أربعة أحرف بالضبط:

```
ls -d /etc/????
```

ويكون التنفيذ مشابه لما يلي:

```
sysadmin@localhost:~$ ls -d /etc/????
/etc/bind /etc/init /etc/motd /etc/perl /etc/skel
/etc/dpkg /etc/ldap /etc/mtab /etc/sgml /etc/udev
sysadmin@localhost:~$
```

تنفيذ الأمر `ls -d /etc/????`

## ٦,٣,٧ الخطوة ٧

باستخدام الأقواس المربعة square brackets ( [ ] ) يمكنك تحديد حرف واحد للمطابقة من مجموعة من الأحرف. نفذ الأمر التالي لعرض كافة الملفات في الدليل /etc التي تبدأ بأحد الحروف: a, b, c, d:

```
ls -d /etc/[abcd]*
```

ويكون التنفيذ مشابه لما يلي:

```

sysadmin@localhost:~$ ls -d /etc/[abcd]*

/etc/adduser.conf      /etc/blkid.conf      /etc/cron.weekly
/etc/adjtime           /etc/blkid.tab       /etc/crontab
/etc/alternatives     /etc/ca-certificates /etc/dbus-1
/etc/apparmor.d       /etc/ca-certificates.conf /etc/debconf.conf
/etc/apt              /etc/calendar        /etc/debian_version
/etc/bash.bashrc      /etc/cron.d          /etc/default
/etc/bash_completion.d /etc/cron.daily      /etc/deluser.conf
/etc/bind             /etc/cron.hourly     /etc/depmod.d
/etc/bindresvport.blacklist /etc/cron.monthly   /etc/dpkg

sysadmin@localhost:~$

```

تنفيذ الأمر `ls -d /etc/[abcd]*`

## ٦,٤ نسخ ونقل وإعادة تسمية الملفات والدلائل

في هذه المهمة، سنقوم بنسخ `copy` ونقل `move` وإزالة `remove` الملفات والدلائل.

### ٦,٤,١ الخطوة ١

قم بعمل نسخة من ملف `/etc/hosts` وضعه في الدليل الحالي `current directory`. ثم قم بسرد `list` محتويات

الدليل الحالي `contents of the current directory` قبل وبعد نسخة:

```

ls

cp /etc/hosts hosts

ls

```

ويكون التنفيذ مشابه لما يلي:

```
sysadmin@localhost:~$ ls
Desktop Documents Downloads Music Pictures Public Templates Videos
sysadmin@localhost:~$ cp /etc/hosts hosts
sysadmin@localhost:~$ ls
Desktop Downloads Pictures Templates hosts
Documents Music Public Videos
sysadmin@localhost:~$
```

تنفيذ الأوامر ls ،cp /etc/hosts hosts ،ls

لاحظ كيف ان الأمر الثاني ls يعرض نسخة من ملف hosts.

٢,٤,٢ الخطوة

الآن سنقوم بإزالة الملف remove the file، ثم نسخه مرة أخرى، ولكن هنا النظام سيقوم بعرض الحدث. يتم ذلك باستخدام الخيار -v أو --verbose. قم بكتابة الأوامر التالية:

```
rm hosts
ls
cp -v /etc/hosts hosts
ls
```

لاحظ أنه يتم استخدام الأمر rm لحذف delete الملف. وسيتم توفير مزيد من المعلومات حول هذا الأمر في وقت لاحق هذا المعمل.

التنفيذ ينبغي أن يكون مشابه لما يلي:

```

sysadmin@localhost:~$ rm hosts

sysadmin@localhost:~$ ls

Desktop Documents Downloads Music Pictures Public Templates Videos

sysadmin@localhost:~$ cp -v /etc/hosts hosts

`/etc/hosts' -> `hosts'

sysadmin@localhost:~$ ls

Desktop      Downloads  Pictures   Templates  hosts
Documents   Music      Public     Videos
sysadmin@localhost:~$

```

تنفيذ الأوامر `ls`، `rm hosts`، `cp -v /etc/hosts hosts`، `ls`

لاحظ أن `-v` يبدل عرض المصدر `source` والهدف `target` عند تنفيذ الأمر `.cp`.

٣، ٤، ٦ الخطوة ٣

أدخل الأوامر التالية لنسخ الملف `/etc/hosts` وذلك باستخدام النقطة `period` (.) للإشارة إلى الدليل الحالي

`current directory` كهدف `target`:

```

rm hosts

ls

cp -v /etc/hosts .

ls

```

التنفيذ ينبغي أن يكون مشابه لما يلي:

```

sysadmin@localhost:~$ rm hosts
sysadmin@localhost:~$ ls
Desktop Documents Downloads Music Pictures Public Templates Videos
sysadmin@localhost:~$ cp -v /etc/hosts .
`/etc/hosts' -> `hosts'
sysadmin@localhost:~$ ls
Desktop Downloads Pictures Templates hosts
Documents Music Public Videos
sysadmin@localhost:~$

```

تنفيذ الأوامر `rm hosts`، `ls`، `cp -v /etc/hosts .`، `ls`

النقطة period ( . ) هي وسيلة سهلة الاستخدام للإشارة إلى الدليل الحالي current directory. ويمكن استخدامها مع جميع أوامر لينكس، وليس فقط مع الأمر `cp`.

٤, ٤, ٦ الخطوة ٤

أدخل الأوامر التالية للنسخ `copy` من الدليل المصدر source directory مع الحفاظ على سمات الملف file attributes باستخدام الخيار `-p`:

```

rm hosts

ls

cd /etc

ls -l hosts

cp -p hosts /home/sysadmin

cd

ls -l hosts

```

التنفيذ ينبغي أن يكون مشابه لما يلي:

```

sysadmin@localhost:~$ rm hosts
sysadmin@localhost:~$ ls
Desktop Documents Downloads Music Pictures Public Templates Videos
sysadmin@localhost:~$ cd /etc
sysadmin@localhost:/etc$ ls -l hosts
-rw-r--r-- 1 root root 150 Jan 22 15:18 hosts
sysadmin@localhost:/etc$ cp -p hosts /home/sysadmin
sysadmin@localhost:/etc$ cd
sysadmin@localhost:~$ ls -l hosts
-rw-r--r-- 1 sysadmin sysadmin 150 Jan 22 15:18 hosts
sysadmin@localhost:~$

```

تنفيذ الأوامر rm hosts ،ls ،cd /etc ،ls ،cp -p hosts /home/sysadmin ،ls

-l hosts

لاحظ أن التاريخ date الأذونات permission تم الحفاظ عليها كما هي. ملاحظة أخرى هي أن الطابع الزمني timestamp في التنفيذ أعلاه هو نفسه لكل من الأصل original والنسخة (١٢ ديسمبر ١٤:١٦).

٥،٤،٦ الخطوة ٥

اكتب الأوامر التالية للنسخ باستخدام أسماء اهدف مختلفة different target name:

```

rm hosts

cp -p /etc/hosts ~

cp hosts newname

ls -l hosts newname

rm hosts newname

```

التنفيذ ينبغي أن يكون مشابه لما يلي:

```
sysadmin@localhost:~$ rm hosts
sysadmin@localhost:~$ ls
Desktop Documents Downloads Music Pictures Public Templates Videos
sysadmin@localhost:~$ cp -p /etc/hosts ~
sysadmin@localhost:~$ cp hosts newname
sysadmin@localhost:~$ ls -l hosts newname
-rw-r--r-- 1 sysadmin sysadmin 150 Jan 22 15:18 hosts
-rw-r--r-- 1 sysadmin sysadmin 150 Jan 22 16:29 newname
sysadmin@localhost:~$ rm hosts newname
sysadmin@localhost:~$
```

تنفيذ الأوامر `rm hosts`، `cp -p /etc/hosts ~`، `cp hosts newname`، `ls -l hosts newname`، `rm hosts newname`، `e`

النسخة الأولى من الخيار `-p` حافظت على الطابع الزمني الأصلي `original timestamp`. تذكر أن علامة تيلدا (`~`) تمثل الدليل الرئيسي `./home/sysadmin`.

النسخة الثانية حددت اسم ملف مختلف `newname` كهدف `target`. وذلك لعدم استخدام الخيار `-p`، النظام استخدم التاريخ والوقت الحالي للهدف `target`، وبالتالي فإنه لم يتم الحفاظ على الطابع الزمني الأصلي والموجود في الملف المصدر `./etc/hosts`.

وأخيراً، لاحظ أنه يمكنك إزالة `remove` أكثر من ملف في وقت واحد كما هو مبين مع الأمر `rm` الأخير، حيث تم إزالة ملفين بنفس السطر.

٦، ٤، ٦ الخطوة ٦

استخدام الخيار `-R` لنسخ جميع الملفات في الدليل. لهذه المهمة سوف ننسخ الدليل `/etc/udev` ونعرض محتويات الدليل المنسوخ `copied directory`:

`mkdir Myetc`

`cp -R /etc/udev Myetc`

`ls -l Myetc`

`ls -lR Myetc`

التنفيذ ينبغي أن يكون مشابه لما يلي:

```

sysadmin@localhost:~$ mkdir Myetc
sysadmin@localhost:~$ cp -R /etc/udev Myetc
sysadmin@localhost:~$ ls -l Myetc
total 0
drwxr-xr-x 1 sysadmin sysadmin 32 Jan 22 16:35 udev
sysadmin@localhost:~$ ls -lR Myetc
Myetc:
total 0
drwxr-xr-x 1 sysadmin sysadmin 32 Jan 22 16:35 udev

Myetc/udev:
total 4

drwxr-xr-x 1 sysadmin sysadmin 56 Jan 22 16:35 rules.d
-rw-r--r-- 1 sysadmin sysadmin 218 Jan 22 16:35 udev.conf

Myetc/udev/rules.d:
total 8
-rw-r--r-- 1 sysadmin sysadmin 306 Jan 22 16:35
70-persistent-cd.rules
-rw-r--r-- 1 sysadmin sysadmin 1157 Jan 22 16:35 README
sysadmin@localhost:~$

```

تنفيذ الأوامر `ls -lR Myetc` ، `ls -l Myetc` ، `cp -R /etc/udev Myetc` ، `mkdir Myetc`

## ٦, ٤, ٧ الخطوة ٧

استخدم الخيار r - لإزالة remove دليل ما مع الأمر :rm

```
ls

rm -r Myetc

ls
```

التنفيذ ينبغي أن يكون مشابه لما يلي:

```
sysadmin@localhost:~$ ls
Desktop  Downloads  Myetc      Public     Videos
Documents Music       Pictures   Templates

sysadmin@localhost:~$ rm -r Myetc

sysadmin@localhost:~$ ls
Desktop Documents Downloads Music Pictures Public Templates Videos

sysadmin@localhost:~$
```

تنفيذ الأوامر ls ، rm -r Myetc ، ls

لاحظ أن الأمر rmdir يمكن أن يستخدم أيضا لحذف الدلائل delete directories، ولكن فقط إذا كان الدليل فارغ empty اي انه لا يحتوي على أية ملفات.

نلاحظ أيضا ان الخيار r - يزيل الدلائل directories ومحتوياتها contents بشكل متكرر.

## ٦, ٤, ٨ الخطوة ٨

ان عملية نقل ملف مشابهة لعمليتي القص واللصق cut and paste. القص يكون للملف من الموقع الأصلي original location ثم اللصق إلى الوجهة المحددة specified destination. نقل ملف من الدليل المحلي local directory يتم من خلال تنفيذ الأوامر التالية:

touch premove

ls

mv premove postmove

ls

rm postmove

الجدول التالي يوضح وصف ما يحدث للأوامر السابقة:

الوصف	امر لينكس
يقوم هذا الأمر بتحديث الملف premove فإن لم يجد أي ملف بذلك الاسم، فإنه يقوم بإنشاء ملف فارغ بنفس الاسم premove	touch premove
هذا الأمر بقص الملف premove ثم لصقه إلى ملف اخر والمسمى postmove	mv premove postmove
ازالة الملف postmove	rm postmove

الجدول يصف أوامر المثال السابق

التنفيذ ينبغي أن يكون مشابه لما يلي:

```
sysadmin@localhost:~$ touch premove
sysadmin@localhost:~$ ls
Desktop    Downloads  Pictures    Templates  premove
Documents  Music      Public     Videos
sysadmin@localhost:~$ mv premove postmove
sysadmin@localhost:~$ ls
Desktop    Downloads  Pictures    Templates  postmove
Documents  Music      Public     Videos
sysadmin@localhost:~$
```

تنفيذ الأوامر rm postmove ،ls ،mv premove postmove ،ls ،touch premove

## المعمل السابع – الأرشفة والضغط

## Archiving and Compression

## ٧,١ مقدمة

هذا هو المعمل السابع: أرشفة وضغط الملفات ثم فك الارشفة والضغط. عن طريق أداء هذا التمرين، سوف تتعلم كيفية التعامل مع ملفات الأرشيف.

في هذا التمرين، سوف نقوم بالمهام التالية:

- إنشاء ملفات الأرشيف باستخدام الأمر tar مع الضغط وبدونه.
- ضغط وفك ضغط الملفات مع الأمر gzip.
- ضغط وفك ضغط الملفات مع الأمر BZIP2.
- استخدام الأمرين zip و unzip لضغط الملفات ثم فك الضغط.

## ٧,٢ أوامر الأرشفة

في هذه المهمة، سوف نستخدم الأوامر gzip و bzip2 و zip/unzip بغرض أرشفة archive واستعادة الملفات restore files. وقد صممت هذه الأوامر لوظيفتين رئيسيتين؛ الأولى دمج ملفات متعددة merge multiple files في ملف واحد، اما الوظيفة الثانية فهي ضغط compress الملفات الكبيرة إلى ملفات أصغر. وفي بعض الحالات سوف تؤدي الوظيفتين معاً.

أرشفة البيانات مهمة لعدة أسباب منها:

- ❖ صعوبة نقل transfer الملفات الكبيرة، فجعل حجم هذه الملفات أصغر يساعد في تسريع عملية النقل.
- ❖ نقل ملفات متعددة multiple files من نظام إلى آخر يصبح مزعجاً عندما يكون هناك العديد من الملفات. فعملية دمجها في ملف واحد للنقل transport يجعل هذه العملية أسهل.
- ❖ الملفات يمكن أن تحتل وبسرعة مساحة space كبيرة جداً، خاصةً على الوسائط الصغيرة القابلة للإزالة removable media مثل محركات أقراص الإبهام thumb drives. لذلك فإن عملية الأرشفة Archiving تقلل reduces من هذه المشكلة.

نقطة واحدة قد تكون مربكة للمستخدمين الجدد لنظام لينكس وهي لماذا يوجد الكثير من أوامر الأرشفة archiving commands المختلفة؟ الجواب هو أن هذه الأوامر لها مميزات features مختلفة عن بعضها البعض، على سبيل المثال؛ بعض من أوامر الأرشفة تسمح لك بحماية protect ملف الأرشيف archive file عن طريق كتابة كلمة سر password. أيضا هناك اختلاف في تقنيات الضغط compression techniques المستخدمة.

المهم أن نتعرف على وظيفة function كل منها، ومع مرور الوقت سوف نتعلم اختيار أداة الأرشفة archive tool الصحيحة للحالات التي قد تواجهك.

١، ٢، ٧ الخطوة ١

استخدم الأمر tar لإنشاء أرشيف في الدليل /etc/udev. احفظ نسخة احتياطية backup في الدليل  
~/mybackups:

```
cd  
  
mkdir mybackups  
  
tar -cvf mybackups/udev.tar /etc/udev  
  
ls mybackups
```

التنفيذ ينبغي أن يكون مشابه لما يلي:

```

sysadmin@localhost:~$ cd
sysadmin@localhost:~$ mkdir mybackups
sysadmin@localhost:~$ tar -cvf mybackups/udev.tar /etc/udev
tar: Removing leading `/' from member names
/etc/udev/
/etc/udev/rules.d/
/etc/udev/rules.d/70-persistent-cd.rules
/etc/udev/rules.d/README
/etc/udev/udev.conf
sysadmin@localhost:~$ ls mybackups/
udev.tar
sysadmin@localhost:~$

```

تنفيذ الأوامر `cd` ، `mkdir mybackups` ، `tar -cvf mybackups/udev.tar /etc/udev` ، `ls mybackups`

تم استخدام الأمر `tar` لدمج الملفات المتعددة `merge multiple files` في ملف واحد. لاحظ ان الأمر السابق لا يقوم بعملية ضغط `compress` للبيانات `data` بل أرشفتها فقط.

الخيار `-c` يُستخدم مع الأمر `tar` لإنشاء ملف الأرشفة `tar file`. بينما الخيار `-v` فهو "verbose" حيث يقوم بإظهار الحدث. بينما يتم استخدام الخيار `-f` لتحديد اسم ملف `tar file`.

لمعلوماتك: `tar` يرمز لأرشيف الشريط `Tape ARchive`. وقد استخدم هذا الأمر أصلاً لإنشاء نسخ احتياطية للأشرطة `tape backups`، ولكن اليوم يتم استخدامه لإنشاء ملفات الأرشيف `archive files`.

ليس هناك حاجة لاستخدام الملحق `extension (tar.)` لأرشفة اسم ملف `archive file name` ، ومع ذلك فإنه يساعد على تحديد نوع الملف `file type`. حيث يعتبر اسلوب جيد وذلك عند إرسال ملف الأرشيف `archive file` إلى شخص آخر.

## ٧,٢,٢ الخطوة ٢

اعرض محتويات ملف tar الذي سبق وقمت بإنشائه سابقاً، لاحظ معاني الخيارات التالية والمصاحبة للأمر tar:

- t : لعرض قائمة المحتويات list contents
- v : لإظهار الحدث verbose
- f : لتحديد اسم الملف filename

```
tar -tvf mybackups/udev.tar
```

التنفيذ ينبغي أن يكون مشابه لما يلي:

```
sysadmin@localhost:~$ tar -tvf mybackups/udev.tar
drwxr-xr-x root/root      0 2015-01-28 16:32 etc/udev/
drwxr-xr-x root/root      0 2015-01-28 16:32 etc/udev/rules.d/
-rw-r--r-- root/root    306 2015-01-28 16:32
                               etc/udev/rules.d/70-persistent-cd.rules
-rw-r--r-- root/root   1157 2012-04-05 19:18 etc/udev/rules.d/README
-rw-r--r-- root/root    218 2012-04-05 19:18 etc/udev/udev.conf
sysadmin@localhost:~$
```

تنفيذ الأمر tar -tvf mybackups/udev.tar

لاحظ أن الملفات تم نسخها احتياطياً backed up بشكل متكرر recursively وذلك باستخدام المسار النسبي relative path names . وهو مهم جداً عند استخراج extract تلك الملفات، لأنه سيتم وضعها في دليلك الحالي your current directory، وليس فوق override الملفات الحالية current files.

## ٧,٢,٣ الخطوة ٣

لأنشاء ملف tar مضغوط استخدم الخيار z - مع الامر السابق:

```
tar -zcvf mybackups/udev.tar.gz /etc/udev
```

```
ls -lh mybackups
```

التنفيذ ينبغي أن يكون مشابه لما يلي:

```
sysadmin@localhost:~$ tar -zcvf mybackups/udev.tar.gz /etc/udev
tar: Removing leading `/' from member names
/etc/udev/
/etc/udev/rules.d/
/etc/udev/rules.d/70-persistent-cd.rules
/etc/udev/rules.d/README
/etc/udev/udev.conf
sysadmin@localhost:~$ ls -lh mybackups/
total 16K
-rw-rw-r-- 1 sysadmin sysadmin 10K Jan 25 04:00 udev.tarf
-rw-rw-r-- 1 sysadmin sysadmin 1.2K Jan 25 04:34 udev.tar.gz
sysadmin@localhost:~$
```

تنفيذ الأمرين `ls -lh mybackups` ، `tar -zcvf mybackups/udev.tar.gz /etc/udev`

لاحظ الفرق في الحجم؛ حيث ان النسخة الاحتياطية backup الأولى ١٠ كيلوبايت وبالتالي أكبر من النسخة الاحتياطية backup الثانية وهي ١,٢ كيلوبايت فقط.

الخيار -Z- يستفيد من مجموعة الأمر `gzip utility` وذلك للقيام بعمل الضغط.

٤,٢,٧ الخطوة ٤

قم باستخراج Extract محتويات الأرشيف `archive contents`. لاحظ ان البيانات `data` يعاد حفظها `restored` في الدليل الحالي `current directory` بشكل افتراضي:

```
cd mybackups

ls

tar -xvf udev.tar.gz

ls etc

ls etc/udev

ls etc/udev/rules.d
```

التنفيذ ينبغي أن يكون مشابه لما يلي:

```
sysadmin@localhost:~$ cd mybackups
sysadmin@localhost:~/mybackups$ ls
udev.tar  udev.tar.gz
sysadmin@localhost:~/mybackups$ tar -xvf udev.tar.gz
etc/udev/
etc/udev/rules.d/
etc/udev/rules.d/70-persistent-cd.rules
etc/udev/rules.d/README
etc/udev/udev.conf
sysadmin@localhost:~/mybackups$ ls
etc  udev.tar  udev.tar.gz
sysadmin@localhost:~/mybackups$ ls etc
udev
sysadmin@localhost:~/mybackups$ ls etc/udev
rules.d  udev.conf
sysadmin@localhost:~/mybackups$ ls etc/udev/rules.d
70-persistent-cd.rules  README
sysadmin@localhost:~/mybackups$
```

تنفيذ الأوامر 'cd mybackups' ، 'tar -xvf udev.tar.gz' ، 'ls' ، 'ls etc' ، 'ls etc/udev' ، 'ls etc/udev'

v/rules.d

إذا أردت ان تعيد الملفات إلى موقعها الأصلي original location، فيجب أولاً كتابة الأمر cd إلى الدليل / ثم تشغيل الأمر tar، في هذا المثال سيطلب منك أن تكون مسجلاً logged in كمسؤول للنظام، لإنشاء ملفات creating files في الدليل /etc directory لا يتم إلا عن طريق مسؤول النظام administrator.

٥، ٢، ٧ الخطوة

لإضافة ملف إلى أرشيف موجود استخدم الخيار r - مع أمر tar. اكتب الأوامر التالية لتنفيذ هذا الإجراء والتحقق من وجود ملف جديد في أرشيف tar:

```
tar -rvf udev.tar /etc/hosts
```

```
tar -tvf udev.tar
```

التنفيذ ينبغي أن يكون مشابه لما يلي:

```
sysadmin@localhost:~/mybackups$ tar -rvf udev.tar /etc/hosts
tar: Removing leading `/' from member names
/etc/hosts
sysadmin@localhost:~/mybackups$ tar -tvf udev.tar
drwxr-xr-x root/root      0 2015-01-28 16:32 etc/udev/
drwxr-xr-x root/root      0 2015-01-28 16:32 etc/udev/rules.d/
-rw-r--r-- root/root    306 2015-01-28 16:32
etc/udev/rules.d/70-persistent-cd.rules
-rw-r--r-- root/root   1157 2012-04-05 19:18 etc/udev/rules.d/README
-rw-r--r-- root/root    218 2012-04-05 19:18 etc/udev/udev.conf
-rw-r--r-- root/root    172 2017-08-21 23:28 etc/hosts
sysadmin@localhost:~/mybackups$
```

تنفيذ الأوامر tar -rvf udev.tar /etc/hosts، tar -tvf udev.tar

## ٦, ٢, ٧ الخطوة ٦

في الأمثلة التالية سوف نستخدم امري `gzip` و `gunzip` لضغط `compress` وفك الضغط `uncompress` عن ملف. نفذ الأوامر التالية لضغط نسخة من الملف `words`:

```
cp /usr/share/dict/words .
```

```
ls -l words
```

```
gzip words
```

```
ls -l words.gz
```

التنفيذ ينبغي أن يكون مشابه لما يلي:

```
sysadmin@localhost:~/mybackups$ cp /usr/share/dict/words .
sysadmin@localhost:~/mybackups$ ls -l words
-rw-r--r-- 1 sysadmin sysadmin 938848 Jan 25 07:39 words
sysadmin@localhost:~/mybackups$ gzip words
sysadmin@localhost:~/mybackups$ ls -l words.gz
-rw-r--r-- 1 sysadmin sysadmin 255996 Jan 25 07:39 words.gz
sysadmin@localhost:~/mybackups$
```

تنفيذ الأوامر `cp /usr/share/dict/words .`، `ls -l words`، `gzip words`، `ls -l words.gz`

لاحظ أن حجم `size` الملف المضغوط `zipped file` يساوي ٢٥٥٩٩٦ بايت في المثال أعلاه وهو أصغر بكثير من الملف الأصلي ٩٣٨٨٤٨.

هام جدا: عند استخدام `gzip`، يتم استبدال الملف الأصلي بواسطة ملف مضغوط. في المثال أعلاه، تم استبدال الملف `words` بالملف `words.gz`.

مهم جدا: عندما تقوم بفك ضغط الملف unzip، سيتم استبدال الملف المضغوط zipped file مع الملف الأصلي original file.

٧,٢,٧ الخطوة ٧

نفذ الأوامر التالية لفك ضغط الملف words.gz:

```
ls -l words.gz
```

```
gunzip words.gz
```

```
ls -l words
```

التنفيذ ينبغي أن يكون مشابه لما يلي:

```
sysadmin@localhost:~/mybackups$ ls -l words.gz
-rw-r--r-- 1 sysadmin sysadmin 255996 Jan 25 07:39 words.gz
sysadmin@localhost:~/mybackups$ gunzip words.gz
sysadmin@localhost:~/mybackups$ ls -l words
-rw-r--r-- 1 sysadmin sysadmin 938848 Jan 25 07:39 words
sysadmin@localhost:~/mybackups$
```

تنفيذ الأوامر ls -l words، gunzip words.gz، ls -l words.

لينكس يوفر عددا كبيرا من مرافق الضغط compression utilities بالإضافة إلى gzip/gunzip. كل منهم له إيجابياته وسلبياته pros and cons مثلًا ضغط أسرع faster compression ومعدلات ضغط أفضل better compression rates وأكثر مرونة more flexible وقابلة للحمل more portable وأسرع بفك الضغط decompression.

الأمران gzip/gunzip تحظى بشعبية كبيرة في لينكس. بينما الأمران BZIP2/ bunzip2 لهما شعبيتهما الكبيرة أيضاً ولكن بعض توزيعات لينكس، ومن حسن الحظ أن معظم وظائفها كطريقة تشغيل run الأوامر هي نفس الأمران السابقان gzip/gunzip.

٨، ٢، ٧ الخطوة ٨

ان استخدام تقنية BZIP2 و bunzip2 لضغط compress وفك الضغط uncompress يشبه استخدام gzip و gunzip. حيث يتم إنشاء ملف مضغوط مع وصلة .bz2 extension، ثم تتم إزالة الوصلة extension عند فك الضغط. نفذ الأوامر التالية لضغط نسخة من ملف words:

```
ls -l words
```

```
bzip2 words
```

```
ls -l words.bz2
```

التنفيذ ينبغي أن يكون مشابه لما يلي:

```
sysadmin@localhost:~/mybackups$ ls -l words
-rw-r--r-- 1 sysadmin sysadmin 938848 Jan 25 07:39 words
sysadmin@localhost:~/mybackups$ bzip2 words
sysadmin@localhost:~/mybackups$ ls -l words.bz2
-rw-r--r-- 1 sysadmin sysadmin 335405 Jan 25 07:39 words.bz2
sysadmin@localhost:~/mybackups$
```

تنفيذ الأوامر ls -l words ، bzip2 words ، ls -l words.bz2

إذا قارنت حجم ملف .bz2. الناتج 335405 مع حجم ملف .gz. الناتج من الخطوة رقم ٧، ستلاحظ أن gzip قام بعمل أفضل عند ضغط هذا الملف.

٩، ٢، ٧ الخطوة

نفذ الأوامر التالية لفك ضغط `uncompress` الملف `words.bz2`:

```
ls -l words.bz2

bunzip2 words.bz

ls -l words
```

التنفيذ ينبغي أن يكون مشابه لما يلي:

```
sysadmin@localhost:~/mybackups$ ls -l words.bz2
-rw-r--r-- 1 sysadmin sysadmin 335405 Jan 25 07:39 words.bz2

sysadmin@localhost:~/mybackups$ bunzip2 words.bz2

sysadmin@localhost:~/mybackups$ ls -l words
-rw-r--r-- 1 sysadmin sysadmin 938848 Jan 25 07:39 words
```

تنفيذ الأوامر `ls -l words`، `bunzip2 words.bz2`، `ls -l words.bz2`

تعتبر ملفات الأرشفة `gzip` و `bzip` هي الأكثر شيوعاً في عالم لينكس، لكن يتم استخدام نوع أرشفة مختلف في أنظمة التشغيل الأخرى مثل ويندوز. ولقد تم دعم `support` تطبيق ويندوز إكسبلورر Windows Explorer ليتمكن من استخراج ملفات الأرشيف `zip archive files`.

لذلك، إذا كنت تخطط لمشاركة أرشيف `archive` مع مستخدم ويندوز، يفضل استخدام `zip`. وعلى خلاف `gzip` و `bzip2`، فإن الأمر `zip` عند ضغط ملف ما فإنه يقوم بعمل نسخة من الملف الأصلي `original file` وهي التي يتم ضغطها؛ أما الأصل فيبقى كما هو وبدون أي ضغط.

١٠، ٢، ٧ الخطوة

استخدم الأمر `zip` لضغط ملف `words`:

```
zip words.zip words
```

```
ls -l words.zip
```

التنفيذ ينبغي أن يكون مشابه لما يلي:

```
sysadmin@localhost:~/mybackups$ zip words.zip words
  adding: words (deflated 73%)
sysadmin@localhost:~/mybackups$ ls -l words.zip
-rw-rw-r-- 1 sysadmin sysadmin 256132 Jan 25 21:25 words.zip
sysadmin@localhost:~/mybackups$
```

تنفيذ الأمرين zip words.zip words ، ls -l words.zip

في المثال أعلاه الجزء words.zip من الأمر zip يعني اسم الملف الذي ترغب في إنشائه. اما الملف words فهو الذي تريد وضعه في ملف مضغوط.

مهم: لا حاجة لاستخدام zip extension. مع اسم ملف مضغوط. ومع ذلك فإنه من مفيد لتحديد نوع الملف. وهو يعتبر أيضا اسلوب جيد عند إرسال ملف الأرشيف إلى شخص آخر.

١١، ٢، ٧ الخطوة ١١

اضغط الدليل /etc/udev ومحتوياته مع الأمر zip:

```
zip -r udev.zip /etc/udev
```

```
ls -l udev.zip
```

التنفيذ ينبغي أن يكون مشابه لما يلي:

```

sysadmin@localhost:~/mybackups$ zip -r udev.zip /etc/udev
  adding: etc/udev/ (stored 0%)
  adding: etc/udev/rules.d/ (stored 0%)
  adding: etc/udev/rules.d/70-persistent-cd.rules (deflated 29%)
  adding: etc/udev/rules.d/README (deflated 50%)
  adding: etc/udev/udev.conf (deflated 24%)
sysadmin@localhost:~/mybackups$ ls -l udev.zip
-rw-rw-r-- 1 sysadmin sysadmin 1840 Jan 25 21:33 udev.zip
sysadmin@localhost:~/mybackups$

```

تنفيذ الأمرين `ls -l udev.zip` و `zip -r udev.zip /etc/udev`

الأمر `tar` ينتقل `descends` تلقائياً خلال الدلائل الفرعية للدليل المحدد ليتم أرشفتها، كما هو الحال في المثال السابق `/etc/udev`، بينما عند التعامل مع الأوامر `BZIP2` و `GZIP` و `zip` يجب تحديد الخيار `-r` من أجل التحرك `recursion` إلى الدلائل الفرعية `.subdirectories`.

١٢، ٢، ٧ الخطوة

لعرض محتويات ملف مضغوط `zip`، استخدام الخيار `-l` مع الأمر `unzip`:

```
unzip -l udev.zip
```

التنفيذ ينبغي أن يكون مشابه لما يلي:

```

sysadmin@localhost:~/mybackups$ unzip -l udev.zip
Archive:  udev.zip
  Length      Date    Time    Name
-----
         0  2015-01-28  16:32  etc/udev/
         0  2015-01-28  16:32  etc/udev/rules.d/
        306  2015-01-28  16:32  etc/udev/rules.d/70-persistent-cd.rules
       1157  2012-04-05  19:18  etc/udev/rules.d/README
        218  2012-04-05  19:18  etc/udev/udev.conf
-----
       1681
                               5 files
sysadmin@localhost:~/mybackups$

```

تنفيذ الأوامر `unzip -l udev.zip`

١٣، ٢، ٧ الخطوة ١٣

لاستخراج أرشيف ملف من نوع `zip`، استخدم الأمر `unzip` دون أي خيارات `options`. في هذا المثال نحن بحاجة أولاً لحذف الملفات `tar` التي تم إنشاؤها في المثال السابق:

```
rm -r etc
```

```
unzip udev.zip
```

التنفيذ ينبغي أن يكون مشابه لما يلي:

```
sysadmin@localhost:~/mybackups$ rm -r etc
sysadmin@localhost:~/mybackups$ unzip udev.zip
Archive:  udev.zip
  creating:  etc/udev/
  creating:  etc/udev/rules.d/
 inflating:  etc/udev/rules.d/70-persistent-cd.rules
 inflating:  etc/udev/rules.d/README
 inflating:  etc/udev/udev.conf
sysadmin@localhost:~/mybackups$
```

تنفيذ الأمرين 'rm -r etc' ، 'unzip udev.zip'

## المعمل الثامن – القنوات القياسية والأنابيب وإعادة

### التوجيه

## Pipes, Redirection and REGEX

## ٨,١ مقدمة

هذا هو المعمل الثامن: القنوات القياسية و الأنابيب وإعادة التوجيه. عن طريق أداء هذا التمرين، سوف نتعلم كيفية إعادة توجيه تيارات النص، واستخدام التعبيرات العادية، واستخدام أوامر الملفات لتصفية النص.

في هذا التمرين، سوف نقوم بالمهام التالية:

- تعلم كيفية إعادة توجيه القنوات القياسية (المدخلات والمخرجات والاطفاء).
- استخدام التعبيرات العادية لتصفية مخرجات الأوامر أو محتويات الملفات.
- عرض الملفات الكبيرة كاملة أو استخدام أوامر الترحيل لعرض أجزاء مختارة منها.

## ٨,٢ قنوات سطر الأوامر وإعادة التوجيه

عادة عند تنفيذ الأوامر يتم عرض الناتج في نافذة terminal. وهذا الناتج output (وأيضا يسمى قناة channel) يطلق عليه التنفيذ القياسي Standard output، ويشار اليه بالرمز (stdout). اما القناة المعبرة عنه فهي القناة رقم ١.

الخطأ القياسي Standard error (stderr) ويظهر عند حدوث أي خطأ أثناء تنفيذ الأوامر. اما القناة المعبرة عنه فهي القناة رقم ٢ ، وبشكل افتراضي يتم اخراج رسائل الخطأ بواسطتها إلى نافذة terminal.

واخيراً الإدخال القياسي Standard input (stdin) وهو عادة ما نقوم بإدخاله من أوامر عن طريق الكتابة على لوحة المفاتيح، اما القناة المعبرة عنه فهي القناة رقم ٠. ومع ذلك ومن خلال إعادة توجيه redirecting الإدخال القياسي Standard input، الملفات يمكن أن تستخدم أيضا كـ stdin.

في هذا التمرين سوف نستخدم الأحرف characters التي تعيد توجيه redirect المخرجات output من التنفيذ القياسي (stdout) standard output والخطأ القياسي (stderr) standard error إلى ملف أو أمر آخر بدلا من نافذة terminal.

## ١,٢,٨ الخطوة ١

يُستخدم رمز إعادة التوجيه `redirect` > لإعادة توجيه التنفيذ `output` من الناتج الافتراضي `stdout` لنافذة `terminal` إلى ملف. اكتب ما يلي لتحقيق ذلك:

```
echo "Hello World"
echo "Hello World" > mymessage
cat mymessage
```

التنفيذ ينبغي أن يكون مشابه لما يلي:

```
sysadmin@localhost:~$ echo "Hello World"
Hello World
sysadmin@localhost:~$ echo "Hello World" > mymessage
sysadmin@localhost:~$ cat mymessage
Hello World
sysadmin@localhost:~$
```

تنفيذ الأوامر الثلاثة `echo "Hello World" > mymessage` ، `echo "Hello World"` ، `cat mymessage`

- الأمر الأول وهو `echo "Hello World"` يقوم بإخراج الجملة السابقة على الإخراج القياسي `stdout` لتظهر في النافذة الطرفية `terminal`.
- أما الأمر الثاني `echo "Hello World" > mymessage` فهو يعيد توجيه التنفيذ؛ فبدلاً من إرساله إلى `terminal`، يتم إرسال التنفيذ إلى ملف يسمى `mymessage`.
- وأخيراً يعرض الأمر `cat mymessage` محتويات ملف `mymessage`.

## ٢,٢,٨ الخطوة ٢

عند استخدام رمز > لإعادة توجيهه stdout، فإن محتويات الملف السابقة تتم ازلتها، اكتب الأوامر التالية لمشاهدة عرض توضيحي:

```
cat mymessage

echo Greetings > mymessage

cat mymessage
```

التنفيذ ينبغي أن يكون مشابه لما يلي:

```
sysadmin@localhost:~$ cat mymessage
Hello World

sysadmin@localhost:~$ echo Greetings > mymessage

sysadmin@localhost:~$ cat mymessage
Greetings

sysadmin@localhost:~$
```

تنفيذ الأوامر الثلاثة cat mymessage ، echo "How are you?" >> mymessage ، cat mymessage  
ymessage

لاحظ أننا استخدمنا رمز إعادة توجيه واحد " > " للكتابة فوق ملف موجود. وهذا ما يسمى بإتلاف ملف  
.lobbering

## ٣,٢,٨ الخطوة ٣

يمكنك تجنب إتلاف ملف clobbering باستخدام رمزي ">>" بدلا من ">". فباستخدام رمزي إعادة التوجيه يتم إلحاق append البيانات الجديدة إلى الملف دون إزالة أو إتلاف للمحتويات السابقة. نفذ الأوامر التالية لمشاهدة عرض توضيحي لهذا:

```
cat mymessage
```

```
echo "How are you?" >> mymessage
```

```
cat mymessage
```

التنفيذ ينبغي أن يكون مشابه لما يلي:

```
sysadmin@localhost:~$ cat mymessage
Greetings
sysadmin@localhost:~$ echo "How are you?" >> mymessage
sysadmin@localhost:~$ cat mymessage
Greetings
How are you?
sysadmin@localhost:~$
```

تنفيذ الأوامر الثلاثة cat m ، echo "How are you?" >> mymessage ،cat mymessage ymessage

لاحظ أنه باستخدام " >> " يتم الاحتفاظ بكافة البيانات الموجودة ويتم إلحاق البيانات الجديدة في نهاية الملف.

٤, ٢, ٨ الخطوة

يستخدم الأمر find للبحث في ملفات النظام filesystem عن ملف ما استنادا إلى أحد المعايير criteria مثل اسم الملف filename. وسنستخدمه الآن لإظهار كيفية عمل stderr، قم بتشغيل الأمر التالي ولاحظ التنفيذ:

```
find /etc -name hosts
```

التنفيذ ينبغي أن يكون مشابه لما يلي:

```
sysadmin@localhost:~$ find /etc -name hosts
find: `/etc/ssl/private': Permission denied
/etc/hosts
sysadmin@localhost:~$
```

تنفيذ الأمر `find /etc -name hosts`

لاحظ رسالة الخطأ `error message` تشير إلى أنك لا تمتلك الصلاحية `permission` للدخول `access` لملف أو دليل ما. وذلك لأن المستخدم العادي `regular user` ليس لديه الحق في النظر داخل بعض الدلائل. ويتم إرسال هذه الأنواع من رسائل الخطأ إلى `stderr`، وليس إلى `stdout`.

ستتم تغطية أمر البحث `find` بمزيد من التفاصيل في وقت لاحق. الآن نستخدمه فقط لإظهار الفرق بين الإخراج القياسي `stdout` والخطأ القياسي `stderr`.

٥, ٢, ٨ الخطوة ٥

لإعادة توجيهه `stderr` (رسائل الخطأ) إلى ملف ما، نفذ الأمر التالي:

```
find /etc -name hosts 2> err.txt
cat err.txt
```

وينبغي أن يكون التنفيذ مشابه لما يلي:

```
sysadmin@localhost:~$ find /etc -name hosts 2> err.txt
/etc/hosts
sysadmin@localhost:~$ cat err.txt
find: `/etc/ssl/private': Permission denied
sysadmin@localhost:~$
```

تنفيذ الأمرين `cat err.txt` و `find /etc -name hosts 2> err.txt`

تذكر أن القناة الواصفة file descriptor لرسائل الخطأ stderr هي القناة رقم ٢، لذلك يتم إستخدامها جنباً إلى جنب مع رمز " > " لإعادة توجيه redirect تنفيذ stderr إلى ملف يسمى err.txt . لاحظ أن الرمز " > 1 " هو نفسه " > " .

ملاحظة: يوضح المثال السابق لماذا فهم عملية إعادة التوجيه مهمة جداً. فعلى سبيل المثال؛ إذا كنت ترغب بتجاهل الأخطاء errors التي يعرضها الأمر find حالياً فيمكنك إعادة توجيه redirect تلك الرسائل messages في ملف ما والنظر إليها في وقت لاحق، مما يجعل من الأسهل أن تركز على ما تبقى من إخراج الأمر.

٦,٢,٨ الخطوة ٦

يمكنك أيضا إعادة توجيه الإخراج القياسي stdout و رسائل الخطأ stderr إلى ملفين منفصلين:

```
find /etc -name hosts > std.out 2> std.err
cat std.err
cat std.out
```

وينبغي أن يكون التنفيذ مشابه لما يلي:

```
sysadmin@localhost:~$ find /etc -name hosts > std.out 2> std.err
sysadmin@localhost:~$ cat std.err
find: `/etc/ssl/private': Permission denied
sysadmin@localhost:~$ cat std.out
/etc/hosts
sysadmin@localhost:~$
```

تنفيذ الأوامر cat std.out ،cat std.err ،find /etc -name hosts > std.out 2> std.err

لاحظ أن المسافة مسموحة ولكنها ليست مطلوبة بعد رمز إعادة التوجيه " > " .

٧, ٢, ٨ الخطوة ٧

لإعادة توجيهه redirect كل من الإخراج القياسي (standard output (stdout) وايضاً رسائل الخطأ (standard error (stderr) إلى ملف واحد، قم أولاً بإعادة توجيهه stdout إلى ملف ثم بعدها أعد توجيهه stderr إلى نفس الملف باستخدام &1>2.

```
find /etc -name hosts >find.out 2>&1
```

```
cat find.out
```

وينبغي أن يكون التنفيذ مشابه لما يلي:

```
sysadmin@localhost:~$ find /etc -name hosts > find.out 2>&1
sysadmin@localhost:~$ cat find.out
find: `/etc/ssl/private': Permission denied
/etc/hosts
sysadmin@localhost:~$
```

تنفيذ الأمرين cat find.out ، find /etc -name hosts >find.out 2>&1

&1>2 هي جزء من الأوامر وتعني إرسال stderr رسائل الخطأ (القناة رقم 2) إلى ملف ثم إرسال الإخراج القياسي stdout (القناة رقم 1) إلى نفس مكان السابق.

٨, ٢, ٨ الخطوة ٨

الإدخال القياسي Standard input (stdin) يمكن أيضاً ان يتم إعادة توجيهه. عادة يتم ادخال stdin من لوحة المفاتيح keyboard، ولكن في بعض الأحيان نريده أن يأتي من ملف بدلا من لوحة المفاتيح. على سبيل

المثال، الأمر `tr` يحول الحروف من صغيرة إلى كبيرة وبالعكس، ويقبل البيانات من `stdin`، ولكنه لا يقبل أبداً جعل ملف ما كمعطيات `argument`.

ملاحظة: اضغط على `D + Control`، للإشارة إلى الأمر `tr` وذلك لإيقاف معالجة `processing` الإدخال القياسي `standard input`:

```
tr a-z A-Z
```

```
this is interesting
```

```
how do I stop this?
```

```
^D
```

وينبغي أن يكون التنفيذ مشابه لما يلي:

```
sysadmin@localhost:~$ tr a-z A-Z
this is interesting
THIS IS INTERESTING
how do I stop this?
HOW DO I STOP THIS?
sysadmin@localhost:~$
```

تنفيذ الأمرين `tr a-z A-Z`، `^D`

ملاحظة: `^D` اختصار إلى `Control+d`.

٩، ٢، ٨ الخطوة

الأمر `tr` يقبل مدخلات لوحة المفاتيح `stdin`، ثم يترجم `translates` الأحرف ويرسل التنفيذ إلى `stdout`. لإنشاء ملف ذو أحرف صغيرة؛ نفذ ما يلي:

```
tr A-Z a-z > myfile
```

```
Wow I SEE NOW
```

```
This WORKS!
```

وينبغي أن يكون التنفيذ مشابه لما يلي:

```
sysadmin@localhost:~$ tr A-Z a-z > myfile
Wow, I SEE NOW
This WORKS!
sysadmin@localhost:~$
```

تنفيذ الأمر tr A-Z a-z > myfile

اضغط على مفتاح Enter للتأكد من ان المؤشر cursor هو على الخط أدناه "This works!", ثم استخدم Control+d لوقف الإدخال. للتحقق من إنشاء الملف، نفذ الأمر التالي:

```
cat myfile
```

```
< CTRL > + d
```

وينبغي أن يكون التنفيذ مشابه لما يلي:

```
sysadmin@localhost:~$ cat myfile
wow, i see now
this works!
sysadmin@localhost:~$
```

تنفيذ الأمر cat myfile

## ١٠, ٢, ٨ الخطوة ١٠

نفذ الأوامر التالية لإستخدام الأمر tr عن طريق إعادة توجيهه stdin من ملف:

```
cat myfile
```

```
tr a-z A-Z < myfile
```

وينبغي أن يكون التنفيذ مشابه لما يلي:

```
sysadmin@localhost:~$ cat myfile
wow, i see now
this works!
sysadmin@localhost:~$ tr a-z A-Z < myfile
WOW, I SEE NOW
THIS WORKS!
sysadmin@localhost:~$
```

تنفيذ الأمرين 'cat myfile'، 'tr a-z A-Z < myfile'

## ١١, ٢, ٨ الخطوة ١١

شكل مشهور آخر لعمليات إعادة التوجيه redirection وهو اخذ تنفيذ output امر ثم إرساله إلى أمر آخر كمدخل input. على سبيل المثال، مخرجات بعض الأوامر يمكن أن تكون جداً ضخمة massive، مما يؤدي إلى إخراج الناتج خارج الشاشة بسرعة كبيرة وبالتالي لا يستطيع المستخدم قراءته. نفذ الأمر التالي لأخذ تنفيذ الأمر ls وإرساله عبر الانبوب " | " إلى الأمر more والذي يعرض صفحة واحدة من البيانات في كل مرة:

```
ls -l /etc | more
```

وينبغي أن يكون التنفيذ مشابه لما يلي:

```

sysadmin@localhost:~$ ls -l /etc | more
total 372
-rw-r--r-- 1 root root 2981 Jan 28 2015 adduser.conf
-rw-r--r-- 1 root root 10 Jan 28 2015 adjtime
drwxr-xr-x 1 root root 900 Jan 29 2015 alternatives
drwxr-xr-x 1 root root 114 Jan 29 2015 apparmor.d
drwxr-xr-x 1 root root 168 Oct 1 2014 apt
-rw-r--r-- 1 root root 2076 Apr 3 2012 bash.bashrc
drwxr-xr-x 1 root root 72 Jan 28 2015 bash_completion.d
drwxr-sr-x 1 root bind 342 Jan 29 2015 bind
-rw-r--r-- 1 root root 356 Apr 19 2012 bindresvport.blacklist
-rw-r--r-- 1 root root 321 Mar 30 2012 blkid.conf
lrwxrwxrwx 1 root root 15 Jun 18 2014 blkid.tab -> /dev/.blkid.tab
drwxr-xr-x 1 root root 16 Jan 29 2015 ca-certificates
-rw-r--r-- 1 root root 7464 Jan 29 2015 ca-certificates.conf
drwxr-xr-x 1 root root 14 Jan 29 2015 calendar
drwxr-xr-x 1 root root 24 Jan 29 2015 cron.d
drwxr-xr-x 1 root root 134 Jan 29 2015 cron.daily
drwxr-xr-x 1 root root 24 Jan 29 2015 cron.hourly
drwxr-xr-x 1 root root 24 Jan 29 2015 cron.monthly
-rw-r--r-- 1 root root 2969 Mar 15 2012 debconf.conf
--More--

```

تنفيذ الأمر `ls -l /etc | more`

سوف تحتاج إلى الضغط على مفتاح مسافة `Spacebar` للمتابعة أو يمكنك أيضا الضغط على `CTRL-c` للخروج من هذه القائمة.

الأمر `cut` مفيد لاستخراج الحقول `fields` من الملفات التي إما أن تكون محددة `delimited` بواسطة حرف، مثل النقطتين (`:`) في `/etc/passwd`، أو أن يكون لها عرض ثابت `fixed width`. وهو يوفر قدرا كبيرا من

المخرجات، وهو ما يظهر في الأمثلة المقبلة، لذلك سنقوم بعملية إعادة التوجيه لتلك المخرجات باستخدام  
| character .

١٢، ٢، ٨ الخطوة ١٢

في المثال التالي، سوف تستخدم أمر cut لاستخراج أسماء المستخدمين الموجودين في قاعدة بيانات  
database تسمى /etc/passwd (ملف يحتوي على معلومات حساب المستخدم user account  
information)، نفذ الأمر cut فقط:

```
cut -d: -f1 /etc/passwd
```

ويأتي في التنفيذ ادناه جزء فقط من إخراج الأمر cut:

```

sysadmin@localhost:~$ cut -d: -f1 /etc/passwd
root
daemon
bin
sys
sync
games
man
lp
mail
news
uucp
proxy
www-data
backup
list
irc
gnats
nobody
libuuid
syslog
bind
sshd
operator
    
```

تنفيذ الأمر `cut -d: -f1 /etc/passwd`

١٣, ٢, ٨ الخطوة ١٣

مخرجات output الأمر السابق لا يتم عرضها بالكامل في الشاشة. سنقوم في الخطوة التالية بأخذ مخرجات الأمر `cut` وإعادة توجيهها إلى الأمر `sort`، كما في المثال التالي:

```
cut -d: -f1 /etc/passwd | sort
```

ويأتي في الشاشة ادناه جزء فقط من التنفيذ :

```
sysadmin@localhost:~$ cut -d: -f1 /etc/passwd | sort
backup
bin
bind
daemon
games
gnats
irc
libuuid
list
lp
mail
man
news
nobody
operator
proxy
root
sshd
sync
sys
```

تنفيذ جزئي للأمر `cut -d: -f1 /etc/passwd | sort`

١٤, ٢, ٨ الخطوة ١٤

الآن تمت عملية فرز التنفيذ `output`، لكن لا تزال النتائج خارج الشاشة. أرسل مخرجات الأمر `sort` إلى الأمر `more` لحل هذه المشكلة:

```
cut -d: -f1 /etc/passwd | sort | more
```

وينبغي أن يكون التنفيذ مشابه لما يلي:

```
sysadmin@localhost:~$ cut -d: -f1 /etc/passwd | sort | more
backup
bin
bind
daemon
games
gnats
irc
libuuid
list
lp
mail
man
news
nobody
operator
proxy
root
sshd
sync
sys
sysadmin
syslog
uucp
--More--
```

تنفيذ الأمر `cut -d: -f1 /etc/passwd | sort | more`

## ٨.٣ استخدام الأمر find للبحث عن الملفات

في هذه المهمة، سوف نستخدم الأمر find للقيام بعملية البحث عن الملفات.

الأمر find هو أمر مرن جداً؛ ويصاحبه مجموعة من الخيارات options؛ مما يسمح للمستخدمين بالبحث عن الملفات على أساس مجموعة من المعايير criteria مثل اسم الملف filename وحجمه size وتاريخه date ونوع الإذن permission type. ابسط صيغة للأمر هي:

```
find <directory to start search> -criteria <argument for criteria>
```

سوف يبدأ الأمر find البحث search في الدليل directory المحدد وبشكل متكرر recursively حتى في الدلائل الفرعية subdirectories.

٨,٣,١ الخطوة ١

ابحث عن الملفات التي تحتوي على اسم باش bash في الدليل الرئيسي home directory :

```
find ~ -name "*bash*"
```

وينبغي أن يكون التنفيذ مشابه لما يلي:

```
sysadmin@localhost:~$ find ~ -name "*bash*"
/home/sysadmin/.bash_logout
/home/sysadmin/.bashrc
sysadmin@localhost:~$
```

تنفيذ الأمر find ~ -name "\*bash\*"

تذكر أن تستخدم رمز تيلدا ~ لتمثيل الدليل الرئيسي للمستخدم الحالي للنظام your home directory.

## ٢,٣,٨ الخطوة ٢

ابحث عن الملفات التي تم تعديلها أو انشائها قبل أقل من ٥ دقائق في الدليل المحدد specified directory باستخدام الأوامر التالية:

```
find ~/Music -mmin -5
```

```
touch ~/Music/mysong
```

```
find ~/Music -mmin -5
```

وينبغي أن يكون التنفيذ مشابه لما يلي:

```
sysadmin@localhost:~$ find ~/Music -mmin -5
sysadmin@localhost:~$ touch ~/Music/mysong
sysadmin@localhost:~$ find ~/Music -mmin -5
/home/sysadmin/Music
/home/sysadmin/Music/mysong
sysadmin@localhost:~$
```

تنفيذ الأوامر الثلاثة find ~/Music -mmin -5، touch ~/Music/mysong، find ~/Music -mmin -5

الأمر find الأول لا يجد أي نتيجة؛ حيث انه لا توجد ملفات تم تعديلها في خلال الوقت المحدد في المثال وهو ٥ دقائق. اما السطر الثاني فهو يحتوي على الأمر touch؛ الذي يقوم كما ذكرنا سابقا بتحديث الملف أو إنشاءه في حالة عدم وجود ملف بنفس الاسم، أخيراً أعد تشغيل الأمر find ~/Music -mmin -5 مرة أخرى، وهو ما أدى إلى اكتشاف ملف جديد.

تم عرض دليل الموسيقى Music directory مع الأمر find الثاني؛ وذلك لأن الدليل تم تعديله بإضافة ملف جديد إليه.

### ٣,٣,٨ الخطوة ٣

نفذ الأمر التالي للبحث عن الملفات التي حجمها أكبر من ٢MB في الدليل /usr directory:

```
find /usr -size +2M
```

وينبغي أن يكون التنفيذ مشابه لما يلي:

```
sysadmin@localhost:~$ find /usr -size +2M
/usr/bin/python2.7
/usr/lib/perl/5.14.2/auto/Encode/JP/JP.so
/usr/lib/perl/5.14.2/auto/Encode/KR/KR.so
/usr/share/GeoIP/GeoIPv6.dat
/usr/share/file/magic.mgc
sysadmin@localhost:~$
```

تنفيذ الأمر `find /usr -size +2M`

### ٤,٣,٨ الخطوة ٤

ابحث عن الملفات التي من نوع دليل directory في الموقع المحدد:

```
find /usr/share/bug -type d
```

وينبغي أن يكون التنفيذ مشابه لما يلي:

```

sysadmin@localhost:~$ find /usr/share/bug -type d
/usr/share/bug
/usr/share/bug/apt
/usr/share/bug/gnupg
/usr/share/bug/initramfs-tools
/usr/share/bug/procps
/usr/share/bug/cron
/usr/share/bug/file
/usr/share/bug/libmagic1
/usr/share/bug/logrotate
/usr/share/bug/man-db
/usr/share/bug/vim-tiny
sysadmin@localhost:~$

```

تنفيذ الأمر `find /usr/share/bug -type d`

٥, ٣, ٨ الخطوة

للتحقق من أن التنفيذ `output` يعرض الدلائل `directories`، استخدم الخيار `-ls`. يستخدم الأمر `find` الخيار `-print` افتراضياً `by default` والذي يقوم بعرض فقط أسماء الملفات `file names`. يوفر الخيار `-ls` تفاصيل الملف:

```
find /usr/share/bug -type d -ls
```

وينبغي أن يكون التنفيذ مشابه لما يلي:

```

sysadmin@localhost:~$ find /usr/share/bug -type d -ls

 2280  0 drwxr-xr-x   1 root  root   138 Jan 29  2015 /usr/share/bug
 2281  0 drwxr-xr-x   1 root  root    12 Jan 28  2015 /usr/share/bug/
apt
 2283  0 drwxr-xr-x   1 root  root   14 Jan 28  2015 /usr/share/bug/
gnupg
 2285  0 drwxr-xr-x   1 root  root   12 Jan 28  2015 /usr/share/bug/
initramfs-tools
 2287  0 drwxr-xr-x   1 root  root   14 Jan 28  2015 /usr/share/bug/
procps
10784  0 drwxr-xr-x   1 root  root  26 Jan 29  2015 /usr/share/bug/
cron
10787  0 drwxr-xr-x   1 root  root   28 Jan 29  2015 /usr/share/bug/
file
10790  0 drwxr-xr-x   1 root  root   28 Jan 29  2015 /usr/share/bug/
libmagic1
10793  0 drwxr-xr-x   1 root  root   12 Jan 29  2015 /usr/share/bug/
logrotate
10795  0 drwxr-xr-x   1 root  root   14 Jan 29  2015 /usr/share/bug/
man-db
10797  0 drwxr-xr-x   1 root  root   26 Jan 29  2015 /usr/share/bug/
vim-tiny

sysadmin@localhost:~$

```

تنفيذ الأمر `find /usr/share/bug -type d -ls`

تذكر أن حرف `d` قبل الأذونات "`rw-r-xr-x`" يشير إلى أن الملف هو دليل `directory`، أما إذا كان ملف فسيتم وضع علامة – للدلالة على ذلك.

## ٨,٤ رؤية الملفات النصية الكبيرة

على الرغم من أن الملفات النصية الكبيرة `large text files` يمكن رؤيتها مع الأمر `cat`، إلا أنه من غير المريح التمرير إلى الأعلى في الملف. بالإضافة إلى ذلك، لا يمكن عرض الملفات الكبيرة بهذه الطريقة لأن نافذة `terminal` لديها سعة تخزين محدودة في الذاكرة وبالتالي لا يظهر إلا عدد محدد من أسطر التنفيذ `lines of .output`.

استخدام الأوامر `more` أو `less` يسمح للمستخدم بعرض البيانات على شكل صفحة أو سطر في وقت واحد. أوامر `pager` أيضا تسمح بأشكال أخرى `other forms` من التنقل `navigation` والبحث `search` الذي سنراه في هذا القسم.

ملاحظة: الأمثلة معطاه باستخدام كلا الأمرين أكثر `more` وأقل `less`. في معظم الأوقات يقوم الأمران بنفس العمل، لكن الأمر أقل `less` هو أكثر تقدما `advanced` ولديه المزيد من الميزات `features`. تعتبر معرفته الأمر أكثر `more` مهمة بمكان وذلك لأن كل توزيعات `distributions` لينكس لديها الأمر `more`؛ وفي المقابل الأمر أقل `less` يوجد في بعض التوزيعات وليس في الجميع.

إذا كنت لا ترغب في عرض الملف بأكمله أو إخراجه كأمر، هناك العديد من الأوامر التي تكون قادرة على تصفية `filter` محتويات `contents` ملف أو إخراجه. في هذا القسم سوف نتعلم استخدام امرا الرأس `head` والذيل `tail` لتكون قادر على استخراج `extract` المعلومات `information` من أعلى أو أسفل التنفيذ `output` أو من محتويات ملف.

### ٨,٤,١ الخطوة ١

من المرجح أن يكون `/etc/passwd` كبيراً جداً ليتم عرضه على من دون التمرير `scrolling` على الشاشة. لمشاهدة عرض توضيحي `demonstration`، استخدم الأمر `cat` لعرض محتويات الملف `/etc/passwd`:

```
cat /etc/passwd
```

وينبغي أن يكون التنفيذ مشابه لما يلي:

```
daemon:x:1:1:daemon:/usr/sbin:/bin/sh
bin:x:2:2:bin:/bin:/bin/sh
sys:x:3:3:sys:/dev:/bin/sh
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/bin/sh
man:x:6:12:man:/var/cache/man:/bin/sh
lp:x:7:7:lp:/var/spool/lpd:/bin/sh
mail:x:8:8:mail:/var/mail:/bin/sh
news:x:9:9:news:/var/spool/news:/bin/sh
uucp:x:10:10:uucp:/var/spool/uucp:/bin/sh
proxy:x:13:13:proxy:/bin:/bin/sh
www-data:x:33:33:www-data:/var/www:/bin/sh
backup:x:34:34:backup:/var/backups:/bin/sh
list:x:38:38:Mailing List Manager:/var/list:/bin/sh
irc:x:39:39:ircd:/var/run/ircd:/bin/sh
gnats:x:41:41:Gnats Bug-Reporting System (admin)
:/var/lib/gnats:/bin/sh
nobody:x:65534:65534:nobody:/nonexistent:/bin/sh
libuuid:x:100:101::/var/lib/libuuid:/bin/sh
syslog:x:101:103::/home/syslog:/bin/false
bind:x:102:105::/var/cache/bind:/bin/false
sshd:x:103:65534::/var/run/sshd:/usr/sbin/nologin
operator:x:1000:37::/root:/bin/sh
sysadmin:x:1001:1001:System Administrator...:/home/sysadmin:
/bin/bash
sysadmin@localhost:~$
```

تنفيذ الأمر `cat /etc/passwd`

٢, ٤, ٨ الخطوة

استخدم الأمر `more` لعرض محتويات الملف `/etc/passwd`:

more /etc/passwd

وينبغي أن يكون التنفيذ مشابه لما يلي:

```
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/bin/sh
bin:x:2:2:bin:/bin:/bin/sh
sys:x:3:3:sys:/dev:/bin/sh
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/bin/sh
man:x:6:12:man:/var/cache/man:/bin/sh
lp:x:7:7:lp:/var/spool/lpd:/bin/sh
mail:x:8:8:mail:/var/mail:/bin/sh
news:x:9:9:news:/var/spool/news:/bin/sh
uucp:x:10:10:uucp:/var/spool/uucp:/bin/sh
proxy:x:13:13:proxy:/bin:/bin/sh
www-data:x:33:33:www-data:/var/www:/bin/sh
backup:x:34:34:backup:/var/backups:/bin/sh
list:x:38:38:Mailing List Manager:/var/list:/bin/sh
irc:x:39:39:ircd:/var/run/ircd:/bin/sh
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/bin/sh
nobody:x:65534:65534:nobody:/nonexistent:/bin/sh
libuuid:x:100:101::/var/lib/libuuid:/bin/sh
syslog:x:101:103::/home/syslog:/bin/false
bind:x:102:105::/var/cache/bind:/bin/false
sshd:x:103:65534::/var/run/sshd:/usr/sbin/nologin
operator:x:1000:37::/root:/bin/sh
--More-- (92%)
```

تنفيذ الأمر more /etc/passwd

ملاحظة: "--More--(92%)" يدل على أنك في الأمر أكثر more، اما النسبة ٩٢٪ فتدل على كمية البيانات الحالية current data.

٣, ٤, ٨ الخطوة ٣

طالما أنك مازلت في الأمر أكثر more؛ فيمكنك عرض شاشة التعليمات help screen عن طريق الضغط pressing على مفتاح h:

h

وينبغي أن يكون التنفيذ مشابه لما يلي:

Most commands optionally preceded by integer argument k. Defaults in brackets

Star (\*) indicates argument becomes new default.

```
-----
<space>          Display next k lines of text [current screen
                  size]
z                Display next k lines of text [current screen
                  size]*
<return>        Display next k lines of text [1]*
d or ctrl-D     Scroll k lines [current scroll size, initially 11
                  ]*
q or Q or <interrupt> Exit from more
s                Skip forward k lines of text [1]
f                Skip forward k screenfuls of text [1]
b or ctrl-B     Skip backwards k screenfuls of text [1]
'                Go to place where previous search started
=                Display current line number
/<regular expression> Search for kth occurrence of regular
                  expression [1]
n                Search for kth occurrence of last r.e [1]
!<cmd> or :!<cmd> Execute <cmd> in a subshell
v                Start up /usr/bin/vi at current line
ctrl-L          Redraw screen
:n              Go to kth next file [1]
:p              Go to kth previous file [1]
:f              Display current file name and line number
.               Repeat previous command
-----
```

```
--More-- (92%)
```

تنفيذ الضغط على مفتاح h للأمر more

## ٤, ٤, ٨ الخطوة ٤

اضغط على مفتاح المسافة Spacebar لعرض بقية الوثيقة:

< SPACE >

في المثال التالي سوف تتعلم كيفية البحث في وثيقة باستخدام الأمران أكثر more أو أقل less.

البحث عن نمط pattern داخل الأمرين أكثر more وأقل less يتم بكتابة الخط المائل / يليه النمط المراد العثور عليه. إذا تم العثور على تطابق فينبغي أن تنتقل الشاشة إلى أول تطابق. للمضي قدما إلى التطابق التالي اضغط على مفتاح n. اما مع الأمر أقل الأمر less يمكنك أن تتحرك للخلف إلى التطابق السابق عن طريق الضغط على مفتاح N (capital n).

## ٥, ٤, ٨ الخطوة ٥

استخدم الأمر أقل less لعرض كامل محتويات الملف /etc/passwd. ثم ابحث عن كلمة bin، استخدم n للتحرك للأمام، وحرف N للرجوع إلى الوراء. وأخيرا انهي الأمر less pager عن طريق كتابة الحرف q:

```
less /etc/passwd
```

```
/bin
```

```
nnnNNNq
```

وينبغي أن يكون التنفيذ مشابه لما يلي:

```

root:x:0:0:root:/root:/bin/bash

daemon:x:1:1:daemon:/usr/sbin:/bin/sh

bin:x:2:2:bin:/bin:/bin/sh

sys:x:3:3:sys:/dev:/bin/sh

sync:x:4:65534:sync:/bin:/bin/sync

games:x:5:60:games:/usr/games:/bin/sh

man:x:6:12:man:/var/cache/man:/bin/sh

lp:x:7:7:lp:/var/spool/lpd:/bin/sh

mail:x:8:8:mail:/var/mail:/bin/sh

news:x:9:9:news:/var/spool/news:/bin/sh

uucp:x:10:10:uucp:/var/spool/uucp:/bin/sh

proxy:x:13:13:proxy:/bin:/bin/sh

www-data:x:33:33:www-data:/var/www:/bin/sh

backup:x:34:34:backup:/var/backups:/bin/sh

list:x:38:38:Mailing List Manager:/var/list:/bin/sh

irc:x:39:39:ircd:/var/run/ircd:/bin/sh

gnats:x:41:41:Gnats BugReporting System(admin):/var/lib/gnats:/bin/sh

nobody:x:65534:65534:nobody:/nonexistent:/bin/sh

libuuid:x:100:101::/var/lib/libuuid:/bin/sh

syslog:x:101:103::/home/syslog:/bin/false

bind:x:102:105::/var/cache/bind:/bin/false

sshd:x:103:65534::/var/run/sshd:/usr/sbin/nologin

operator:x:1000:37::/root:/bin/sh

/etc/passwd

```

تنفيذ الامر `less /etc/passwd`

ثم بعد ذلك نقوم بعملية البحث عن كلمة `bin` كما في التنفيذ التالي:

/bin

## البحث عن bin

وتأتي نتيجة البحث كما يلي:

```
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/bin/sh
bin:x:2:2:bin:/bin:/bin/sh
sys:x:3:3:sys:/dev:/bin/sh
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/bin/sh
man:x:6:12:man:/var/cache/man:/bin/sh
lp:x:7:7:lp:/var/spool/lpd:/bin/sh
mail:x:8:8:mail:/var/mail:/bin/sh
news:x:9:9:news:/var/spool/news:/bin/sh
uucp:x:10:10:uucp:/var/spool/uucp:/bin/sh
proxy:x:13:13:proxy:/bin:/bin/sh
www-data:x:33:33:www-data:/var/www:/bin/sh
backup:x:34:34:backup:/var/backups:/bin/sh
list:x:38:38:Mailing List Manager:/var/list:/bin/sh
irc:x:39:39:ircd:/var/run/ircd:/bin/sh
gnats:x:41:41:Gnats BugReporting System(admin):/var/lib/gnats:/bin/sh
nobody:x:65534:65534:nobody:/nonexistent:/bin/sh
libuid:x:100:101:./var/lib/libuid:/bin/sh
syslog:x:101:103:./home/syslog:/bin/false
bind:x:102:105:./var/cache/bind:/bin/false
sshd:x:103:65534:./var/run/sshd:/usr/sbin/nologin
operator:x:1000:37:./root:/bin/sh
:
```

نتيجة البحث عن كلمة bin

ونستخدم حرف n للانتقال الى النتيجة التالية للبحث كما يظهر في الشكل التالي:

```
daemon:x:1:1:daemon:/usr/sbin:/bin/sh
bin:x:2:2:bin:/bin:/bin/sh
sys:x:3:3:sys:/dev:/bin/sh
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/bin/sh
man:x:6:12:man:/var/cache/man:/bin/sh
lp:x:7:7:lp:/var/spool/lpd:/bin/sh
mail:x:8:8:mail:/var/mail:/bin/sh
news:x:9:9:news:/var/spool/news:/bin/sh
uucp:x:10:10:uucp:/var/spool/uucp:/bin/sh
proxy:x:13:13:proxy:/bin:/bin/sh
www-data:x:33:33:www-data:/var/www:/bin/sh
backup:x:34:34:backup:/var/backups:/bin/sh
list:x:38:38:Mailing List Manager:/var/list:/bin/sh
irc:x:39:39:ircd:/var/run/ircd:/bin/sh
gnats:x:41:41:Gnats BugReporting System (admin) :/var/lib/gnats:/bin/sh
nobody:x:65534:65534:nobody:/nonexistent:/bin/sh
libuuid:x:100:101::/var/lib/libuuid:/bin/sh
syslog:x:101:103::/home/syslog:/bin/false
bind:x:102:105::/var/cache/bind:/bin/false
sshd:x:103:65534::/var/run/sshd:/usr/sbin/nologin
operator:x:1000:37::/root:/bin/sh
sysadmin:x:1001:1001:System Administrator,,,:/home/sysadmin:/bin/bash
(END)
```

تنفيذ البحث عن النتيجة التالية عن طريق حرف n

واخيراً، للعودة الى نتيجة البحث السابقة نقوم بالضغط على الحرف N كما يظهر ادناه:

```
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/bin/sh
bin:x:2:2:bin:/bin:/bin/sh
sys:x:3:3:sys:/dev:/bin/sh
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/bin/sh
man:x:6:12:man:/var/cache/man:/bin/sh
lp:x:7:7:lp:/var/spool/lpd:/bin/sh
mail:x:8:8:mail:/var/mail:/bin/sh
news:x:9:9:news:/var/spool/news:/bin/sh
uucp:x:10:10:uucp:/var/spool/uucp:/bin/sh
proxy:x:13:13:proxy:/bin:/bin/sh
www-data:x:33:33:www-data:/var/www:/bin/sh
backup:x:34:34:backup:/var/backups:/bin/sh
list:x:38:38:Mailing List Manager:/var/list:/bin/sh
irc:x:39:39:ircd:/var/run/ircd:/bin/sh
gnats:x:41:41:Gnats BugReporting System (admin):/var/lib/gnats:/bin/sh
nobody:x:65534:65534:nobody:/nonexistent:/bin/sh
libuuid:x:100:101::/var/lib/libuuid:/bin/sh
syslog:x:101:103::/home/syslog:/bin/false
bind:x:102:105::/var/cache/bind:/bin/false
sshd:x:103:65534::/var/run/sshd:/usr/sbin/nologin
operator:x:1000:37::/root:/bin/sh
:
```

نتيجة الضغط على حرف N

يختلف الأمر `more` عن الأمر `less`، حيث ان الأول يخرج تلقائيا `automatically` عند الوصول إلى نهاية الملف، اما الآخر فيجب الضغط على مفتاح `q` لإنهاء عمله.

٦، ٤، ٨ الخطوة ٦

يمكنك استخدام الأمر `head` رأس `head` لعرض الجزء العلوي من ملف. افتراضيا؛ سيقوم الأمر `head` بعرض أول عشرة أسطر من الملف:

```
head /etc/passwd
```

وينبغي أن يكون التنفيذ مشابه لما يلي:

```
sysadmin@localhost:~$ head /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/bin/sh
bin:x:2:2:bin:/bin:/bin/sh
sys:x:3:3:sys:/dev:/bin/sh
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/bin/sh
man:x:6:12:man:/var/cache/man:/bin/sh
lp:x:7:7:lp:/var/spool/lpd:/bin/sh
mail:x:8:8:mail:/var/mail:/bin/sh
news:x:9:9:news:/var/spool/news:/bin/sh
sysadmin@localhost:~$
```

تنفيذ الأمر `head /etc/passwd`

٧، ٤، ٨ الخطوة ٧

إستخدم الأمر `tail` لعرض آخر عشر اسطر من الملف `/etc/passwd`:

```
tail /etc/passwd
```

وينبغي أن يكون التنفيذ مشابه لما يلي:

```
sysadmin@localhost:~$ tail /etc/passwd
list:x:38:38:Mailing List Manager:/var/list:/bin/sh
irc:x:39:39:ircd:/var/run/ircd:/bin/sh
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats
:/bin/sh
nobody:x:65534:65534:nobody:/nonexistent:/bin/sh
libuuid:x:100:101::/var/lib/libuuid:/bin/sh
syslog:x:101:103::/home/syslog:/bin/false
bind:x:102:105::/var/cache/bind:/bin/false
sshd:x:103:65534::/var/run/sshd:/usr/sbin/nologin
operator:x:1000:37::/root:/bin/sh
sysadmin:x:1001:1001:System Administrator....:/home/sysadmin:
/bin/bash
sysadmin@localhost:~$
```

تنفيذ الأمر tail /etc/passwd

٨، ٤، ٨ الخطوة

استخدم أمر رأس head لعرض أول سطرين من الملف /etc/passwd:

```
head -2 /etc/passwd
```

وينبغي أن يكون التنفيذ مشابه لما يلي:

```
sysadmin@localhost:~$ head -2 /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/bin/sh
sysadmin@localhost:~$
```

تنفيذ الأمر `head -2 /etc/passwd`

٩, ٤, ٨ الخطوة ٩

نفذ السطر التالي للإسناد مخرجات الأمر `ls` الى الأمر الذيل `tail`، وبالتالي عرض اخر خمسة أسماء من ملفات الدليل `/etc`:

```
ls /etc | tail -5
```

وينبغي أن يكون التنفيذ مشابه لما يلي:

```
sysadmin@localhost:~$ ls /etc | tail -5
update-motd.d
updatedb.conf
vim
wgetrc
xml
sysadmin@localhost:~$
```

تنفيذ الأمر `ls /etc | tail -5`

كما رأينا في الأمرين الرأس `head` والذيل `tail`؛ فان تنفيذ `output` عشرة اسطر يكون افتراضيا. يمكنك أيضا استخدام خيار `#` - (أو يمكنك استخدام الخيار `# -n` ، حيث ان علامة الشبكة `#` ترمز لعدد اسطر المخرجات). أيضاً كلا الأمران يمكن أن يستخدموا لقراءة الإدخال القياسي `read standard input` من القناة التي تستقبل التنفيذ `receives output` من الأوامر `command`.

لقد رأينا أيضا الإختلاف بين الأمرين الرأس **head** والذيل **tail**، حيث ان الأول يبدأ عد اسطر التنفيذ من الجزء العلوي من البيانات **top of the data**، في حين ان الأمر الذيل يعد اسطر التنفيذ من الجزء السفلي للبيانات **bottom of the data**. أيضا هناك بعض الإختلافات الإضافية بين هذين الأمرين كما سيتم توضيحه في المهام القادمة.

١٠، ٤، ٨، الخطوة ١٠

بجانب الطريقة السابقة لتحديد عدد الاسطر، فإننا نستطيع ومع الأمر رأس **head** استخدام الخيار **-n -#** ؛ لاحظ ان رمز (-) أمام رمز **#**. حيث ان رمز الشبكة **#** يرمز لعدد الاسطر التي سيتم استبعادها **exclude** وتحديدًا من الجزء السفلي من التنفيذ. بعبارة أخرى فان الامر **head -n -20 /etc/passwd** يقوم بإنقاص عدد ٢٠ سطر من العدد الكلي للأسطر بداية من الأسفل. على سبيل المثال، إذا كان الملف **/etc/passwd** يحتوي على ٢٤ سطر فان الأمر التالي سوف يعرض الاسطر الأربعة الأولى من السطر ١ الى السطر ٤ وبالتالي استثناء الأسطر العشرين الأخيرة:

```
head -n -20 /etc/passwd
```

وينبغي أن يكون التنفيذ مشابه لما يلي:

```
sysadmin@localhost:~$ head -n -20 /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/bin/sh
bin:x:2:2:bin:/bin:/bin/sh
sys:x:3:3:sys:/dev:/bin/sh
sysadmin@localhost:~$
```

تنفيذ الأمر **head -n -20 /etc/passwd**

## ٨,٥ البحث عن النص عن طريق التعابير العادية

في هذه المهمة، سوف نستخدم مجموعة الأوامر `grep` مع التعابير العادية `regular expressions` من أجل البحث عن سلسلة `string` محددة من الأحرف في قطعة `stream` من البيانات ؛ على سبيل المثال ملف نصي `.text file`.

يستخدم الأمر `grep` مع التعابير العادية الأساسية `basic regular expressions` والاحرف الخاصة `special characters` مثل أحرف البديل `wildcards` التي تطابق `match` أنماط البيانات `data patterns`. يقوم الأمر `grep` بإرجاع كامل السطر الذي يحتوي على النمط المطابق.

الخيار `-E` يمكن اضافته للأمر `grep` لإجراء عمليات بحث مع التعابير العادية الطويلة `extended regular expressions`، وهناك أيضا طريقة أخرى لإستخدام التعابير العادية الطويلة وهي إستخدام الأمر `egrep`.

يستخدم الأمر `fgrep` لمطابقة الأحرف الحرفية `literal characters`، وتجاهل المعنى الخاص من احرف التعبير العادية `regular expression characters`.

### ٨,٥,١ الخطوة ١

إن ابسط شكل للأمر `grep` هو البحث عن سلسلة معطاه من الاحرف `string of characters`، مثل البحث عن `SSD` في الملف `/etc/passwd`. الأمر `grep` سيقوم بطباعة كامل السطر `entire line` والذي يوجد به التطابق `match`:

```
cd /etc
```

```
grep sshd passwd
```

وينبغي أن يكون التنفيذ مشابه لما يلي:

```
sysadmin@localhost:~$ cd /etc
sysadmin@localhost:/etc$ grep sshd passwd
sshd:x:103:65534:./var/run/ssh:/usr/sbin/nologin
sysadmin@localhost:/etc$
```

تنفيذ الأمرين `cd /etc` ، `grep sshd passwd`

٢,٥,٨ الخطوة ٢

التعبير العادية Regular expressions سوف تطابق كل حرف every single instance من نمط محدد specified pattern:

```
grep root passwd
```

وينبغي أن يكون التنفيذ مشابه لما يلي:

```
sysadmin@localhost:/etc$ grep root passwd
root:x:0:0:root:/root:/bin/bash
operator:x:1000:37:./root:/bin/sh
sysadmin@localhost:/etc$
```

تنفيذ الأمر `grep root passwd`

لاحظ ان الضوء الأحمر يشير إلى ما يتطابق بالضبط. يمكنك أن ترى أيضا أن كافة ما يظهر في الجذر occurrences of root متطابق matched في كل سطر.

٣,٥,٨ الخطوة ٣

للحد من المخرجات limit the output، يمكنك استخدام تعبيرات عادية regular expressions لتحديد نمط أكثر دقة precise pattern. على سبيل المثال ادخال الحرف (^) يستخدم لعمل تطابق مع نمط pattern

في بداية السطر. لذلك عند تنفيذ سطر الأوامر التالي، الاسطر التي تبدأ بالجذر فقط root يجب أن تكون مطابقة ومعروضه باللون الاحمر:

```
grep '^root' passwd
```

وينبغي أن يكون التنفيذ مشابه لما يلي:

```
sysadmin@localhost:/etc$ grep '^root' passwd
root:x:0:0:root:/root:/bin/bash
sysadmin@localhost:/etc$
```

تنفيذ الأمر `grep '^root' passwd`

لاحظ أن هناك حالتين إضافيتين من كلمة جذر root ولكن فقط واحدة تظهر باللون الأحمر وهي الموجودة في بداية السطر.

أفضل الممارسات: استخدم علامات الاقتباس المفردة لا المزدوجة مع التعبيرات العادية regular expressions وذلك لمنع برنامج القشرة shell من محاولة تفسيرها interpret.

٤, ٥, ٨ الخطوة ٤

طابق النمط sync في أي مكان على السطر:

```
grep 'sync' passwd
```

وينبغي أن يكون التنفيذ مشابه لما يلي:

```
sysadmin@localhost:/etc$ grep 'sync' passwd
sync:x:4:65534:sync:/bin:/bin/sync
sysadmin@localhost:/etc$
```

تنفيذ الأمر `grep 'sync' passwd`

٥, ٥, ٨ الخطوة ٥

استخدم الرمز `$` لتجد التطابق مع النمط `sync` في نهاية السطر:

```
grep 'sync$' passwd
```

وينبغي أن يكون التنفيذ مشابه لما يلي:

```
sysadmin@localhost:/etc$ grep 'sync$' passwd
sync:x:4:65534:sync:/bin:/bin/sync
sysadmin@localhost:/etc$
```

تنفيذ الأمر `grep 'sync$' passwd`

الأمر الأول يطابق كل حرف من الأحرف. أما الثاني يطابق فقط المثل في نهاية السطر.

٦, ٥, ٨ الخطوة ٦

استخدام حرف النقطة `.` period (.) لمطابقة أي حرف. على سبيل المثال، نفذ الأمر التالي لمطابقة أي حرف متبوع بحرف `y`:

```
grep '.y' passwd
```

وينبغي أن يكون التنفيذ مشابه لما يلي:

```
sysadmin@localhost:/etc$ grep '.y' passwd
sys:x:3:3:sys:/dev:/bin/sh
sync:x:4:65534:sync:/bin:/bin/sync
proxy:x:13:13:proxy:/bin:/bin/sh
gnats:x:41:41:Gnats Bug-Reporting System
(admin):/var/lib/gnats:/bin/sh
nobody:x:65534:65534:nobody:/nonexistent:/bin/sh
syslog:x:101:103::/home/syslog:/bin/false
sysadmin:x:1001:1001:System Administrator,,:/home/sysadmin:
/bin/bash
sysadmin@localhost:/etc$
```

تنفيذ الأمر `grep '.y' passwd`

٧,٥,٨ الخطوة ٧

حرف المستقيم " | "، أو ما يسمى بعامل التناوب `alternation operator`، يستخدم لعمل جملة أو `or operator`. على سبيل المثال، نفذ ما يلي لمحاولة مطابقة إما الجذر `sshd` أو المشغل `operator`:

```
grep 'sshd|root|operator' passwd
```

وينبغي أن يكون التنفيذ مشابه لما يلي:

```
sysadmin@localhost:/etc$ grep 'sshd|root|operator' passwd
sysadmin@localhost:/etc$
```

تنفيذ الأمر `grep 'sshd|root|operator' passwd`

نلاحظ أن الأمر `grep` لا يميز المدخلات افتراضيا. وبالتالي فهو ينظر الى الحرف المستقيم " | " كحرف عادي `plain character` في نمط ما ويجب أن يتطابق للحصول على نتيجة. ولحل تلك المشكلة، يجب استخدام إما `grep -E` أو `egrep`، وهو ما يسمح باستخدام التعابير العادية الطويلة `extended regular expressions` بما في ذلك حروف التناوب `alternation` مثل الحرف المستقيم " | ".

## ٨,٥,٨ الخطوة ٨

استخدام رمز التبديل -E للسماح للأمر grep للعمل في الوضع الموسع extended mode من أجل التعرف على عامل التناوب alternation operator:

```
grep -E 'sshd|root|operator' passwd
```

وينبغي أن يكون التنفيذ مشابه لما يلي:

```
sysadmin@localhost:/etc$ grep -E 'sshd|root|operator' passwd
root:x:0:0:root:/root:/bin/bash
sshd:x:103:65534::/var/run/sshd:/usr/sbin/nologin
operator:x:1000:37::/root:/bin/sh
sysadmin@localhost:/etc$
```

تنفيذ الأمر grep -E 'sshd|root|operator' passwd

## ٨,٥,٩ الخطوة ٩

هنا فإننا نستخدم تعبير عادي موسع extended regular expression آخر، وهذه المرة مع الأمر egrep، لكن الاختلاف هو ان التطابق يتم مع نمط مجموعة كاملة. مثل السلاسل nob و non:

```
egrep 'no(b|n)' passwd
```

وينبغي أن يكون التنفيذ مشابه لما يلي:

```
sysadmin@localhost:/etc$ egrep 'no(b|n)' passwd
nobody:x:65534:65534:nobody:/nonexistent:/bin/sh
sysadmin@localhost:/etc$
```

تنفيذ الأمر egrep 'no(b|n)' passwd

ملاحظة: الأقواس parenthesis ( ) تستخدم للحد من المدى scope للحرف |. مثلاً nob|n النمط بدون الأقواس قد يعني تطابق إما nob أو n.

١٠، ٥، ٨ الخطوة ١٠

يمكن أيضا استخدام الأقواس المربعة [ ] لمطابقة حرف واحد، ولكن على عكس النقطة period (.)، يتم استخدام [ ] لتحديد بالضبط ما الحرف الذي ترغب في مطابقته. على سبيل المثال، إذا كنت تريد أن تطابق حرف رقمي، يمكنك التحديد من [٠-٩]. نفذ الأمر التالي:

```
head passwd | grep '[0-9]'
```

وينبغي أن يكون التنفيذ مشابه لما يلي:

```
sysadmin@localhost:/etc$ head passwd | grep '[0-9]'
```

```
root:x:0:0:root:/root:/bin/bash
```

```
daemon:x:1:1:daemon:/usr/sbin:/bin/sh
```

```
bin:x:2:2:bin:/bin:/bin/sh
```

```
sys:x:3:3:sys:/dev:/bin/sh
```

```
sync:x:4:65534:sync:/bin:/bin/sync
```

```
games:x:5:60:games:/usr/games:/bin/sh
```

```
man:x:6:12:man:/var/cache/man:/bin/sh
```

```
lp:x:7:7:lp:/var/spool/lpd:/bin/sh
```

```
mail:x:8:8:mail:/var/mail:/bin/sh
```

```
news:x:9:9:news:/var/spool/news:/bin/sh
```

```
sysadmin@localhost:/etc$
```

تنفيذ الأمر 'head passwd | grep '[0-9]'

ملاحظة: تم استخدام الأمر الرأس head للحد من إخراج output الأمر grep.

## ١١، ٥، ٨ الخطوة ١١

افتراض أنك تريد البحث عن نمط يحتوي على سلسلة من ثلاثة أرقام. يمكنك استخدام الأقواس { } مع عدد ما للتعبير بأنك تريد تكرار نمط محدد ولعدد من المرات. على سبيل المثال: {٣}، لاحظ إن استخدام تصفيات رقمية numeric qualifier يتطلب الوضع الموسع extended mode من grep:

```
grep -E '[0-9]{3}' passwd
```

وينبغي أن يكون التنفيذ مشابه لما يلي:

```
sysadmin@localhost:/etc$ grep -E '[0-9]{3}' passwd
sync:x:4:65534:sync:/bin:/bin/sync
nobody:x:65534:65534:nobody:/nonexistent:/bin/sh
libuuid:x:100:101:./var/lib/libuuid:/bin/sh
syslog:x:101:103:./home/syslog:/bin/false
bind:x:102:105:./var/cache/bind:/bin/false
sshd:x:103:65534:./var/run/sshd:/usr/sbin/nologin
operator:x:1000:37:./root:/bin/sh
sysadmin:x:1001:1001:System Administrator....:/home/sysadmin:
/bin/bash
sysadmin@localhost:/etc$
```

تنفيذ الأمر grep -E '[0-9]{3}' passwd

## المعمل التاسع – البرمجة الاساسية

## Basic Scripting

## ٩,١ مقدمة

هذا هو المعمل التاسع أوامر البرمجة Scripting Commands في لينكس. عن طريق أداء هذا التمرين؛ سوف تتعلم كيفية استخدام محرر vi لإنشاء برنامج نصي بسيط في القشرة basic shell scripts باستخدام أوامر القشرة shell commands والمتغيرات variables وجمل التحكم control statements.

في هذا التمرين، سوف نقوم بالمهام التالية:

- استخدام محرر vi لإنشاء وتحرير الملفات النصية.
- إنشاء البرامج النصية البسيطة في القشرة.
- إنشاء البرامج النصية في القشرة مع التنفيذ المشروط.
- استخدام الحلقات في برنامج النصية لغرض التكرار.

## ٩,٢ تحرير النص الأساسي

معظم توزيعات distributions لينكس لديها أكثر من محرر نصوص text editor واحد. محرر النص يمكن ان يحتوي على محرر نص بسيط فقط simple text-only editors مثل المحرر nano، أو يكون لديه محرر رسومي graphical editors مثل المحرر gedit.

في هذه المهمة، سوف تستكشف بعض مميزات تحرير النصوص الأساسية للمحرر vi editor. جميع توزيعات لينكس لديها نسخ من المحرر vi والذي يعتبر محرر نص قوي خاص بالتدريب ولكنه قادر على أداء مجموعة متنوعة من مهام تحرير النص text editing tasks.

محرر vi له قسمان رئيسيان هما وضع الأوامر command ووضع الإدراج insert. الوضع الاول تستطيع فيه القيام بالعمليات؛ مثل الملاححة navigation والبحث searching والحفظ saving والخروج من المحرر exiting the editor. أما الوضع الثاني وهو الإدراج؛ وفيه نستطيع كتابة أي نص في المستند document.

## ٩,٢,١ الخطوة ١

لإنشاء ملف جديد، نفذ الأمر التالي:

```
vi myfile
```

اكتب `i` للدخول إلى وضع إدراج `insert` في المحرر `vi`. ثم أدخل النص التالي:

```
i
```

```
Welcome to the vi editor.
```

```
It is a very powerful text editor.
```

```
Especially for those who master it.
```

ثم اضغط على مفتاح `Esc` لترك وضع الإدراج `insert mode`. اكتب `wq`: لكتابة الملف في القرص ثم الخروج.

```
< ESC >
```

```
:wq
```

ملاحظة: ستتم تغطية كل من الأوامر السابقة بمزيد من التفصيل لاحقاً في هذا المعمل؛ حيث أن الغرض من الخطوة السابقة إنشاء ملف والعمل عليه خلال هذا الدرس.

## ٩,٢,٢ الخطوة ٢

استدعي المحرر `vi editor` لتعديل الملف الذي قمت بإنشائه. عندما يتم استدعاء `vi editor`، تصبح في وضع الأوامر وليس الإدراج:



### ٣,٢,٩ الخطوة ٣

اضغط `press` على كل من المفاتيح التالية مرتين ولاحظ كيف يتحرك المؤشر `cursor`. تذكر أنك في وضع الأوامر `command mode`:

المفتاح	الوظيفة
j	يتحرك المؤشر للأسفل سطر واحد (مماثل لمفتاح سهم للأسفل)
K	يتحرك المؤشر فوق السطر (مماثل لمفتاح سهم للأعلى)
L	يتحرك المؤشر إلى اليمين حرف واحد (مماثل لمفتاح سهم لليمن)
H	يتحرك المؤشر إلى اليسار حرف واحد (مماثل لمفتاح سهم لليساار)
W	يتحرك المؤشر إلى بداية الكلمة التالية
E	يتحرك المؤشر إلى نهاية الكلمة
b	يتحرك المؤشر إلى بداية الكلمة السابقة

جدول مفاتيح التحكم لتحريك المؤشر في وضع الأوامر

تحذير: إذا قمت بالضغط على أي مفاتيح أخرى غير المذكورة أعلاه، فربما تنتقل الى وضع الإدراج `insert mode`. لا داعي للقلق! اضغط على مفتاح `Esc` للخروج من ذلك الوضع، ثم في وضع الأوامر أكتب `q!`:  
 + مفتاح `Enter`. وهذا ينبغي أن يخرجك من المحرر `vi` دون حفظ أي تغييرات. ثم نفذ `vi myfile` للعودة والعمل على المحرر `vi`.

### ٤,٢,٩ الخطوة ٤

المزيد عن المحرر `vi`: اضغط على المفاتيح التالية ولاحظ كيف يتحرك المؤشر:

المفتاح	الوظيفة
\$	يتحرك المؤشر إلى نهاية السطر الحالي (مماثل لمفتاح النهاية End)
0 (zero)	يتحرك المؤشر إلى بداية السطر الحالي (مماثل لمفتاح البداية Home)
3G	يقفز إلى السطر الثالث (nG يقفز إلى السطر n)
1G	يقفز إلى السطر الأول
G	يقفز إلى السطر الأخير

جدول فيه المزيد من مفاتيح التحكم لتحريك المؤشر في وضع الأوامر

٥,٢,٥ الخطوة ٥

يتم نقل المؤشر cursor إلى بداية الكلمة very عن طريق الضغط على المفاتيح التالية:

```
G
k
8l
```

يجب أن يكون المؤشر على حرف v في كلمة very كما هو مبين أدناه:

```
Welcome to the vi editor.
It is a very powerful text editor.

Especially for those who master it.
~
~
```

تنفيذ لمفاتيح التحكم بالمؤشر G، k، 8l

٦,٢,٩ الخطوة ٦

احذف كلمة very عن طريق الأمر dw (حذف كلمة delete word):

dw

وينبغي أن يكون التنفيذ مشابه لما يلي:

```
Welcome to the vi editor.  
It is a powerful text editor.  
  
Especially for those who master it.  
~
```

تنفيذ الأمر dw

٧,٢,٩ الخطوة ٧

التراجع عن العملية الأخيرة:

u

وينبغي أن يكون التنفيذ مشابه لما يلي:

```
Welcome to the vi editor.  
It is a very powerful text editor.  
  
Especially for those who master it.  
~  
~
```

تنفيذ الأمر u

٨,٢,٩ الخطوة ٨

حذف كلمتين:

2dw

وينبغي أن يكون التنفيذ مشابه لما يلي:

```
Welcome to the vi editor.  
It is a text editor.  
  
Especially for those who master it.  
~  
~
```

تنفيذ الأمر 2dw

٩,٢,٩ الخطوة ٩

التراجع عن العملية الأخيرة:

u

وينبغي أن يكون التنفيذ مشابه لما يلي:

```
Welcome to the vi editor.  
It is a very powerful text editor.  
  
Especially for those who master it.  
~  
~
```

تنفيذ الأمر u

١٠، ٢، ٩ الخطوة ١٠

اضغط على الحرف x أربع مرات لحذف أربعة أحرف:

XXXX

وينبغي أن يكون التنفيذ مشابه لما يلي:

```
Welcome to the vi editor.  
It is a █ powerful text editor.  
  
Especially for those who master it.  
~  
~
```

تنفيذ الأمر XXXX

١١، ٢، ٩ الخطوة ١١

التراجع عن العمليات الاربعة الأخيرة واستعادة الأحرف المحذوفة:

4u

وينبغي أن يكون التنفيذ مشابه لما يلي:

```
Welcome to the vi editor.  
It is a █ very powerful text editor.  
  
Especially for those who master it.  
~  
~
```

تنفيذ الأمر 4u

١٢, ٢, ٩ الخطوة ١٢

حذف ١٤ حرفاً:

14x

وينبغي أن يكون التنفيذ مشابه لما يلي:

```
Welcome to the vi editor.  
It is a utext editor.  
Especially for those who master it.  
~  
~
```

تنفيذ الأمر 14x

١٣, ٢, ٩ الخطوة ١٣

التراجع عن العملية الأخيرة:

u

وينبغي أن يكون التنفيذ مشابه لما يلي:

```
Welcome to the vi editor.  
It is a uvery powerful text editor.  
Especially for those who master it.  
~  
~
```

تنفيذ الأمر u

١٤, ٢, ٩ الخطوة ١٤

لحذف الأحرف الخمسة إلى يسار المؤشر اكتب `٥` ثم `X`:

5X

وينبغي أن يكون التنفيذ مشابه لما يلي:

```
Welcome to the vi editor.  
It very powerful text editor.  
  
Especially for those who master it.  
~  
~
```

تنفيذ الأمر 5X

١٥, ٢, ٩ الخطوة ١٥

التراجع عن العملية الأخيرة:

u

وينبغي أن يكون التنفيذ مشابه لما يلي:

```
Welcome to the vi editor.  
It is a very powerful text editor.  
  
Especially for those who master it.  
~  
~
```

تنفيذ الأمر u

١٦، ٢، ٩ الخطوة ١٦

حذف السطر الحالي:

dd

وينبغي أن يكون التنفيذ مشابه لما يلي:

```
Welcome to the vi editor.  
Especially for those who master it.  
~  
~
```

تنفيذ الأمر dd

١٧، ٢، ٩ الخطوة ١٧

مهما كان ما تم حذفه أو قصه فانه يمكن لصقه. الصق الاسطر المحذوفة تحت السطر الحالي:

p

وينبغي أن يكون التنفيذ مشابه لما يلي:

```
Welcome to the vi editor.  
Especially for those who master it.  
It is a very powerful text editor.  
~  
~
```

تنفيذ الأمر p

١٨, ٢, ٩ الخطوة ١٨

للتراجع عن العمليتين الماضيتين:

2u

وينبغي أن يكون التنفيذ مشابه لما يلي:

```
Welcome to the vi editor.  
It is a very powerful text editor.  
Especially for those who master it.  
~  
~
```

تنفيذ الأمر 2u

١٩, ٢, ٩ الخطوة ١٩

حذف سطرين، الحالي والذي يليه:

2dd

وينبغي أن يكون التنفيذ مشابه لما يلي:

```
Welcome to the vi editor.  
~  
~
```

تنفيذ الأمر 2dd

٢٠, ٢, ٩ الخطوة ٢٠

التراجع عن العملية الأخيرة:

u

وينبغي أن يكون التنفيذ مشابه لما يلي:

```
Welcome to the vi editor.  
It is a very powerful text editor.  
Especially for those who master it.  
~  
~
```

تنفيذ الأمر u

٢١، ٢، ٩ الخطوة ٢١

الانتقال إلى الكلمة الرابعة ثم الحذف من المكان الحالي إلى نهاية السطر باستخدام D:

4w

D

وينبغي أن يكون التنفيذ مشابه لما يلي:

```
Welcome to the vi editor.  
It is a very  
Especially for those who master it.  
~  
~
```

تنفيذ الأمرين 4w، D

يحذف الأمر d\$ أيضا إلى نهاية السطر. لان \$ كما تقدم تعني الانتقال إلى نهاية السطر. وبالتالي فإن d\$ سوف تحذف إلى نهاية السطر.

٢٢، ٢، ٩ الخطوة ٢٢

التراجع عن العملية الأخيرة:

u

وينبغي أن يكون التنفيذ مشابه لما يلي:

```
Welcome to the vi editor.  
It is a very powerful text editor.  
  
Especially for those who master it.  
~  
~
```

تنفيذ الأمر u

٢٣، ٢، ٩ الخطوة ٢٣

ضم سطرين، الحالي والذي يليه عن طريق كتابة حرف J:

J

وينبغي أن يكون التنفيذ مشابه لما يلي:

```
Welcome to the vi editor.  
It is a very powerful text editor. Especially for those who master it  
.  
~  
~
```

تنفيذ الأمر J

٢٤, ٢, ٩ الخطوة ٢٤

التراجع عن العملية الأخيرة:

u

وينبغي أن يكون التنفيذ مشابه لما يلي:

```
Welcome to the vi editor.  
It is a very powerful text editor.  
  
Especially for those who master it.  
~  
~
```

تنفيذ الأمر u

٢٥, ٢, ٩ الخطوة ٢٥

نسخ أو نثر الكلمة الحالية:

yw

عند نسخ النص، لا يحدث أي تغيير على الشاشة.

٢٦, ٢, ٩ الخطوة ٢٦

لصق (أو "وضع") كلمة منسوخة قبل المؤشر الحالي:

P

وينبغي أن يكون التنفيذ مشابه لما يلي:

```
Welcome to the vi editor.  
It is a very powerful powerful text editor.  
  
Especially for those who master it.  
~  
~
```

تنفيذ الأمر P

٢٧, ٢, ٩ الخطوة ٢٧

التراجع عن العملية الأخيرة:

u

وينبغي أن يكون التنفيذ مشابه لما يلي:

```
Welcome to the vi editor.  
It is a very powerful text editor.  
  
Especially for those who master it.  
~  
~
```

تنفيذ الأمر u

٢٨, ٢, ٩ الخطوة ٢٨

انتقل إلى السطر الأول، ثم ضم ثلاثة أسطر مع بعضها البعض:

1G

3J

وينبغي أن يكون التنفيذ مشابه لما يلي:

```
Welcome to the vi editor. It is a very powerful text editor. Especiall
y for those who
master it.
~
~
```

تنفيذ لمفاتيحي التحكم بالمؤشر 1G، 3J

٢٩، ٢، ٩ الخطوة ٢٩

التراجع عن العملية الأخيرة:

u

وينبغي أن يكون التنفيذ مشابه لما يلي:

```
Welcome to the vi editor.
It is a very powerful text editor.
Especially for those who master it.
~
~
```

تنفيذ الأمر u

٣٠، ٢، ٩ الخطوة ٣٠

البحث وحذف كلمة text (أضف مسافة قبل كلمة text و بعدها):



المفتاح	وظيفته
1G	اذهب إلى بداية الملف.
i	الدخول إلى وضع إدراج.
Hello and	إضافة نص إلى وثيقة مع مسافة بعد and.

جدول يصف الأوامر 1G، i، Hello and

وينبغي أن يكون التنفيذ مشابه لما يلي:

```
Hello and | welcome to the vi editor.
It is a very powerful text editor.
Especially for those who master it.
~
~
```

تنفيذ الأوامر 1G، i، Hello and

٣٢، ٢، ٩ الخطوة ٣٢

اخرج من وضع الإدراج الحالي وقم بالعودة إلى وضع الأوامر عن طريق الضغط على مفتاح الهروب  
:Escape key

< ESC >

٣٣، ٢، ٩ الخطوة ٣٣

التقدم بالمؤشر مسافة واحدة عن طريق الضغط على حرف | لوضع المؤشر على حرف w ثم الضغط على رمز  
~ تيلدا tilde عن طريق ( ` + Shift ):

المفتاح	الوظيفة
l	'L' صغيرة و التحرك مسافة واحدة إلى الأمام
~	Shift + ` لكتابة رمز التيلدا

جدول لوصف الأمرين l و ~

وينبغي أن يكون التنفيذ مشابه لما يلي:

```
Hello and welcome to the vi editor.
It is a very powerful text editor.
Especially for those who master it.
~
~
```

تنفيذ الأمرين l، ~

٣٤، ٢، ٩ الخطوة ٣٤

احفظ الملف واضغط على مفتاح الخروج Esc لضمان أنك في وضع الأوامر. ثم اكتب w: ثم اضغط على المفتاح ادخال:

```
:w
```

٣٥، ٢، ٩ الخطوة ٣٥

اضغط على Enter لتأكيد التغيير، لاحظ الرسالة في أسفل اليسار يشير إلى ملف قد تمت كتابته:

```
< Enter >
```

وينبغي أن يكون التنفيذ مشابه لما يلي:

```
~
~
"myfile" 3 lines, 102 characters written
```

تنفيذ الضغط على مفتاح الادخال

٣٦, ٢, ٩ الخطوة ٣٦

انتقل إلى الفراغ بين كلمة powerful وكلمة editor في السطر الثاني كما هو مبين في الصورة أدناه. يمكنك القيام بذلك بالضغط على z تليها 10l أو استخدام مفاتيح الأسهم:

المفتاح	الوظيفة
z	الانتقال إلى السطر في الاسفل
10l	١٠ تليها 'L' صغيرة

جدول يصف الأمرين z، 10l

وينبغي أن يكون التنفيذ مشابه لما يلي:

```
Hello and welcome to the vi editor.
It is a very powerful editor.

Especially for those who master it.
~
~
```

تنفيذ الأمرين z، 10l

٣٧, ٢, ٩ الخطوة ٣٧

أدرج كلمة text على يمين المؤشر عن طريق الضغط على حرف a. سوف يتحرك المؤشر إلى اليمين ويتم الانتقال إلى وضع الإدراج. اكتب كلمة text متبوعة بمسافة كما هو مبين في الصورة أدناه:

المفتاح	وظيفته
a	الدخول إلى وضع الإدراج
text	نص متبوعاً بمسافة

جدول يصف a، text

وينبغي أن يكون التنفيذ مشابه لما يلي:

```
Hello and welcome to the vi editor.
It is a very powerful text editor.

Especially for those who master it.
~
~
```

تنفيذ الأمرين a، text

٣٨، ٢، ٩ الخطوة ٣٨

الخروج من وضع الإدراج عن طريق الضغط على مفتاح هروب Esc.

< ESC >

٣٩، ٢، ٩ الخطوة ٣٩

انشئ سطر جديد فارغ تحت السطر الحالي عن طريق كتابة حرف o صغير:

o

وينبغي أن يكون التنفيذ مشابه لما يلي:

```
Hello and welcome to the vi editor.  
It is a very powerful text editor.  
█  
Especially for those who master it.  
~  
~
```

تنفيذ الأمر o

٤٠, ٢, ٩ الخطوة ٤٠

أدخل النص التالي:

```
This line added by pressing lowercase o.
```

وينبغي أن يكون التنفيذ مشابه لما يلي:

```
Hello and welcome to the vi editor.  
It is a very powerful text editor.  
This line was added by pressing lowercase o. █ Especially for those who m  
aster it.  
~  
~
```

تنفيذ o

٤١, ٢, ٩ الخطوة ٤١

الخروج من وضع الإدراج عن طريق الضغط على مفتاح هروب Esc.

```
< ESC >
```

٤٢, ٢, ٩ الخطوة ٤٢

انشاء سطر فارغ فوق السطر الحالي عن طريق الضغط على الحرف الكبير O:

O

وينبغي أن يكون التنفيذ مشابه لما يلي:

```
Hello and welcome to the vi editor.  
It is a very powerful text editor.  
  
This line was added by pressing lowercase o.  
Especially for those who master it.  
~  
~
```

O تنفيذ

٤٣, ٢, ٩ الخطوة ٤٣

أدخل النص التالي:

You just pressed O to open a line above.

وينبغي أن يكون التنفيذ مشابه لما يلي:

```
Hello and welcome to the vi editor.
It is a very powerful text editor.
You just pressed O to open a line above.
This line was added by pressing lowercase o.
Especially for those who master it.
```

### تنفيذ النص السابق

٤٤, ٢, ٩ الخطوة ٤٤

الخروج من وضع الإدراج عن طريق الضغط على مفتاح خروج.

٤٥, ٢, ٩ الخطوة ٤٥

احفظ الملف واغلق المحرر vi editor باستخدام أي من الطرق التالية والتي تحفظ التغييرات:

الأمر	الوظيفة/ المفتاح
:x	حفظ وإغلاق الملف.
:wq	للكتابة على ملف والخروج.
:wq!	للكتابة على ملف للقراءة فقط، إذا كان ذلك ممكناً والخروج.
ZZ	حفظ وإغلاق. لاحظ أنه لا يتم استخدام نقطتين : في هذه الحالة.
:q!	الخروج من غير حفظ
:e!	تجاهل التغييرات وإعادة تحميل الملف.
:w!	للكتابة على ملف للقراءة فقط، إذا كان ذلك ممكناً.

جدول يصف أوامر لإغلاق المحرر

٤٦، ٢، ٩ الخطوة ٤٦

مرة أخرى قم بفتح myfile باستخدام المحرر vi editor:

٤٧، ٢، ٩ الخطوة ٤٧

انتقل إلى السطر الثالث، ثم احذف السطر الثالث والرابع:

```
3G
```

```
2dd
```

وينبغي أن يكون التنفيذ مشابه لما يلي:

```
Hello and welcome to the vi editor.  
It is a very powerful text editor.  
Especially for those who master it.  
~  
~
```

تنفيذ الأمرين 3G، 2dd

٤٨، ٢، ٩ الخطوة ٤٨

اضغط على مفتاح Esc للتأكد أنك في وضع الأوامر.

```
< ESC >
```

٤٩، ٢، ٩ الخطوة ٤٩

قم بإنهاء محرر vi دون حفظ التغييرات:

:q!

٥٠، ٢، ٩ الخطوة

افتح الملف myfile مع المحرر vi:

vi myfile

لاحظ أن الأسطر ٣ و ٤ لا تزال موجودة.

٥١، ٢، ٩ الخطوة

ابحث عن الكلمة line. ستلاحظ تحرك المؤشر إلى بداية كلمة line كما هو مبين في الصورة أدناه:

/line

وينبغي أن يكون التنفيذ مشابه لما يلي:



```
Hello and welcome to the vi editor.  
It is a very powerful text editor.  
You just pressed O to open a line above.  
This |line was added by pressing lowercase o.  
Especially for those who master it.
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
/ line
```

تنفيذ n

٥٣, ٢, ٩ الخطوة ٥٣

ابحث عن النتيجة السابقة لكلمة line. ستلاحظ تحرك المؤشر إلى بداية التشابه السابق لكلمة line كما يظهر أدناه:

```
?line
```

وينبغي أن يكون التنفيذ مشابه لما يلي:

```
Hello and welcome to the vi editor.  
It is a very powerful text editor.  
You just pressed O to open a |line above.  
This line was added by pressing lowercase o.  
Especially for those who master it.
```

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

?line

تنفيذ الأمر ?line

٥٤, ٢, ٩ الخطوة ٥٤

ابحث عن التشابه السابق لكلمة line عن طريق الضغط على حرف n. وبما أنه لا يوجد تطابق في هذا الاتجاه

الخلفي فان المحرر vi سوف يلتف حول الوثيقة:

n

وينبغي أن يكون التنفيذ مشابه لما يلي:

```
Hello and welcome to the vi editor.  
It is a very powerful text editor.  
You just pressed O to open a line above.  
This line was added by pressing lowercase o.  
Especially for those who master it.
```

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

search hit TOP, continuing at BOTTOM

تنفيذ الأمر n

٥٥،٢،٩ الخطوة ٥٥

سيتم استبدال كلمة line مع كلمة entry. عند الضغط على cw فإن كلمة line التي عليها المؤشر سوف تختفي، وسيكون المستخدم في وضع الإدراج:

cw

entry

وينبغي أن يكون التنفيذ مشابه لما يلي:

```
Hello and welcome to the vi editor.  
It is a very powerful text editor.  
You just pressed O to open a line above.  
This entry was added by pressing lowercase o.  
Especially for those who master it.  
~  
~
```

تنفيذ الأمرين cw ، entry

٥٦، ٢، ٩ الخطوة ٥٦

اضغط على مفتاح ESC للخروج من وضع الإدراج.

< ESC >

٥٧، ٢، ٩ الخطوة ٥٧

أضف نص في بداية السطر. أدخل إلى وضع الإدراج مرة أخرى وأضف سطر عن طريق الضغط على الحرف الكبير من i:

|

وينبغي أن يكون التنفيذ مشابه لما يلي:

```
Hello and welcome to the vi editor.  
It is a very powerful text editor.  
You just pressed O to open a line above.  
This entry was added by pressing lowercase o.  
Especially for those who master it.  
~  
~
```

### تنفيذ الأمر I

ويتشمل وضع الإدراج على: i، I، a، A، o، O و.

٥٨، ٢، ٩ الخطوة ٥٨

اضغط على مفتاح ESC للعودة إلى وضع الأوامر.

< ESC >

٥٩، ٢، ٩ الخطوة ٥٩

أضف نص في نهاية السطر (الحرف الكبير A). أولاً تحرك إلى السطر الثاني واضف عبارة في Indeed!:

2G

A

[Space]Indeed!

اضغط على مفتاح ESC للعودة إلى وضع الأوامر.

< ESC >

الشاشة يجب أن تشبه ما يلي:

```
Hello and welcome to the vi editor.  
It is a very powerful text editor. Indeed!  
You just pressed O to open a line above.  
This entry was added by pressing lowercase o.  
Especially for those who master it.
```

~

~

تنفيذ الأوامر 2G، A، [Space]Indeed!

٦٠، ٢، ٩ الخطوة ٦٠

حفظ التغييرات والخروج من المحرر vi:

:X

٩، ٣ برمجة القشرة الأساسية

برمجة القشرة تسمح بأخذ سلسلة معقدة من الأوامر ووضعها في ملف ثم القيام بتشغيل ذلك الملف كبرنامج؛ مما يوفر الكثير من الوقت لعدم الحاجة لكتابة سلاسل طويلة من الأوامر المستخدمة بشكل روتيني.

سوف يركز هذا التمرين على كيفية إنشاء برنامج نصي بسيط للقشرة simple shell scripts، لإتمام ذلك فمن المفترض أن تعرف كيفية استخدام محرر النصوص. لا تتردد في اختيار محرر مثل: vi، nano، gedit أو أي محرر آخر تريده.

## ٩,٣,١ الخطوة ١

لإنشاء برنامج نصي بسيط للقشرة، تحتاج إلى إنشاء ملف نصي وإضافة أوامر عليه. قم بإنشاء ملف يسمى `sample.sh` واضف الأسطر التالية:

```
vi sample.sh

i

echo "Hello there! Here is the calendar for this month:"

cal
```

وينبغي أن يكون التنفيذ مشابه لما يلي:

```
echo "Hello there! Here is the calendar for this month:"
cal
~
~
```

تنفيذ "echo "Hello there! Here is the calendar for this month:" cal

## ٩,٣,٢ الخطوة ٢

للتوضيح، هذا برنامج نصي لقشرة الباش `BASH shell script`، تحتاج إلى تضمين سطر خاص في الجزء العلوي من ملف يسمى "`shbang`" (أو `shebang`). يبدأ هذا السطر مع `#!` ثم يحتوي على المسار إلى قشرة الباش القابلة للتنفيذ. أضف السطر التالي في أعلى الملف `sample.sh`:

```
<ESC>
```

ثم بعد ذلك قم بكتابة التالي:

```
1G
```

```
I
```

```
#!/bin/bash
```

وينبغي أن يكون التنفيذ مشابه لما يلي:

```
#!/bin/bash
echo "Hello there! Here is the calendar for this month:"
cal
~
~
```

تنفيذ الأمر `#!/bin/bash`

٣,٣,٩ الخطوة ٣

إحدى الطرق التي يمكنك بها تشغيل هذا البرنامج هي عن طريق كتابة `bash` قبل اسم الملف. نفذ ما يلي لتنفيذ البرنامج:

```
<ESC>
:x
<Enter>
bash sample.sh
```

وينبغي أن يكون التنفيذ مشابه لما يلي:

```
sysadmin@localhost:~$ bash sample.sh
Hello there! Here is the calendar for this month:

    April 2016
Su Mo Tu We Th Fr Sa
                1  2
 3  4  5  6  7  8  9
10 11 12 13 14 15 16
17 18 19 20 21 22 23
24 25 26 27 28 29 30

sysadmin@localhost:~$
```

تنفيذ الأمر `bash sample.sh`

٤, ٣, ٩ الخطوة

يمكنك تجنب الحاجة إلى كتابة `bash` أمام اسم الملف عن طريق جعل الملف قابل للتنفيذ `executable` لجميع المستخدمين. اكتب الأوامر التالية:

```
ls -l sample.sh

chmod a+x sample.sh

ls -l sample.sh

./sample.sh
```

وينبغي أن يكون التنفيذ مشابه لما يلي:

```

sysadmin@localhost:~$ ls -l sample.sh
-rw-rw-r-- 1 sysadmin sysadmin 73 Apr  9 22:44 sample.sh
sysadmin@localhost:~$ chmod a+x sample.sh
sysadmin@localhost:~$ ./sample.sh
Hello there! Here is the calendar for this month:

    April 2016
Su Mo Tu We Th Fr Sa
                1  2
 3  4  5  6  7  8  9
10 11 12 13 14 15 16
17 18 19 20 21 22 23
24 25 26 27 28 29 30

sysadmin@localhost:~$

```

تنفيذ الأوامر `./sample.sh` ، `ls -l sample.sh` ، `chmod a+x sample.sh` ، `ls -l sample.sh`

تم استخدام الأمر `chmod` لتغيير الأذونات على الملف بحيث يمكن تنفيذ الملف (ستتعلم المزيد عن هذا الأمر في معاملة لاحقة).

٥،٣،٩ الخطوة

وثمة سمة مشتركة تستخدم في البرمجة هي "backquoting". ومع هذه التقنية؛ يمكنك تشغيل أمر القشرة داخل قشرة أخرى. حيث سيتم إرجاع تنفيذ الأمر الداخلي كمعطيات `argument` لأمر خارجي. أضف ما يلي إلى أسفل الملف `sample.sh`:

```
vi sample.sh
```

ثم بعد ذلك أضف التالي:

```
3G
```

```
O
```

```
echo "Today is" `date +%A`
```

ثم بعد ذلك قم بالخروج من وضع الإدراج:

```
<ESC>
```

وأخيرا قم بحفظ عملك والخروج من محرر النصوص:

```
:X
```

والآن نفذ التالي لعرض محتويات الملف ثم تشغيله:

```
cat sample.sh
```

وينبغي أن يكون التنفيذ كالتالي:

```
sysadmin@localhost:~$ cat sample.sh
#!/bin/bash
echo "Hello there! Here is the calendar for this month:"
cal
echo "Today is" `date +%A`
sysadmin@localhost:~$
```

قم بتشغيل برنامج sample.sh كالتالي:

```
./sample.sh
```

وينبغي أن يكون التنفيذ مشابه لما يلي:

```

sysadmin@localhost:~$ ./sample.sh
Hello there! Here is the calendar for this month:

    April 2016
Su Mo Tu We Th Fr Sa
                1  2
 3  4  5  6  7  8  9
10 11 12 13 14 15 16
17 18 19 20 21 22 23
24 25 26 27 28 29 30

Today is Friday
sysadmin@localhost:~$

```

تنفيذ الأوامر `./sample.sh` ، `cat sample.sh` ، `echo "Today is" `date +%A``

٦، ٣، ٩ الخطوة ٦

لقد تم استخدام `./` أمام اسم الملف `sample.sh` للإشارة إلى أن الملف موجود في الدليل الحالي. نفذ ما يلي لترى كيف سوف تفشل القشرة في العثور على الملف إذا كنت لا تستخدم الرموز `./` :

```
sample.sh
```

وينبغي أن يكون التنفيذ مشابه لما يلي:

```

sysadmin@localhost:~$ sample.sh
-bash: sample.sh: command not found
sysadmin@localhost:~$

```

تنفيذ الأمر `sample.sh`

## ٧,٣,٧ الخطوة

تذكر أن المتغير \$PATH يستخدم للبحث عن الأوامر التي كتبتها. نفذ ما يلي لروية المتغير \$PATH لحساب  
:sysadmin

```
echo $PATH
```

وينبغي أن يكون التنفيذ مشابه لما يلي:

```
sysadmin@localhost:~$ echo $PATH
/home/sysadmin/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/s
bin:/bin:/usr/games
sysadmin@localhost:~$
```

تنفيذ الأمر echo \$PATH

## ٨,٣,٨ الخطوة

لاحظ أن /home/sysadmin/bin هو واحد من الدلائل في المتغير \$PATH. وهو مكان ممتاز لوضع  
النصوص القشرة الخاص بك:

```
mkdir bin
```

```
mv sample.sh bin
```

```
sample.sh
```

وينبغي أن يكون التنفيذ مشابه لما يلي:

```
sysadmin@localhost:~$ mkdir bin
sysadmin@localhost:~$ mv sample.sh bin
sysadmin@localhost:~$ sample.sh

Hello there! Here is the calendar for this month:

      April 2016
Su Mo Tu We Th Fr Sa
                1  2
 3  4  5  6  7  8  9
10 11 12 13 14 15 16
17 18 19 20 21 22 23
24 25 26 27 28 29 30

Today is Friday
sysadmin@localhost:~$
```

تنفيذ الأوامر 'mkdir bin' ، 'mv sample.sh bin' ، 'sample.sh'

## ٩,٤ التنفيذ المشروط والمتكرر

لاحظ أن الأمثلة examples التي ستعرض demonstrated في هذا القسم أكثر تعقيداً .more complex . استخدم تقنية technique أو اسلوب لوصف describe ما يحدث في البرنامج program . وسيكون هذا الاسلوب مشابه لما يلي:

أدخل هذا العامود إلى drive.sh	يصف هذا العامود الكود (لن يدخل في الملف)
echo "الرجاء إدخال عمرك"	# طباعة الموجه
قراءة العمر	# قراءة إدخال المستخدم ومكان المتغير \$age

جدول لما يحدث في البرنامج السابق

عند اتباع الإرشادات المقدمة instructions provided، فأنت الآن تقوم بإدخال النص enter text من العامود الأيمن right column في الملف المحدد (drive.sh في المثال أعلاه). أما العامود الأيسر left column فهو يستخدم لوصف describe أسطر البرنامج program lines. يتم استخدام علامة الشبكة (#) لأنه في البرمجة النصية للقشرة shell script يمكنك وضع تعليقات comments ضمن البرنامج program باستخدام علامة #.

٩,٤,١ الخطوة ١

البرامج النصية Scripts المركبة complex يمكن ان يكون فيها تنفيذ شرطي conditional execution. التنفيذ الشرطي مثل جملة إذا الشرطية if statement، ويمكن الاستفادة من نتائج أمر يسمى اختبار test. جملة الاختبار test statement تقارن compares بين رقمين أو سلسلتين لأشياء محددة، مثلاً يساوي equal to أو أقل من less than.

أنشأ الملف التالي (drive.sh) واجعله قابل للتنفيذ لترى كيف تعمل جملة إذا الشرطية if وجملة الاختبار test. أبدأ من خلال وضع ما يلي في الملف drive.sh:

أدخل هذا العامود إلى drive.sh	يصف هذا العامود الكود (لن يكتب في الملف)
#!/bin/bash	يحتوي هذا السطر على المسار إلى قشرة الباش القابلة للتنفيذ
echo "Please enter your age"	# طباعة الموجه
read age	# قراءة إدخال المستخدم وضعه في متغير \$age
if test \$age -lt 16	# اختبر المتغير \$age إذا اعداد ١٦ فهو صحيح
then	وإذا اعداد عدد أقل من ١٦
echo "You are not old enough to drive."	# نفذ اذا كانت نتيجة الاختبار صحيح
else	
echo "You can drive!"	# نفذ اذا كانت نتيجة الاختبار خاطئة
fi	# انتهاء الجملة

جدول يصف برنامج للرخصة

لإنشاء الملف؛ اكتب الامر التالي:

```
vi drive.sh
```

وللدخول لوضع الادراج؛ اكتب حرف i:

```
i
```

ثم بعد ذلك؛ نقوم بكتابة البرنامج:

```
#!/bin/bash

echo "Please enter your age"

read age

if test $age -lt 16

then

echo "You are not old enough to drive."

else

echo "You can drive!"

fi

<ESC>

:x

<Enter>
```

لتشغيل هذا الملف؛ نفذ الأوامر التالية:

```
cat drive.sh

chmod a+x drive.sh

./drive.sh
```

وينبغي أن يكون التنفيذ مشابه لما يلي:

```

sysadmin@localhost:~$ cat drive.sh
#!/bin/bash
echo "Please enter your age"
read age
if test $age -lt 16
then
    echo "You are not old enough to drive."
else
    echo "You can drive!"
fi
sysadmin@localhost:~$ chmod a+x drive.sh
sysadmin@localhost:~$ ./drive.sh
Please enter your age
14
You are not old enough to drive.
sysadmin@localhost:~$

```

تنفيذ الأوامر `./drive.sh` ، `chmod a+x drive.sh` ، `cat drive.sh`

تستطيع قراءة الجملة إذا الشرطية وهي "إذا كان العمر `$age` أقل من ١٦ عاماً، أظهر رسالة أنت لست بالعمر الكافي للقيادة؛ وإلا أظهر الرسالة يمكنك قيادة". والرمز `fi` يُنهي جملة إذا الشرطية.

ملاحظة: `$age` يجب أن يكون قيمة عددية. إذا لم يكن كذلك، سوف يتعطل البرنامج.

٢،٤،٩ الخطوة ٢

جملة الاختبار تُستدعى تلقائياً عند وضع معطيات `arguments` بين قوسين معقوفين (`[ ]`) ويحيط بتلك المعطيات مسافة بينها وبين الأقواس. عدل جملة إذا في الملف `drive.sh` بحيث تبدو كما يلي:

```
vi drive.sh
```

```
4G
```

```
dd
```

```
o
```

```
if [ $age -lt 16 ]
```

```
<ESC>
```

```
:x
```

```
<Enter>
```

اعد تشغيل البرنامج مرة اخرى:

```
cat drive.sh
```

```
./drive.sh
```

وينبغي أن يكون التنفيذ مشابه لما يلي:

```

sysadmin@localhost:~$ cat drive.sh
#!/bin/bash
echo "Please enter your age"
read age
if [ $age -lt 16 ]
then
    echo "You are not old enough to drive."
else
    echo "You can drive!"
fi
sysadmin@localhost:~$ ./drive.sh
Please enter your age
21
You can drive!
sysadmin@localhost:~$

```

تنفيذ الأوامر `./drive.sh` ، `cat drive.sh` ، `if [ $age -lt 16 ]`

للاطلاع على القائمة كاملة لجملته الاختبار، شغل الأمر `man test`.

هام: يجب أن يكون هناك مسافات حول الأقواس المربعة. فمثلاً الكتابة بهذا الشكل `[ $age -lt 16 ]` سوف تفشل، ولكن عند ترك مسافة بهذه الطريقة `[ $age -lt 16 ]` فإن الجملة ستعمل.

٣,٤,٩ الخطوة ٣

يمكنك أيضا استخدام نتائج أوامر قشرة أخرى لإرجاع نتيجة بالنجاح أو بالفشل. على سبيل المثال، أنشأ وشغل البرنامج التالي، والذي يستخدم لتحديد ما إذا كان حساب المستخدم موجود على هذا النظام أو لا. أضف ما يلي إلى الملف `check.sh`:

```
vi chech.sh  
  
i  
  
#!/bin/bash  
  
echo "Enter a username to check: "  
  
read name  
  
if grep $name /etc/passwd > /dev/null  
  
then  
  
echo "$name is on this system"  
  
else  
  
echo "$name does not exist"  
  
fi  
  
<ESC>  
  
:X  
  
<Enter>  
  
cat chech.sh
```

غير الاذونات بالأمر التالي:

```
chmod a+x check.sh
```

الآن؛ شغل البرنامج عن طريق الأوامر التالية:

```
./check.sh
```

```
root
```

```
./check.sh
```

```
bobby
```

عند السؤال للحصول على اسم المستخدم، اعطي اسم `root`. نفذ الأمر `./check.sh` مرة أخرى وأعطي اسم `bobby`. الشاشة يجب أن تبدو كما يلي:

```
sysadmin@localhost:~$ cat check.sh
#!/bin/bash
echo "Enter a username to check: "
read name
if grep $name /etc/passwd > /dev/null
then
    echo "$name is on this system"
else
    echo "$name does not exist"
fi
sysadmin@localhost:~$ chmod a+x check.sh
sysadmin@localhost:~$ ./check.sh
Enter a username to check:
root
root is on this system
sysadmin@localhost:~$ ./check.sh
Enter a username to check:
bobby
bobby does not exist
sysadmin@localhost:~$
```

تنفيذ البرنامج السابق

## ٤, ٤, ٩ الخطوة ٤

جملة شرطية أخرى وهي حلقة بينما `while loop`. تستخدم هذه الجملة لتنفيذ التعليمات البرمجية بشكل متكرر طالما يرجع الشرط بنتيجة صحيحة. أبدأ من خلال كتابة ما يلي في الملف المسمى `num.sh`:

يصف هذا العامود الكود (لن يُكتب في الملف)	أدخل التالي إلى <code>num.sh</code>
	<code>#!/bin/bash</code>
	<code>echo "Please enter a number greater than 100"</code>
	<code>read num</code>
# تنفيذ التعليمات البرمجية من "do" إلى "نفذ إذا حالة الاختبار حقيقية	<code>while [ \$num -le 100 ]</code>
	<code>do</code>
	<code>echo "\$num is NOT greater than 100."</code>
	<code>echo "Please enter a number greater than 100"</code>
	<code>read num</code>
# نهاية جملة الشرط	<code>done</code>
	<code>\$num is greater than ،echo "Finally 100"</code>

### جدول يصف أوامر حلقة `loop`

عند السؤال للحصول على رقم أدخل الرقم ٢٥. وعند السؤال في المرة التالية أدخل الرقم ٩٩. وأخيرا أدخل ١٠١ عند المطالبة للحصول على رقم للمرة الثالثة.

ادخل البرنامج التالي في الملف `num.sh`:

```
vi num.sh  
  
i  
  
#!/bin/bash  
  
echo "Please enter a number greater than 100"  
  
read num  
  
while [ $num -le 100 ]  
  
do  
  
echo "$num is NOT greater than 100."  
  
echo "Please enter a number greater than 100."  
  
read num  
  
done  
  
echo "Finally, $num is greater than 100"  
  
<ESC>  
  
:X  
  
<Enter>  
  
cat num.sh  
  
chmod a+x num.sh  
  
./num.sh
```

الآن، قم بإدخال الأرقام التالية:

25

99

101

الشاشة يجب أن تبدو كما يلي:

```
sysadmin@localhost:~$ cat num.sh
#!/bin/bash
echo "Please enter a number greater than 100"
read num
while [ $num -le 100 ]
do
    echo "$num is NOT greater than 100."
    echo "Please enter a number greater than 100."
    read num
done
echo "Finally, $num is greater than 100"
sysadmin@localhost:~$ chmod a+x num.sh
sysadmin@localhost:~$ ./num.sh

Please enter a number greater than 100
25
25 is NOT greater than 100.
Please enter a number greater than 100.
99
99 is NOT greater than 100.
Please enter a number greater than 100.
101
Finally, 101 is greater than 100
sysadmin@localhost:~$
```

تنفيذ برنامج loop

إذا رجعت قيمة الشرط صحيحة للجملة ( [ \$num -le 100 ] ) فسيتم عندها تنفيذ الجملة بين do و done. بمجرد الانتهاء من تنفيذ تلك الجملة، يتم فحص جملة الشرط من جديد. إذا كانت النتيجة صحيحة مرة أخرى، يتم التنفيذ مرة أخرى للجملة بين do و done.

وبالتالي سيستمر التكرار حتى ترجع النتيجة خاطئة، أو عندما تكون القيمة أكبر من 100.

٥,٤,٩ الخطوة ٥

البرمجية النصية Scripting code هي جزء من قشرة الباش BASH shell، مما يعني أنه يمكنك استخدام هذه الجملة statements في سطر الأوامر command line مثل ما كنت تستخدمها في المحرر vi الخاص بالقشرة. عملية البرمجة النصية مفيدة جداً لجملة مثل جملة for؛ التي من شأنها اسناد assign قائمة من القيم list of values في وقت واحد إلى متغير variable. والذي من شأنه إجراء مجموعة من العمليات a set of operations على كل قيمة value من تلك القيم. على سبيل المثال، اكتب الأوامر التالية:

```
for name in /etc/passwd /etc/hosts /etc/group
do
wc $name
done
```

وينبغي أن يكون التنفيذ مشابه لما يلي:

```

sysadmin@localhost:~$ for name in /etc/passwd /etc/hosts /etc/group
> do
>     wc $name
> done
 24  30 1001 /etc/passwd
  7  15  161 /etc/hosts
 46  46  561 /etc/group
sysadmin@localhost:~$

```

### تنفيذ جملة for

لاحظ أن الأمر `wc` تم تشغيله ثلاث مرات: مرة مع `/etc/passwd` ومرة مع `/etc/hosts` ومرة أخيرة مع `/etc/group`.

٦، ٤، ٩ الخطوة ٦

غالباً ما يتم استخدام الأمر `seq` بالتعاون مع جملة `for`؛ حيث أن الأمر `seq` يمكن ان يولد قائمة من قيم الاعداد الصحيحة، على سبيل المثال من ١ إلى ١٠. مثلاً، قم بتنفيذ البرنامج التالي في سطر الأوامر لإنشاء ١٢ ملف تبدأ اسمها بالاسم `test1 test2 test3` وتنتهي بالاسم `test12`:

```

ls

for num in `seq 1 12`
do
touch test$num
done

ls

```

ويكون التنفيذ كالتالي:

```
sysadmin@localhost:~$ ls
Desktop  Downloads  Pictures  Templates  check.sh  num.sh
Documents Music      Public   Videos    drive.sh

sysadmin@localhost:~$ for num in `seq 1 12`
> do
> touch test$num
> done

sysadmin@localhost:~$ ls
Desktop  Music  Templates  drive.sh  test10  test2  test5  test8
Documents Pictures Videos    num.sh   test11  test3  test6  test9
Downloads Public   check.sh  test1    test12  test4  test7

sysadmin@localhost:~$
```

تنفيذ جملة seq

## المعمل العاشر – فهم عتاد الحاسب

# Understanding Computer Hardware

## ١٠,١ مقدمة

هذا هو المعمل العاشر: فهم عتاد الحاسب Computer Hardware. في هذا المعمل سيتم شرح أوامر تعرض معلومات عن مكونات جهاز الحاسب الآلي.

في هذا التمرين، سوف نقوم بالمهام التالية:

- استخدم الأوامر لعرض قائمة العتاد.

## ١٠,٢ قائمة عتاد الحاسب

في هذه المهمة، سوف نقوم بتنفيذ execute بعض الأوامر commands؛ ودراسة بعض الملفات files وذلك لعرض display تكوين الأجهزة hardware configuration في الجهاز الذي تعمل عليه.

### ١٠,٢,١ الخطوة ١

نفذ الأمر من أجل تحديد نوع وحدة المعالجة المركزية:

```
lscpu
```

وينبغي أن يكون التنفيذ مشابه لما يلي:

```
sysadmin@localhost:~$ lscpu
Architecture:          x86_64
CPU op-mode(s):      32-bit, 64-bit
Byte Order:           Little Endian
CPU(s):               24
On-line CPU(s) list: 0-23
Thread(s) per core:  2
Core(s) per socket:  6
Socket(s):            2
NUMA node(s):        2
Vendor ID:            GenuineIntel
CPU family:           6
Model:                45
Stepping:             6
CPU MHz:              1412.265
BogoMIPS:              4001.62
Virtualization:       VT-x
L1d cache:            32K
L1i cache:            32K
L2 cache:             256K
L3 cache:             15360K
NUMA node0 CPU(s):   0-5,12-17
NUMA node1 CPU(s):   6-11,18-23
sysadmin@localhost:~$
```

تنفيذ الأمر lscpu

إن عرض display معلومات عن وحدة المعالجة المركزية (CPU) - Central Processing Unit مهم عند محاولة تحديد determine مدى إمكانية الاستفادة من مميزات لينكس المتقدمة advanced Linux features وذلك بغرض استخدامها used على نظامك system. لمزيد من التفاصيل details حول وحدة المعالجة المركزية الخاصة بك CPU(s) يمكنك فحص examine الملف المسمى /proc/cpuinfo ؛ وتحديداً الأعلام flags والتي يتم سردها listed لرؤية تفاصيل وحدة المعالجة المركزية الخاصة بك وبالتالي معرفة هل توجد مميزات أولاً.

٢,٢,١٠ الخطوة ٢

عرض ملف /proc/cpuinfo:

```
cat /proc/cpuinfo
```

وينبغي أن يكون التنفيذ مشابه لما يلي:

```

sysadmin@localhost:~$ cat /proc/cpuinfo
physical id      : 1

siblings        : 12

core id         : 5

cpu cores       : 6

apicid          : 43

initial apicid  : 43

fpu             : yes

fpu_exception   : yes

cpuid level     : 13

wp              : yes

flags           : fpu vme de pse tsc msr pae mce cx8 apic sep mtrr
pge mca cmov pat pse36 clflush dts acpi mmx fxsr sse sse2 ss ht tm pbe
syscall nx pdpe1gb rdt
scp lm constant_tsc arch_perfmon pebs bts rep_good nopl xtopology nons
top_tsc ap
erfmpperf eagerfpu pni pclmulqdq dtes64 monitor ds_cpl vmx smx est tm2
ssse3 cx16 xtpr pdcm pcid dca sse4_1 sse4_2 x2apic popcnt tsc_deadlin
e_timer aes xsave avx lahf_lm epb tpr_shadow vnmi flexpriority ept vpi
d xsaveopt dtherm arat pln pts
bugs           :

bogomips        : 4001.57

clflush size    : 64

cache_alignment : 64

address sizes   : 46 bits physical, 48 bits virtual

power management:

sysadmin@localhost:~$

```

تنفيذ الأمر cat /proc/cpuinfo

٣, ٢, ١٠ الخطوة ٣

لاكتشاف مقدار ذاكرة الوصول العشوائي RAM ومساحة التبادل swap space المستخدمه في جهازك، استخدم الامر free:

```
free -m
```

```
free -g
```

عند استخدام الخيار -m مع الأمر free فسيتم اظهار مقدار الذاكرة بالميغا بايت megabytes؛ أما في حالة استخدام الخيار -g فسيتم عرض حجم الذاكرة بالغيغابايت gigabytes.

وينبغي ان يكون التنفيذ كما يلي:

```
sysadmin@localhost:~$ free -m
              total        used         free   shared    buffers     cached
Mem:          64394        35057        29337         0         1100         2631
-
/+ buffers/cache:        31326        33068
Swap:           975           54           921

sysadmin@localhost:~$ free -g
              total        used         free   shared    buffers     cached
Mem:             62           34           28         0           1           2
-
/+ buffers/cache:           30           32
Swap:              0           0           0

sysadmin@localhost:~$
```

تنفيذ الأمرين free -m، free -g

في التنفيذ أعلاه، يمكنك أن ترى أن النظام system لديه ١٦٠٤٩ ميغا بايت (ما يقرب من ١٥ غيغابايت) من الذاكرة الفعلية (RAM) physical memory المستخدمة. المستخدم used منها فقط ١٠٦٦ ميغا بايت، وهو مؤشر جيد good sign أن لديك ذاكرة كافية enough لاحتياجات النظام system's needs الخاص بك في الوقت الحالي.

في حال نفاذ الذاكرة run out؛ يتم استخدام التبادل. تقنية التبادل Swap تعني اخذ مساحة space من القرص الصلب hard drive واستخدامها مؤقتًا لتخزين البيانات temporarily store data التي من المفترض أن تكون مخزنة stored في ذاكرة الوصول العشوائي RAM.

٤, ٢, ١٠ الخطوة ٤

لمعرفة examine ما هي الأجهزة devices المتصلة connected بناقل PCI bus ، استخدم الأمر التالي lspci:

```
lspci
```

لاحظ في التنفيذ الجزئي partial output أدناه؛ أن العديد من الأجهزة many devices متصلة connected بلوحة النظام system board، كما هو معروض:

```
sysadmin@localhost:~$ lspci

00:00.0 Host bridge: Intel Corporation Xeon E5/Core i7 DMI2 (rev 07)

00:01.0 PCI bridge: Intel Corporation Xeon E5/Core i7 IIO PCI Express
Root Port 1a (rev 07)

00:01.1 PCI bridge: Intel Corporation Xeon E5/Core i7 IIO PCI Express
Root Port 1b (rev 07)

00:02.0 PCI bridge: Intel Corporation Xeon E5/Core i7 IIO PCI Express
Root Port 2a (rev 07)

00:03.0 PCI bridge: Intel Corporation Xeon E5/Core i7 IIO PCI Express
Root Port
3a in PCI Express Mode (rev 07)

00:04.0 System peripheral: Intel Corporation Xeon E5/Core i7 DMA
Channel 0 (rev 07)

00:04.1 System peripheral: Intel Corporation Xeon E5/Core i7 DMA
Channel 1 (rev 07)

00:04.2 System peripheral: Intel Corporation Xeon E5/Core i7 DMA
Channel 2 (rev 07)

00:04.3 System peripheral: Intel Corporation Xeon E5/Core i7 DMA
Channel 3 (rev 07)

00:04.4 System peripheral: Intel Corporation Xeon E5/Core i7 DMA
Channel 4 (rev 07)

00:04.5 System peripheral: Intel Corporation Xeon E5/Core i7 DMA
Channel 5 (rev 07)
```

### تنفيذ الأمر lspci

نتائج الأمر `lspci` هامة لتحديد الأجهزة التي لا تدعمها نواة لينكس. بعض الأجهزة مثل بطاقات الفيديو `video cards` قد لا توفر سوى الوظائف الأساسية `basic functionality` إذا لم يتم تثبيت `install` برنامج التشغيل المناسب لها.

٥,٢,١٠ الخطوة ٥

استخدم الخيار `-k` مع الأمر `lspci` لإظهار الأجهزة مع مشغل النواة `kernel driver` والوحدات المستخدمة:

lspci -k

وينبغي ان يكون التنفيذ كما يلي:

```
sysadmin@localhost:~$ lspci -k
00:00.0 Host bridge: Intel Corporation Xeon E5/Core i7 DMI2 (rev 07)
        Subsystem: Super Micro Computer Inc Device 0626

00:01.0 PCI bridge: Intel Corporation Xeon E5/Core i7 IIO PCI Express
Root Port
1a (rev 07)

        Kernel driver in use: pcieport

00:01.1 PCI bridge: Intel Corporation Xeon E5/Core i7 IIO PCI Express
Root Port
1b (rev 07)

        Kernel driver in use: pcieport

00:02.0 PCI bridge: Intel Corporation Xeon E5/Core i7 IIO PCI Express
Root Port
2a (rev 07)

        Kernel driver in use: pcieport

00:03.0 PCI bridge: Intel Corporation Xeon E5/Core i7 IIO PCI Express
Root Port
3a in PCI Express Mode (rev 07)

        Kernel driver in use: pcieport

00:04.0 System peripheral: Intel Corporation Xeon E5/Core i7 DMA
Channel 0 (rev 07)

        Subsystem: Super Micro Computer Inc Device 0626

        Kernel driver in use: ioatdma

00:04.1 System peripheral: Intel Corporation Xeon E5/Core i7 DMA
Channel 1 (rev 07)
```

تنفيذ الأمر `lspci -k`

## ٦, ٢, ١٠ الخطوة ٦

لسرد قائمة بالأجهزة المتصلة بالنظام عبر المنفذ التسلسلي العالمي (USB) – Universal Serial Bus؛  
استخدم الامر التالي:

```
lsusb
```

لاحظ أن الإخراج التالي مختلف؛ حيث لم يتم الكشف عن أي أجهزة USB متصلة بالنظام:

```
sysadmin@localhost:~$ lsusb
unable to initialize libusb: -99
sysadmin@localhost:~$
```

تنفيذ الأمر lsusb لنظام تشغيل افتراضي

ذلك لأن النظام السابق هو نظام افتراضي virtualized؛ وفيه قد لا تظهر أجهزة USB المتصلة بالنظام كما يظهر عادة عند تنفيذ الأمر lsusb في الأجهزة العادية. والآن؛ سنجرّب الأمر السابق في أحد الأنظمة العادية؛ وستكون النتيجة فيه كالتالي:

```
sysadmin@localhost:~$ lsusb
Bus 001 Device 001: ID 1d6b:0001 Linux Foundation 1.1 root hub
sysadmin@localhost:~$
```

تنفيذ الأمر lsusb لنظام تشغيل حقيقي

HAL هو طبقة العتاد المجردة Hardware Abstraction Layer ( والتي هي عبارة عن مجموعة من الإجراءات الروتينية في البرامج التي تحاكي بعض التفاصيل الخاصة بالنظام الأساسي، وبالتالي تسمح لبرامج سطح المكتب من اكتشاف موارد الجهاز والوصول إليها). البرنامج الخفي HAL لـ daemon هو hald (البرنامج الخفي هو الذي يعمل في الخلفية بغير إدراك من المستخدم العادي)، يقوم hald بتحديث update قاعدة بيانات database عن الأجهزة المتصلة بالنظام connected devices. وعند حدوث تغيير بطريقة

أو بأخرى لحالة state الأجهزة المتصلة؛ فإن hald يبيث broadcasts هذه المعلومات إلى أي عمليات سجلت registered هذا الحدث event. في الانظمة التي تستخدم HAL، فإن الأمر lshal يكون قادر على سرد قائمة بالأجهزة devices في ذلك النظام.

٧, ٢, ١٠ الخطوة ٧

نواة لينكس Linux kernel عادة تُحمل برنامج تشغيل driver أو وحدة نمطية module خاصة بالأجهزة لتتمكن تلك الاجهزة من القيام بوظائفها. استخدم الأمر lsmod لعرض وحدات يتم تحميلها حاليا:

### lsmod

الشاشة أدناه هي تنفيذ جزئي partial output. ونجد ان العמוד الأول هو اسم الوحدة النمطية module، أما العמוד الثاني فهو مقدار الذاكرة المستخدمة من الوحدة. أما الرقم في العמוד الثالث "Used by" فيشير إلى كم عدد الوحدات الأخرى التي تستخدم نفس الوحدة. ويمكن أيضا أن نجد أسماء تلك الوحدات في هذا العמוד، ولكن غالبا ما تكون تلك الأسماء غير كاملة:

```
sysadmin@localhost:~$ lsmod

Module                Size  Used by
dccp_diag             16384  0
dccp                  73728  1 dccp_diag
udp_diag             16384  0
unix_diag            16384  0
tcp_diag             16384  0
inet_diag            20480  3 dccp_diag,udp_diag,tcp_diag
xt_nat                16384  457
veth                  16384  0
vxlan                 49152  0
ip6_udp_tunnel       16384  1 vxlan
udp_tunnel           16384  1 vxlan
ipmi_devintf         20480  0
ib_iser              49152  0
rdma_cm              49152  1 ib_iser
iw_cm                 45056  1 rdma_cm
ib_cm                 45056  1 rdma_cm
ib_sa                 36864  2 rdma_cm,ib_cm
ib_mad                53248  2 ib_cm,ib_sa
ib_core               106496 6 ib_iser,rdma_cm,iw_cm,ib_cm,ib_sa,ib_mad
```

### تنفيذ الأمر lsmod

٨,٢,١٠ الخطوة ٨

لوحة النظام system board في العديد من أجهزة الحاسب تحتوي على ما يعرف باسم BIOS؛ وهو نظام الإدخال والإخراج الأساسي Basic Input and Output System. نظام إدارة BIOS، أو SMBIOS، هو معيار لتحديد standard defining هياكل البيانات data structures وكيفية نقل المعلومات

communicate information عن عتاد الحاسب computer hardware. الأمر dmidecode قادر على قراءة وطباعة المعلومات من SMBIOS في العديد من النظم. وهو لن يعمل هنا، لأنه لا يعمل بشكل صحيح ضمن البيئة الافتراضية virtual environment.

الأمر fdisk يمكن استخدامه بطريقتين: بشكل تفاعلي interactively وغير تفاعلي non-interactively. عند استخدام الخيار -l مع fdisk؛ سيصبح الأمر غير تفاعلي مع قائمة كتل الأجهزة، والتي تشمل الأقراص (الأقراص الصلبة) وحدات التخزين المنطقية.

بدون خيار -l يدخل الأمر fdisk إلى الوضع التفاعلي والذي يستخدم عادة لتعديل الأقسام partitions على القرص disk device.

٩، ٢، ١٠ الخطوة ٩

نفذ الأمر fdisk لسرد أجهزة الأقراص غير التفاعلية على شكل قطاعات sectors؛ وبدون عرض تحذيرات التوافق DOS:

```
fdisk -cul
```

التنفيذ الجزئي partial output لهذا الأمر يُظهر ثلاثة نتائج، أول قرصين هما /dev/sda و/dev/sdb. والأقسام الخاصة بهما. أما القرص الثالث والموجود بإسم /dev/sdc فهو وحدات التخزين المنطقية logical volumes التي تم حذفها omitte؛ وسترى ذلك في التنفيذ التالي:

```

sysadmin@localhost:~$ fdisk -cul

Disk /dev/sda: 21.5 GB, 21474836480 bytes
255 heads, 63 sectors/track, 2610 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x000571a2

   Device Boot      Start         End      Blocks   Id  System
/dev/sda1  *           1         2481     19921920   83  Linux
/dev/sda2                2481         2611      1046529    5  Extended
/dev/sda5                2481         2611      1046528   82  Linux swap /
Solaris

sysadmin@localhost:~$

```

### تنفيذ الأمر fdisk -cul

استخدام الخيار -u مع الأمر fdisk يعرض وحدات التخزين على شكل قطاعات بدلا من اسطوانات. أما الخيار

-c فيقوم بمنع الأمر fdisk من طباعة تحذيرات التوافق DOS.

## المعمل الحادي عشر – ادارة الحزم والعمليات

# Managing Packages and Processes

## ١١,١ مقدمة

هذا هو المعمل الحادي عشر: ادارة الحزم والعمليات. عن طريق أداء هذا التمرين؛ سوف تتعلم المواقع locations التي بها معلومات النواة kernel information، ومعرفة معلومات المعالجة process information والمكتبات libraries وملفات السجل log files، وحزم البرمجيات software packages.

في هذا التمرين، سوف نقوم بالمهام التالية:

- التحقق من كيفية استخدام نظام الملفات /proc من قبل النواة.
- استخدام الأمر ps لعرض معلومات عملية من العمليات.
- تعلم كيفية إدارة العمليات، والتي تشمل البدء، والإيقاف، والاستئناف.
- عرض ملفات السجل.
- إدارة امكانية تحميل مكتبات مشتركة.

## ١١,٢ النواة و /proc

في هذه المهمة سوف نستكشف الدليل /proc والأوامر التي تتصل communicate مع نواة لينكس Linux kernel. يظهر الدليل /proc على انه دليل عادي، مثل /usr أو /etc / ولكن الحقيقة غير ذلك تماماً. فالدليل /proc هو نظام الملفات الصوري أو الغير حقيقي pseudo filesystem ويتم الاحتفاظ به في ذاكرة الحاسب. وذلك عكس الدليلين /usr أو /etc (والتي عادة ما تكون محفوظة في محرك الأقراص).

يستطيع الدليل /proc الوصول الى معلومات كل عملية process قيد التشغيل running على النظام. برامج مثل ps و top تقوم بقراءة المعلومات حول العمليات processes قيد التشغيل running من خلال الدليل /proc وأدلتها الفرعية. والتي تحتوي على معلومات حول نظام التشغيل والأجهزة الخاصة به مثل /proc/cpuinfo (الخاص بمعلومات المعالج) و /proc/meminfo (الخاص بمعلومات الذاكرة) و /proc/devices (الخاص بالأجهزة).

بالإضافة الى الدلائل الثلاثة السابقة؛ يحتوي `/proc` على دليل فرعي باسم `/proc/sys` والذي هو جزء من الملفات الصورية `pseudo files` والتي يمكن استخدامها لتغيير `alter` إعدادات `settings` نواة التشغيل `running kernel`. وبما أن هذه الملفات ليست ملفات حقيقية `not real` فانه لا ينبغي أن يُستخدم محرر `editor` لتعديلها. بدلاً من ذلك يجب استخدام أحد الأمرين `echo` أو `sysctl` للكتابة فوق `overwrite` محتويات `contents` تلك الملفات. وللسبب نفسه، فلا تقم بمحاولة عرض `view` هذه الملفات في محرر `editor`، ولكن استخدام الأمر `cat` أو `sysctl` لذلك.

لعمل تغييرات دائمة `permanent changes` في التكوين `configuration`؛ فإن النواة `kernel` تستخدم الملف `/etc/sysctl.conf`؛ والذي يُستخدم عادة لإجراء تغييرات `changes` على ملفات `/proc` عندما يكون النظام في مرحلة التشغيل `starting up`.

١,٢,١ الخطوة ١

في هذه المهمة، سوف نستعرض بعض الملفات الموجودة في الدليل `/proc`:

```
ls /proc
```

وينبغي أن يكون التنفيذ مشابه لما يلي:

```

sysadmin@localhost:~$ ls /proc
1          cpuinfo      kallsyms     mpt          sysrq-trigger
19         crypto       kcore        mtrr         sysvipc
23         devices     key-users    net          thread-self
25         diskstats   keys         pagetypeinfo timer_list
41         dma         kmsg         partitions   timer_stats
50         driver      kpagecount   sched_debug  tty
62         execdomains kpageflags   schedstat    uptime
74         fb          loadavg      scsi         version
acpi       filesystems locks         self         version_signature
buddyinfo fs           mdstat       slabinfo     vmallocinfo
bus        interrupts  meminfo      softirqs     vmstat
cgroups   iomem       misc         stat         zoneinfo
cmdline   ioports     modules      swaps
consoles  irq         mounts       sys
sysadmin@localhost:~$

```

### تنفيذ الأمر ls /proc

تذكر أن الدلائل directories تحتوي على أرقام تمثل العمليات الجارية running processes على النظام on the system. العملية الأولى هي دائما /sbin/init وبالتالي فإن الدليل /proc/1 سيحتوي ملفات contains files بها معلومات حول عملية تشغيل .init.

ملف cmdline هو ملف داخل دليل المعالجة (على سبيل المثال /proc/1/cmdline) سوف يُظهر الأوامر التي تمت كتابتها وتنفيذها. وهنا يجب أن تلاحظ أن ترتيب العمليات التي تم تشغيلها يختلف اختلافا كبيرا من نظام إلى آخر. وحيث ان محتوى هذا الملف لا يحتوي على حرف سطر جديد، سيتم تنفيذ أمر echo ليذهب الموجه إلى سطر جديد.

٢,٢, ١١ الخطوة ٢

استخدم الأمران cat ثم ps لعرض معلومات حول عملية /sbin/init (معرف العملية PID هو الرقم ١):

```
cat /proc/1/cmdline; echo
```

وينبغي أن يكون التنفيذ مشابه لما يلي:

```
sysadmin@localhost:~$ cat /proc/1/cmdline; echo
/sbin/init
```

تنفيذ الأمر `cat /proc/1/cmdline; echo`

ملاحظة: في المثال السابق؛ تم تنفيذ الأمر `echo` على الفور بعد الأمر `cat`. ولعدم وجود معطيات `argument`؛ قتم وضع موجه الأوامر في سطر جديد.

ولمعرفة لماذا تم استخدام `echo`؛ نفذ الأمر `cat` فقط. وينبغي أن يكون التنفيذ مشابه لما يلي:

```
sysadmin@localhost:~$ cat /proc/1/cmdline
/sbin/init
```

لاحظ أن التنفيذ ظهر في نفس سطر موجه الأوامر؛ بينما استخدام الأمر `echo` سيجعله يظهر في سطر جديد.

والآن؛ نفذ الامر التالي لعرض العمليات قيد التشغيل؛ والخيار `-p` لتحديد رقم العملية التي نريدها:

```
ps -p 1
```

وينبغي أن يكون التنفيذ مشابه للتالي:

```
sysadmin@localhost:~$ ps -p 1
  PID TTY          TIME CMD
    1 ?            00:00:00 init
```

تنفيذ الأمر `ps -p 1`

الملفات الأخرى في الدليل `/proc` تحتوي على معلومات حول نظام التشغيل `operating system`. وفي المهام التالية سنتعلم عرض وتعديل هذه الملفات.

### ٣, ٢, ١١ الخطوة ٣

اعرض الملف `/proc/cmdline` لترى ما هي المعطيات `arguments` التي تم تمريرها `passed` للنواة `kernel` وقت الإقلاع `boot time`:

```
cat /proc/cmdline
```

وينبغي أن يكون التنفيذ مشابه لما يلي:

```
sysadmin@localhost:~$ cat /proc/cmdline
BOOT_IMAGE=/vmlinuz-4.2.0-34-generic root=/dev/mapper/vlabs--vg-root
ro cgroup_enable=
memory swapaccount=1
sysadmin@localhost:~$
```

تنفيذ الأمر `cat /proc/cmdline`

### ٣, ١١ إدارة العمليات

في هذه المهمة سوف نقوم ببدء العمليات وإيقافها.

### ١, ٣, ١١ الخطوة ١

في النافذة الطرفية `terminal`، اكتب الأمر التالي:

```
ping localhost > /dev/null
```

وينبغي أن يكون التنفيذ مشابه لما يلي:

```
sysadmin@localhost:~$ ping localhost > /dev/null
```

تنفيذ الأمر ping localhost > /dev/null

الأمر ping يقوم باختبار الاتصال، وفي المثال السابق تم إعادة توجيه ping إلى الملف /dev/null (والمعروف باسم سلة البت bit bucket). وعلى الرغم من انه كسلة للمحذوفات؛ إلا انه وبعد إرسال الأشياء التي لا نريدها إليه؛ فإنه يزودنا برقم الوظيفة ومعرف العملية.

لاحظ أن النافذة الطرفية terminal تبدو عالقة أو متوقفة عن الاستجابة hang up عند التنفيذ السابق. السبب في ذلك ان هذا الأمر يعمل في المقدمة foreground؛ وبالتالي سيستمر النظام بتنفيذ الأمر ping حتى يتم إنهاء العملية terminated أو تعطيلها suspended من قبل المستخدم user.

٢,٣,١١ الخطوة ٢

لإنهاء العملية Terminate process التي بالمقدمة foreground بشكل نهائي وليس للإيقاف، اضغط على مفتاح التحكم مع C:

Ctrl-c

وينبغي أن يكون التنفيذ مشابه لما يلي:

```
sysadmin@localhost:~$ ping localhost > /dev/null
^C
sysadmin@localhost:~$
```

تنفيذ الضغط على Ctrl-c

٣,٣,١١ الخطوة ٣

اعد كتابة نفس الامر السابق وقم بجعله يعمل في الخلف background عند طريق وضع علامة & كما يلي:

```
ping localhost > /dev/null &
```

وينبغي أن يكون التنفيذ مشابه لما يلي:

```
sysadmin@localhost:~$ ping localhost > /dev/null &
[1] 158
```

تنفيذ الأمر `ping localhost > /dev/null &`

عن طريق إضافة علامة `&` في نهاية الأمر السابق؛ فسيتم تنفيذ تلك العملية في الخلف `background`، وبالتالي يستمر المستخدم في التحكم والسيطرة على النافذة الطرفية `terminal`.

لاحظ أن الرقم ١٥٨ هو معرف العملية السابقة PID.

وهناك طريقة لإعادة كتابة الأمر السابق وهي الاستفادة من سجل الأوامر `history`. وذلك بالضغط على مفتاح سهم لأعلى `↑` على لوحة المفاتيح، ثم إضافة مسافة وعلامة `&` وأخيراً الضغط على مفتاح ادخال `<Enter>` (وهذه الطريقة موفرة للوقت عند ادخال أوامر مشابهة).

لاحظ أن الأمر السابق يقوم بإرجاع المعلومات التالية:

```
[1] 158
```

العدد بين الأقواس المربعة [ ] هو رقم الوظيفة `job` لهذه العملية `process`. أما الرقم الذي يليه فهو معرف العملية (PID) `Process ID`. إذاً رقم الوظيفة `job number` في المثال السابق هو ١، أما رقم معرف العملية PID هو ١٥٨.

(معلومة: كل نافذة طرفية/ قشرة `terminal/shell` سوف يكون لها رقم وظيفة فريد، كما ان لكل عملية PID رقم هوية ID number فريد على نطاق النظام بالكامل).

هذه المعلومات `information` مهمة `important` عند إجراء تعديلات `manipulations` على عملية من العمليات، مثل وقف `stop` العملية أو تغيير `changing` قيمة الأولوية لها `priority value`.

ملاحظة: من المرجح أن يكون معرف العملية في نظامك مختلف عن الأرقام الموجودة هنا.

٤, ٣, ١١ الخطوة ٤

الأمر التالي يستخدم لمعرفة الأوامر تعمل حالياً في النافذة الطرفية `current terminal`:

```
jobs
```

وينبغي أن يكون التنفيذ مشابه لما يلي:

```
sysadmin@localhost:~$ jobs
[1]+  Running                  ping localhost > /dev/null &
```

تنفيذ الأمر `jobs`

لاحظ أن الأمر `ping` ما زال قيد التشغيل، لذلك تم استخدامه في هذه الفقرة، عكس أمر مثل `echo` والذي يؤدي عمله ثم ينتهي وتصبح حالته `Done`، جرب استخدام الأمر `echo Hi > /dev/null% &`، ثم اكتب الأمر `jobs`.

٥, ٣, ١١ الخطوة ٥

شغل أمر `ping` آخر في الخلفية عن طريق كتابة ما يلي:

```
ping localhost > /dev/null &
```

وينبغي أن يكون التنفيذ مشابه لما يلي:

```
sysadmin@localhost:~$ ping localhost > /dev/null &
[2] 100
```

تنفيذ الأمر `ping localhost > /dev/null &`

لاحظ أن رقم الوظيفة `job number` ومعرف العملية `process ID` مختلف لهذا الأمر الجديد.

## ١١,٣,٦ الخطوة ٦

الآن، يجب أن يكون هناك اثنين من أوامر ping تعمل في الخلفية، للتحقق من ذلك، اكتب jobs مرة أخرى:

```
jobs
```

وينبغي أن يكون التنفيذ مشابه لما يلي:

```
sysadmin@localhost:~$ jobs
[1]-  Running                  ping localhost > /dev/null &
[2]+  Running                  ping localhost > /dev/null &
```

تنفيذ الأمر jobs

## ١١,٣,٧ الخطوة ٧

وبمجرد الانتهاء من التحقق من أن اثنين من أوامر ping قيد التشغيل، احضر bring الأمر الأول الى المقدمة foreground عن طريق كتابة ما يلي:

```
fg %1
```

وينبغي أن يكون التنفيذ مشابه لما يلي:

```
sysadmin@localhost:~$ fg %1
ping localhost > /dev/null
```

تنفيذ الأمر fg %1

## ١١,٣,٨ الخطوة ٨

لاحظ أن الأمر ping يأخذ السيطرة مرة أخرى على النافذة الطرفية terminal، لإيقاف العملية stopped (وليس إنهاؤها) واستعادة السيطرة على النافذة الطرفية، اكتب على Ctrl-z:

Ctrl-z

وينبغي أن يكون التنفيذ مشابه لما يلي:

```
sysadmin@localhost:~$ fg %1
ping localhost > /dev/null
^Z
[1]+  Stopped                  ping localhost > /dev/null
```

تنفيذ الأمر Ctrl-z

٩، ٣، ١١ الخطوة ٩

لإعادة استكمال continue executing العملية السابقة previous process، وجعلها تعمل في الخلفية background، نفذ الأمر التالي: execute الأمر التالي:

bg %1

وينبغي أن يكون التنفيذ مشابه لما يلي:

```
sysadmin@localhost:~$ bg %1
[1]+ ping localhost > /dev/null &
```

تنفيذ الأمر bg %1

١٠، ٣، ١١ الخطوة ١٠

أكتب الأمر jobs مرة أخرى للتحقق من العمليات قيد التشغيل running processes:

jobs

وينبغي أن يكون التنفيذ مشابه لما يلي:

```
sysadmin@localhost:~$ jobs
```

```
[1]-  Running                ping localhost > /dev/null &  
[2]+  Running                ping localhost > /dev/null &
```

تنفيذ الأمر jobs

١١,٣,١١ الخطوة ١١

بعد ذلك؛ أكتب أمر ping جديد عن طريق كتابة ما يلي:

```
ping localhost > /dev/null &
```

وينبغي أن يكون التنفيذ مشابه لما يلي:

```
sysadmin@localhost:~$ ping localhost > /dev/null &
```

```
[3] 101
```

تنفيذ الأمر ping localhost > /dev/null &

١٢,٣,١٢ الخطوة ١٢

أكتب الأمر jobs مرة أخرى للتحقق من ان هناك ثلاث three عمليات قيد التشغيل running processes:

```
jobs
```

وينبغي أن يكون التنفيذ مشابه لما يلي:

```
sysadmin@localhost:~$ jobs
```

```
[1]  Running                ping localhost > /dev/null &  
[2]-  Running                ping localhost > /dev/null &  
[3]+  Running                ping localhost > /dev/null &
```

تنفيذ الأمر jobs

### ١٣, ٣, ١١ الخطوة ١٣

استعمل رقم الوظيفة job number ، لإنهاء الأمر ping باستخدام الأمر kill ، ثم بعد ذلك تحقق من الإنهاء بتنفيذ الأمر jobs:

```
kill %3
```

```
jobs
```

وينبغي أن يكون التنفيذ مشابه لما يلي:

```
sysadmin@localhost:~$ kill %3
sysadmin@localhost:~$ jobs
[1]  Running          ping localhost > /dev/null &
[2]-  Running          ping localhost > /dev/null &
[3]+  Terminated      ping localhost > /dev/null
```

تنفيذ الأمرين kill %3 ، jobs

### ١٤, ٣, ١١ الخطوة ١٤

وأخيراً، يمكنك إنهاء كافة الأوامر التي من نوع واحد باستخدام الأمر killall. فمثلاً قم بإنهاء جميع أوامر ping ، بعد تنفيذ الامر السابق؛ انتظر بضع لحظات ثم قم بتشغيل الأمر jobs للتحقق من أن جميع العمليات قد أنهيت:

```
killall ping
```

```
jobs
```

وينبغي أن يكون التنفيذ مشابه لما يلي:

```
sysadmin@localhost:~$ killall ping
[1]- Terminated          ping localhost > /dev/null
[2]+ Terminated          ping localhost > /dev/null

sysadmin@localhost:~$ jobs
sysadmin@localhost:~$
```

تنفيذ الأمرين killall ping، jobs

## ١١,٤ استخدام أمر top لعرض العمليات

في هذه المهمة، سوف نستخدم الأمر top للعمل مع العمليات. افتراضياً؛ يقوم برنامج top بفرز sorts العمليات تنازلياً descending order وذلك بحسب نسبة الاستخدام بالمعالج CPU usage، وبالتالي فإن البرامج النشطة والمشغولة busiest programs ستكون في أعلى القائمة.

١١,٤,١ الخطوة ١

في النافذة الطرفية terminal، اكتب الأوامر التالية:

```
ping localhost > /dev/null &

ping localhost > /dev/null &
```

وينبغي أن يكون التنفيذ مشابه لما يلي:

```
sysadmin@localhost:~$ ping localhost > /dev/null &
[1] 112
sysadmin@localhost:~$ ping localhost > /dev/null &
[2] 113
```

تنفيذ الأمرين ping localhost > /dev/null &، ping localhost > /dev/null &

اعمل مذكرة لأرقام العمليات PIDS! وذلك لاستخدامها في الخطوات اللاحقة subsequent steps. حيث أن أرقام العمليات في نظامك ستكون مختلفة عن الأرقام الموجودة في المنهج.

١١,٤,٢ الخطوة ٢

اكتب الأمر top في النافذة الطرفية terminal:

top

وينبغي أن يكون التنفيذ مشابه لما يلي:

```
top - 00:56:34 up 8:18, 1 user, load average: 0.09, 0.14, 0.20
Tasks: 10 total, 1 running, 9 sleeping, 0 stopped, 0 zombie
Cpu(s): 2.1%us, 1.8%sy, 0.0%ni, 96.1%id, 0.0%wa, 0.0%hi, 0.0%si, 0.0%st
Mem: 65969788k total, 7629004k used, 58340784k free, 266340k buffers
Swap: 2097148k total, 0k used, 2097148k free, 939076k cached

  PID USER      PR  NI  VIRT  RES  SHR  S  %CPU  %MEM    TIME+  COMMAND
    1 root        20   0 17868 2872 2624 S   0   0.0   0:00.14 init
   19 syslog    20   0 171m 2796 2420 S   0   0.0   0:00.05 rsyslogd
   23 root        20   0 19120 2020 1824 S   0   0.0   0:00.00 cron
   25 root        20   0 50048 3416 2812 S   0   0.0   0:00.00 sshd
   41 bind       20   0 376m 19m 5988 S   0   0.0   0:00.04 named
   50 root        20   0 54460 2584 2176 S   0   0.0   0:00.00 login
   62 sysadmin  20   0 18084 3200 2700 S   0   0.0   0:00.01 bash
  112 sysadmin  20   0 6504 1824 1688 S   0   0.0   0:00.46 ping
  113 sysadmin  20   0 6504 1788 1656 S   0   0.0   0:00.40 ping
  114 sysadmin  20   0 17212 2240 2008 R   0   0.0   0:00.01 top
```

تنفيذ الأمر top

ملاحظة: إخراج الأمر top يتغير changes كل ثانيتين every 2 seconds.

٣, ٤, ١١ الخطوة

الأمر top هو برنامج تفاعلي interactive program، مما يعني أنه يمكنك إصدار الأوامر داخله. فمثلاً سيتم استخدام الأمر top لإنهاء عمليات pings عن طريق كتابة الحرف k:

k

لاحظ ان الموجه سيظهر أسفل كلمة Swap كما يلي:

```
top - 01:52:14 up 9:13, 1 user, load average: 0.04, 0.11, 0.19
Tasks: 10 total, 1 running, 9 sleeping, 0 stopped, 0 zombie
Cpu(s): 4.2%us, 2.6%sy, 0.0%ni, 93.0%id, 0.1%wa, 0.0%hi, 0.1%si, 0.0%st
Mem: 65969788k total, 8137188k used, 57832600k free, 286512k buffers
Swap: 2097148k total, 0k used, 2097148k free, 1110844 cached
PID to kill:
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
1	root	20	0	17868	2872	2624	S	0	0.0	0:00.14	init
19	syslog	20	0	171m	2796	2420	S	0	0.0	0:00.07	rsyslogd
23	root	20	0	19120	2020	1824	S	0	0.0	0:00.00	cron
25	root	20	0	50048	3416	2812	S	0	0.0	0:00.00	sshd
41	bind	20	0	376m	19m	5988	S	0	0.0	0:00.07	named
50	root	20	0	54460	2584	2176	S	0	0.0	0:00.00	login
62	sysadmin	20	0	18084	3200	2700	S	0	0.0	0:00.01	bash
103	sysadmin	20	0	6504	1824	1688	S	0	0.0	0:00.96	ping
104	sysadmin	20	0	6504	1788	1656	S	0	0.0	0:00.85	ping
108	sysadmin	20	0	17212	2240	2008	R	0	0.0	0:01.03	top

تنفيذ الحرف k

## ٤,٤, ١١ الخطوة ٤

في خانة الموجه والخاصة بالقضاء على العملية PID to kill، اكتب رقم العملية (PID) التي تريد انهاءها، وهي عملية اختبار الاتصال الأولى ping ذات الرقم ٩٩، ثم بعد ذلك اضغط على مفتاح Enter.

99

< Enter >

لاحظ التغييرات السريعة التي ستحدث في النافذة الطرفية terminal:

```
top - 07:25:53 up 101 days, 10:52, 1 user, load average: 1.22, 0.75, 0.69
Tasks: 11 total, 1 running, 10 sleeping, 0 stopped, 0 zombie
Cpu(s): 1.7%us, 1.8%sy, 0.0%ni, 96.1%id, 0.3%wa, 0.0%hi, 0.1%si, 0.0%st
Mem: 65940536k total, 42156404k used, 23784132k free, 774280k buffers
Swap: 999420k total,123248k used, 876172k free,2039748k cached
PID to kill: 99
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
1	root	20	0	17864	2888	2636	S	0	0.0	0:00.16	init
17	syslog	20	0	171m	2732	2352	S	0	0.0	0:00.06	rsyslogd
21	root	20	0	19116	2032	1840	S	0	0.0	0:00.00	cron
23	root	20	0	50048	3468	2860	S	0	0.0	0:00.00	sshd
39	bind	20	0	1128m	37m	6044	S	0	0.1	0:01.74	named
68	root	20	0	54456	2572	2156	S	0	0.0	0:00.00	login
80	sysadmin	20	0	18080	3268	2772	S	0	0.0	0:00.01	bash
99	sysadmin	20	0	6500	1856	1720	S	0	0.0	0:00.02	ping
101	sysadmin	20	0	6500	1780	1644	S	0	0.0	0:00.00	ping
102	sysadmin	20	0	6500	1788	1652	S	0	0.0	0:00.00	ping
103	sysadmin	20	0	17208	2300	2068	R	0	0.0	0:00.01	top

تنفيذ الأمر PID

## ٥, ٤, ١١ الخطوة ٥

في خانة الموجه والخاصة بالقضاء على العملية مع إشارة [15]: Kill PID with signal ، إذا كنت تريد استخدام رقم الإشارة الافتراضي وهو [15] فقم بالضغط على مفتاح Enter، أما إذا كنت تريد رقم إشارة آخر فقم بكتابته ليتم ارساله الى العملية التي تريد إنهاؤها. لاحظ أن الأمر ping الأول ستم إزالته ويبقى أمر ping واحد فقط في القائمة.

(قد تحتاج إلى الانتظار بضع ثوان حتى يقوم أمر top بالتحديث):

```
top - 07:28:01 up 101 days, 10:55, 1 user, load average: 0.87, 0.76
, 0.70
Tasks: 10 total, 1 running, 9 sleeping, 0 stopped, 0 zombie
Cpu(s): 0.5%us, 1.3%sy, 0.0%ni, 98.1%id, 0.1%wa, 0.0%hi, 0.0%si
, 0.0%st
Mem: 65940536k total, 42373968k used, 23566568k free, 774780k
buffers
Swap: 999420k total,123248k used, 876172k free, 2042156k cached
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
1	root	20	0	17864	2888	2636	S	0	0.0	0:00.16	init
17	syslog	20	0	171m	2732	2352	S	0	0.0	0:00.06	rsyslogd
21	root	20	0	19116	2032	1840	S	0	0.0	0:00.00	cron
23	root	20	0	50048	3468	2860	S	0	0.0	0:00.00	sshd
39	bind	20	0	1128m	37m	6044	S	0	0.1	0:01.74	named
68	root	20	0	54456	2572	2156	S	0	0.0	0:00.00	login
80	sysadmin	20	0	18080	3268	2772	S	0	0.0	0:00.01	bash
101	sysadmin	20	0	6500	1780	1644	S	0	0.0	0:00.04	ping
102	sysadmin	20	0	6500	1788	1652	S	0	0.0	0:00.04	ping
103	sysadmin	20	0	17208	2300	2068	R	0	0.0	0:00.01	top

تنفيذ الضغط على مفتاح ادخال

تمعن في التالي: هناك العديد من القيم الرقمية المختلفة التي يمكن إرسالها إلى عملية. وهذه هي القيم محددة مسبقا ولكل منها معنى مختلف. إذا كنت ترغب في معرفة المزيد عن هذه القيم، اكتب الأمر `man kill` في إطار النافذة الطرفية `.terminal`.

يشير الموجه إلى ان الاشارة الافتراضية للإنتهاء هي بواسطة `SIGTERM` أو العدد `١٥`.

٦,٤,١١ الخطوة ٦

انهي عملية `ping` المتبقية كما بالسابق، ولكن في خانة الموجه والخاصة بالقضاء على العملية `Kill PID with`

`[15] signal`، استخدم القيمة ٩ بدلا من العدد الافتراضي `١٥`. اضغط `Enter` للموافقة ثم دخول.

```
top - 07:29:59 up 101 days, 10:57, 1 user, load average: 1.84, 1.07, 0.82
Tasks: 9 total, 1 running, 8 sleeping, 0 stopped, 0 zombie
Cpu(s): 0.7%us, 0.7%sy, 0.0%ni, 98.5%id, 0.1%wa, 0.0%hi, 0.0%si, 0.0%st
Mem: 65940536k total, 42466120k used, 23474416k free, 775024k buffers
Swap: 999420k total, 123248k used, 876172k free,2043532k cached
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
102	sysadmin	20	0	6500	1788	1652	S	0	0.0	0:00.08	ping
1	root	20	0	17864	2888	2636	S	0	0.0	0:00.16	init
17	syslog	20	0	171m	2732	2352	S	0	0.0	0:00.06	rsyslogd
21	root	20	0	19116	2032	1840	S	0	0.0	0:00.00	cron
23	root	20	0	50048	3468	2860	S	0	0.0	0:00.00	sshd
39	bind	20	0	1128m	37m	6044	S	0	0.1	0:01.74	named
68	root	20	0	54456	2572	2156	S	0	0.0	0:00.00	login
80	sysadmin	20	0	18080	3268	2772	S	0	0.0	0:00.01	bash
103	sysadmin	20	0	17208	2300	2068	R	0	0.0	0:00.06	top

تنفيذ التغيير إلى العدد ٩ بدلا من ١٥

تمعن في التالي : إشارة الإنهاء ٩ أو SIGKILL هي إشارة قوية " forcefull " لا يمكن تجاهلها، على عكس القيمة الافتراضية ١٥ . لاحظ أنه تمت إزالة كل ما يشير إلى الأمر ping من قائمة الأمر .top

١١,٤,٧ الخطوة ٧

اكتب q للخروج من الأمر .top. وتظهر الشاشة التالية أن كلا أمرى ping قد أنهيا:

```
[1]- Terminated          ping localhost > /dev/null
[2]+ Killed              ping localhost > /dev/null
```

تنفيذ الأمر q

١١,٥ استخدام pkill وkill لإنهاء العمليات

في هذه المهمة، سوف نستمر بالعمل مع العمليات. وسوف نستخدم الأمرين pkill وkill لإنهاء terminate العمليات .processes

١١,٥,١ الخطوة ١

للبدء، اكتب الأوامر التالية في نافذة terminal:

```
sleep 888888 &
sleep 888888 &
```

وينبغي أن يكون التنفيذ مشابه لما يلي:

```
sysadmin@localhost:~$ sleep 888888 &
[1] 85
sysadmin@localhost:~$ sleep 888888 &
[2] 86
sysadmin@localhost:~$
```

تنفيذ الأمر `sleep 888888 &`، `sleep 888888 &`

عادة يستخدم الأمر `sleep` لعمل نقطة توقف `pause` لأحد برنامج القشرة (shell script) وذلك لفترة محددة من الزمن. في هذه الحالة يتم استخدامه مع الأمر الذي سوف يستغرق وقت طويل للتشغيل `.run`.

تأكد من ملاحظة PIDS على النظام الخاص بك مع الأمر `!sleep` سوف تكون أرقام العمليات PIDS الخاص بك مختلفة عن التي تظهر في المعمل.

٢,٥,١١ الخطوة ٢

حدد `determine` الوظائف `jobs` التي تعمل حالياً `currently running` بكتابة الأمر:

```
jobs
```

وينبغي أن يكون التنفيذ مشابه لما يلي:

```
sysadmin@localhost:~$ jobs
[1]-  Running                  sleep 888888 &
[2]+  Running                  sleep 888888 &
sysadmin@localhost:~$
```

تنفيذ الأمر `jobs`

### ٣,٥,١١ الخطوة ٣

الآن، استخدم الأمر `kill` لوقف التنفيذ الأول للأمر `sleep` عن طريق كتابة PID (رقم العملية للأمر `sleep` الأول والذي يظهر في جهازك). ثم بعد ذلك، نفذ الأمر `jobs` للتحقق من إيقاف العملية:

```
ps
kill PID
jobs
```

وينبغي أن يكون التنفيذ مشابه لما يلي:

```
sysadmin@localhost:~$ ps
TIME CMD
62 ? 00:00:00 bash
89 ? 00:00:00 sleep
90 ? 00:00:00 sleep
91 ? 00:00:00 ps
PID TTY
sysadmin@localhost:~$ kill 89
sysadmin@localhost:~$ jobs
[1]- Terminated sleep 888888
[2]+ Running sleep 888888 &
sysadmin@localhost:~$
```

تنفيذ الأمرين `kill PID`، `jobs`

تلميح مفيد: إذا لم تتمكن من تذكر PID من العملية الأولى فقط اكتب الأمر `ps` (عملية `process`)، كما هو موضح أعلاه.

### ٤,٥,١١ الخطوة ٤

بعد ذلك، استخدم الأمر `pkill` لإنهاء الأمر `sleep` المتبقي، وذلك باستخدام اسم البرنامج بدلا من PID:

pskill -15 sleep

وينبغي أن يكون التنفيذ مشابه لما يلي:

```
sysadmin@localhost:~$ pkill -15 sleep
[2]+  Terminated                  sleep 888888
sysadmin@localhost:~$
```

تنفيذ الأمر pkill -15 sleep

## ١١,٦ استخدام ps لتحديد وفرز العمليات

الأمر ps يمكن استخدامه لعرض view العمليات processes. افتراضيا فإن الأمر ps يقوم فقط بعرض العمليات قيد التشغيل running processes في القشرة الحالية current shell.

### ١١,٦,١ الخطوة ١

ابدأ عملية في الخلفية background باستخدام الرمز " & " للأمر ping. ثم قم بعرض العمليات الحالية current processes باستخدام الأمر ps:

```
ping localhost > /dev/null &
```

```
ps
```

وينبغي أن يكون التنفيذ مشابه لما يلي:

```
sysadmin@localhost:~$ ping localhost > /dev/null &
```

```
[1] 98
```

```
sysadmin@localhost:~$ ps
```

PID	TTY	TIME	CMD
62	?	00:00:00	bash
98	?	00:00:00	ping
99	?	00:00:00	ps

تنفيذ الأمرين `ping localhost > /dev/null &` و `ps`

اعمل ملاحظة بأرقام العمليات PID للأمر `ping`، والتي سوف تستخدمها في خطوات لاحقة.

٢،٦،١١ الخطوة

نفذ `execute` الأمر `ps` مع الخيار `-e` لعرض كافة العمليات وليس فقط العمليات التي قمت بإنشائها:

```
ps -e
```

وينبغي أن يكون التنفيذ مشابه لما يلي:

```
sysadmin@localhost:~$ ps -e
```

PID	TTY	TIME	CMD
1	?	00:00:00	init
19	?	00:00:00	rsyslogd
23	?	00:00:00	cron
25	?	00:00:00	sshd
41	?	00:00:00	named
50	?	00:00:00	login
62	?	00:00:00	bash
98	?	00:00:00	ping
100	?	00:00:00	ps

تنفيذ الأمر `ps -e`

ونظرا لأن هذه البيئة افتراضية virtualized operating system؛ فالعمليات التي تظهر هي أقل بكثير من الموجودة عادة عند تشغيل نظام لينكس مباشرة directly على الأجهزة hardware.

٣, ٦, ١١ الخطوة ٣

استخدم الأمر ps مع الخيار -o لتحديد الأعمدة التي تريد إخراجها.

```
ps -o pid, tty, time, %cpu, cmd
```

وينبغي أن يكون التنفيذ مشابه لما يلي:

```
sysadmin@localhost:~$ ps -o pid, tty, time, %cpu, cmd
PID TT          TIME %CPU CMD
 62 ?           00:00:00  0.0 -bash
 98 ?           00:00:00  0.0 ping localhost
102 ?           00:00:00  0.0 ps -o pid, tty, time, %cpu, cmd
```

تنفيذ الأمر ps -o pid, tty, time, %cpu, cmd

٤, ٦, ١١ الخطوة ٤

قم باستخدام الخيار --sort لتحديد أي عامود تريد فرزها. افتراضيا، سيكون الفرز تصاعديا ascending order للعامود المحدد، كما يمكن اختياره بوضع الرمز زائد "+" أمام اسم العامود. والعكس صحيح؛ فاستخدام الرمز ناقص "-" أمام اسم العامود كفيل بجعل عملية الفرز تنازلية.

فرز تنفيذ الأمر ps بواسطة العامود %mem:

```
ps -o pid, tty, time, %mem, cmd --sort %mem
```

وينبغي أن يكون التنفيذ مشابه لما يلي:

```
sysadmin@localhost:~$ ps -o pid, tty, time, %mem, cmd --sort %mem
PID TT          TIME %MEM CMD
103 ?           00:00:00  0.0 ps -o pid, tty, time, %mem, cmd --sort %mem
 98 ?           00:00:00  0.0 ping localhost
 62 ?           00:00:00  0.0 -bash
```

تنفيذ الأمر `ps -o pid, tty, time, %mem, cmd --sort %mem`

١١,٦,٥ الخطوة ٥

في حين أن الأمر `ps` يمكن أن يظهر النسبة المئوية من الذاكرة التي تستخدمها عملية ما، فإن الأمر `free` يُظهر الاستخدام الكلي لذاكرة النظام:

```
free
```

وينبغي أن يكون التنفيذ مشابه لما يلي:

```
sysadmin@localhost:~$ free
              total        used        free      shared    buffers
cached
Mem:           65969788    6242744    59727044         0       300852    106
2284
-/+ buffers/cache:    4879608    61090180
Swap:           2097148         0       2097148
```

تنفيذ الأمر `free`

١١,٦,٦ الخطوة ٦

أنهي الأمر `ping` بواسطة الأمر `kill`، ثم تحقق من التنفيذ مع الأمر `jobs`:

```
kill PID
```

```
jobs
```

وينبغي أن يكون التنفيذ مشابه لما يلي:

```
sysadmin@localhost:~$ kill 98
sysadmin@localhost:~$ jobs
[1]+  Terminated                  ping localhost > /dev/null
```

تنفيذ الأمرين kill PID ، jobs

## ١١,٧ عرض سجلات النظام

سجلات النظام System logs مهمة لكثير من المهام tasks بما في ذلك تحديد مشاكل نظام التشغيل fixing operating system والتأكد ensure من أن النظام الخاص بك آمن secure. معرفة مكان تخزين stored ملفات سجل النظام system log files وكيفية المحافظة maintain عليها هو أمر مهم لمسؤول النظام .system administrator.

هناك برنامجين خفيه daemons للتعامل مع سجل الرسائل log messages: الأول syslogd والثاني klogd. عادة لا تحتاج للقلق حول البرنامج klogd حيث أنه يعالج handles فقط رسائل سجل النواة kernel log messages ويرسل sends معلومات السجل الموجودة به الى البرنامج syslogd.

الرسائل Messages التي تم إنشاؤها generated من قبل النواة kernel في وقت التمهيد boot time يتم تخزينها stored في الملف /var/log/dmesg. يسمح الأمر dmesg بعرض view رسائل النواة الحالية current kernel messages ، فضلا عن توفير السيطرة control إذا ما تم عرض تلك الرسائل في النافذة الطرفية terminal console window.

ملف السجل الرئيسي main log file والمكتوب بواسطة البرنامج الخفي والذي يتعامل مع سجل الرسائل syslogd هو /var/log/messages.

بالإضافة إلى أن التسجيل logging يتم من خلال البرنامج السابق syslogd؛ فهناك العديد من العمليات processes الأخرى التي لديها التسجيل logging الخاص بها. من الأمثلة على العمليات التي تقوم بتسجيل

خاص بها خادم الويب أباتشي Apache web server (ملف السجل log file موجود في دليل  
Common Unix Printing System (/var/log/httpd)، مثال آخر هو نظام يونيكس للطباعة  
(/var/log/cups) والبرنامج الخفي auditd daemon (/var/log/audit).

ملاحظة: في أنظمة CentOS systems، يسمى البرنامج syslogd باسم rsyslogd.

١١,٧,١ الخطوة ١

الأوامر القليلة المقبلة سوف تتطلب صلاحيات rights مستخدم رئيسي superuser، استخدم الأمر su لتبديل  
المستخدم switch user إلى حساب المستخدم الجذر root account:

```
su - root
```

```
{Enter the password: netlab123}
```

وينبغي أن يكون التنفيذ مشابه لما يلي:

```
sysadmin@localhost:~$ su - root
Password:
root@localhost:~#
```

تنفيذ الأمر su -root

١١,٧,٢ الخطوة ٢

يتم تخزين stored سجلات النظام system logs في الدليل /var/log. قم بسرد قائمة بتلك الملفات:

```
ls /var/log
```

وينبغي أن يكون التنفيذ مشابه لما يلي:

```
root@localhost:~# ls /var/log
alternatives.log  boot          cron.log      faillog       lastlog       news
wtmp
apt               bootstrap.log dmesg         fsck          mail.err      syslog
auth.log         bttmp        dpkg.log      kern.log      mail.log
upstart
root@localhost:~#
```

تنفيذ الأمر `ls /var/log`

٣،٧،١١ الخطوة ٣

كل ملف سجل `log file` يمثل خدمة أو ميزة `service or feature`. على سبيل المثال؛ يعرض الملف `auth.log` معلومات بشأن الصلاحيات `authorization` أو المصادقات `authentication`، مثل محاولات تسجيل دخول المستخدم `user login attempts`. البيانات الجديدة يتم تخزينها `stored` في الجزء السفلي `bottom` من الملف. نفذ الأوامر التالية لمشاهدة مثال على ذلك:

```
ssh localhost

{At the first prompt, type yes}

{At the second prompt, type abc}

{At the third prompt, type abc}

{At the fourth prompt, type abc}

tail -5 /var/log/auth.log
```

وينبغي أن يكون التنفيذ مشابه لما يلي:

```

root@localhost:~# ssh localhost
The authenticity of host 'localhost (:::1)' can't be established.
ECDSA key fingerprint is
5f:e2:43:0f:f9:26:e5:d5:77:ba:9e:95:72:9e:ee:64.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'localhost' (ECDSA) to the list of known
hosts.
root@localhost's password:
Permission denied, please try again.
root@localhost's password:
Permission denied, please try again.
root@localhost's password:
Permission denied (publickey,password).
root@localhost:~# tail -5 /var/log/auth.log
Apr  8 20:25:13 localhost sshd[117]: pam_unix(sshd:auth):
authentication failure; logname= uid=0 euid=0 tty=ssh ruser=
rhost=localhost      user=root
Apr  8 20:25:16 localhost sshd[117]: Failed password for root from :::1
port 58940 ssh2
Apr  8 20:25:28  sshd[117]: last message repeated 2 times
Apr  8 20:25:28 localhost sshd[117]: Connection closed by :::1[preauth]
Apr  8 20:25:28 localhost sshd[117]: PAM 2 more authentication
failures; logname= uid=0 euid=0 tty=ssh ruser= rhost=localhost
user=root
root@localhost:~#

```

#### تنفيذ ملف السجل

تم استخدام الأمر ssh لتوليد البيانات في الملف `/var/log/auth.log`. يتم تسجيل حتى محاولات الدخول الفاشلة في الملف `/var/log/auth.log`.

٤,٧,١١ الخطوة ٤

لرؤية مثال آخر على إدخالات السجل log entries، نفذ الأوامر التالية:

## crontab -e

وينبغي أن يكون التنفيذ كالتالي:

```
# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h dom mon dow   command
~
```

تنفيذ الأمر crontab -e

قم بإضافة السطر التالي إلى الوثيقة (تذكر حرف i سوف يسمح لك بالدخول إلى وضع إدراج) ثم حفظ والخروج مع ESC، ثم wq: وأخيرا مفتاح أدخل Enter:

```
i  
  
0 2 * * 0 who >> /tmp/whothere  
  
< ESC >  
  
:wq  
  
< Enter >
```

بعد الانتهاء من التعديلات، اعرض ملف السجل لخدمة crontab عن طريق الأمرين التاليين:

```
crontab -l | tail -2  
  
tail /var/log/cron.log
```

وينبغي أن يكون التنفيذ مشابه لما يلي:

```

root@localhost:~# crontab -l | tail -2
# m h dom mon dow    command
 0 2 * * 0    who >> /tmp/whothere

root@localhost:~# tail /var/log/cron.log
Apr  8 17:33:32 localhost /usr/sbin/cron[22]: (CRON) INFO (pidfile
fd = 3)
Apr  8 17:33:32 localhost /usr/sbin/cron[23]: (CRON) STARTUP (fork ok)
Apr  8 17:33:32 localhost /usr/sbin/cron[23]: (CRON) INFO (Running
@reboot jobs)
Apr  8 18:17:01 localhost /USR/SBIN/CRON[79]: (root) CMD ( cd / &&
run-parts -
-report /etc/cron.hourly)
Apr  8 19:17:02 localhost /USR/SBIN/CRON[82]: (root) CMD ( cd / &&
run-parts -
-report /etc/cron.hourly)
Apr  8 20:17:01 localhost /USR/SBIN/CRON[99]: (root) CMD ( cd / &&
run-parts -
-report /etc/cron.hourly)
Apr  8 20:31:18 localhost crontab[121]: (root) BEGIN EDIT (root)
Apr  8 20:32:42 localhost crontab[121]: (root) REPLACE (root)
Apr  8 20:32:42 localhost crontab[121]: (root) END EDIT (root)
Apr  8 20:33:19 localhost crontab[132]: (root) LIST (root)

root@localhost:~#

```

تنفيذ مثال على مدخلات السجل

٥,٧,١١ الخطوة ٥

اعرض الأسطر الخمسة الأخيرة من الملف `/var/log/dmesg` لرؤية رسائل النواة `kernel messages` من وقت التمهيد `boot time`، ثم نفذ الأمر `dmesg` مع الأمر `tail` لعرض رسائل النواة الخمسة الماضية:

```
tail -5 /var/log/dmesg
```

```
dmesg | tail -5
```

وينبغي أن يكون التنفيذ مشابه لما يلي:

```
root@localhost:~# tail -5 /var/log/dmesg

[  2.922003] type=1400 audit(1386098331.347:10): apparmor="STATUS"
operation="profile_load" name="/usr/sbin/tcpdump" pid=848
comm="apparmor_parser"

[  2.989112] Bridge firewalling registered

[  3.007035] ip_tables: (C) 2000-2006 Netfilter Core Team

[  3.010733] nf_conntrack version 0.5.0 (16384 buckets, 65536 max)

[  3.020096] input: ImPS/2 Generic Wheel Mouse a
s
          /devices/platform/i8042/serio1/input/input2

root@localhost:~# dmesg | tail -5

[279447.718341] device veth0pl17180 left promiscuous mode
[279447.718408] br998ad950-b830: port 1(veth0pl17180) entered
disabled state

[279448.519497] bre2e72298-4b5e: port 1(veth0pl17664) entered
disabled state

[279448.525087] device veth0pl17664 left promiscuous mode
[279448.525091] bre2e72298-4b5e: port 1(veth0pl17664) entered
disabled state

root@localhost:~#
```

تنفيذ الأمرين `tail -5 /var/log/dmesg`، `tail -5 dmesg |`

من المحتمل ان تتساءل الآن ماذا تعني كل هذه الرسائل؟ الجواب على هذا السؤال ليس بسيط، ولكن هذا الدرس لا يهدف إلى شرح `explain` معنى جميع رسائل السجل `log messages`، وإنما لمعرفة أين توجد `where` `to find` تلك الرسائل.

وكما اكتسبت المزيد من الخبرة في التعامل مع نظام التشغيل لينكس `Linux`، فسوف تبدأ في استكشاف المشاكل وحلها `troubleshoot problems`. وفي معظم المشاكل التي قد تواجهها، فإن المكان الأول الذي تريد أن تتحقق منه هو ملفات السجل `log files`.

من أجل توفير مثال واقعي لاستكشاف الأخطاء وإصلاحها، قم بتنفيذ المجموعة التالية من المهام.

## ٦,٧,١١ الخطوة ٦

بما أنك في حساب المستخدم الجذر root user؛ أدخل الأمر التالي لتعطيل disable قدرة ability المستخدم sysadmin user من إنشاء إدخلات في crontab، ثم قم بالخروج من حساب المستخدم الجذر والانتقال إلى حساب المستخدم sysadmin:

```
echo "sysadmin" > /etc/cron.deny
```

```
exit
```

وينبغي أن يكون التنفيذ مشابه لما يلي:

```
root@localhost:~# echo "sysadmin" > /etc/cron.deny
root@localhost:~# exit
logout
sysadmin@localhost:~$
```

تنفيذ الأمرين echo "sysadmin" > /etc/cron.deny، exit

## ٧,٧,١١ الخطوة ٧

حاول تنفيذ الأمر crontab التالي:

```
crontab -e
```

وينبغي أن يكون التنفيذ مشابه لما يلي:

```
sysadmin@localhost:~$ crontab -e
You (sysadmin) are not allowed to use this program (crontab)
See crontab(1) for more information
sysadmin@localhost:~$
```

محاولة تنفيذ الأمر crontab -e

ملاحظة: يفشل هذا الأمر بسبب وجود اسم المستخدم في الملف /etc/cron.deny. في حالة وجود اسم المستخدم في الملف السابق؛ فإنه لا يمكنه استخدام الأمر crontab.

٨,٧,١١ الخطوة ٨

بدلاً من تبديل المستخدم switching user إلى مستخدم جذر root مع الأمر su؛ استخدم الأمر sudo لتنفيذ الأوامر بامتيازات الجذر root privileges كما في المثال التالي:

```
sudo tail -5 /var/log/cron.log
```

```
{Enter the password: netlab123}
```

وينبغي أن يكون التنفيذ مشابه لما يلي:

```
sysadmin@localhost:~$ sudo tail -5 /var/log/cron.log
[sudo] password for sysadmin:
Apr  8 20:31:18 localhost crontab[121]: (root) BEGIN EDIT (root)
Apr  8 20:32:42 localhost crontab[121]: (root) REPLACE (root)
Apr  8 20:32:42 localhost crontab[121]: (root) END EDIT (root)
Apr  8 20:33:19 localhost crontab[132]: (root) LIST (root)
Apr  8 20:47:06 localhost crontab[139]: (sysadmin) AUTH (crontab
command not allowed)
sysadmin@localhost:~$
```

تنفيذ الأمر sudo tail -5 /var/log/cron.log

وكما هو ملاحظ في السطر الأخير من تنفيذ الأمر tail، فإنه لا يُسمح للمستخدم sysadmin من استخدام الأمر

crontab.

## ١١,٨ المكتبات المشتركة

المكتبات المشتركة Shared libraries هي ملفات تحتوي على اكواد (تعليمات برمجية)، حيث يمكن لملفات البرنامج program files القابل للتنفيذ executable program الارتباط link بتلك المكتبات من أجل استخدام الاكواد الموجودة بها بدلاً من كتابتها مرة أخرى. وبسبب أن العديد من البرامج multiple programs غالباً ما ترتبط بملف مكتبة واحد single library؛ فإن ذلك يساعد على تقليل reduce مقدار المساحة space المطلوبة لتنفيذ هذا الكود. وبعبارة أخرى البرنامج يستطيع استدعاء أحد المكتبات متى ما أراد ذلك وبالتالي لا يحتاج الى وجود نسخة خاصة باكواد تلك المكتبة بداخله.

وغالبا ما تخزن stored ملفات المكتبات library files في الدليلين /lib و /usr/lib. ويمكن إدراج دلائل إضافية additional directories عن طريق تحرير ملف التكوين configuration file، etc/ld.so.conf/ أو يمكنك أيضا إنشاء create ملفات files تنتهي بالامتداد التالي conf. ووضعها في الدليل /etc/ld.so.conf.d/. وأخيرا، يمكنك أيضا تعيين set متغير البيئة environment variable التالي LD\_LIBRARY\_PATH كذلك.

### ١١,٨,١ الخطوة ١

عند التنفيذ execute بحساب المستخدم الجذر root؛ فإن الأمر ldconfig يمكن استخدامه لتحديث update ذاكرة التخزين cache والروابط الرمزية symbolic links للمكتبات المشتركة shared libraries في النظام. كمستخدم عادي an ordinary user سوف تقوم بتنفيذ الأمر ldconfig لطباعة print قائمة المكتبات المشتركة list of shared libraries كالتالي:

```
ldconfig -p | less
```

وينبغي أن يكون التنفيذ الجزئي كالتالي:

```

148 libs found in cache `/etc/ld.so.cache'

    libz.so.1 (libc6,x86-64) => /lib/x86_64-linux-
gnu/libz.so.1
    libxml2.so.2 (libc6,x86-64) => /usr/lib/x86_64-linux-
gnu/libxml2.so.2
    libxcb.so.1 (libc6,x86-64) => /usr/lib/x86_64-linux-
gnu/libxcb.so.1
    libwrap.so.0 (libc6,x86-64) => /lib/x86_64-linux-
gnu/libwrap.so.0
    libwind.so.0 (libc6,x86-64) => /usr/lib/x86_64-linux-
gnu/libwind.so.0
    libuuid.so.1 (libc6,x86-64) => /lib/x86_64-linux-
gnu/libuuid.so.1
    libutil.so.1 (libc6,x86-
64, OS ABI: Linux 2.6.24) => /lib/x86_64-linux-g
nu/libutil.so.1

    libusb-1.0.so.0 (libc6,x86-64) => /lib/x86_64-linux-gnu/libusb-
1.0.so.0
    libusb-1.0.so.0 (libc6,x86-64) => /usr/lib/x86_64-linux-
gnu/libusb-1.0.s
o.0

    libusb-0.1.so.4 (libc6,x86-64) => /lib/x86_64-linux-gnu/libusb-
0.1.so.4
    libunistring.so.0 (libc6,x86-64) => /usr/lib/x86_64-linux-
gnu/libunistri
ng.so.0

    libudev.so.0 (libc6,x86-64) => /lib/x86_64-linux-
gnu/libudev.so.0
    libtinfo.so.5 (libc6,x86-64) => /lib/x86_64-linux-
gnu/libtinfo.so.5
    libtic.so.5 (libc6,x86-64) => /usr/lib/x86_64-linux-
gnu/libtic.so.5
    libthread_db.so.1 (libc6,x86-
64, OS ABI: Linux 2.6.24) => /lib/x86_64-li
nux-
gnu/libthread_db.so.1

    libtasn1.so.3 (libc6,x86-64) => /usr/lib/x86_64-linux-
gnu/libtasn1.so.3
    libstdc++.so.6 (libc6,x86-64) => /usr/lib/x86_64-linux-
gnu/libstdc++.so.
6
:

```

تنفيذ جزئي للأمر ldconfig -p | less

تذكر الضغط على الحرف q لأتهاء صفحات الامر اقل .less pager.

٢, ٨, ١١ الخطوة ٢

من أجل عرض view المكتبات libraries والمرتبطة بملفات القابلة للتنفيذ linked to an executable مثل /bin/bash، نفذ الأمر :ldd

```
ldd /bin/bash
```

وينبغي أن يكون التنفيذ مشابه لما يلي:

```
sysadmin@localhost:~$ ldd /bin/bash
linux-vdso.so.1 => (0x00007ffce6fbd000)
libtinfo.so.5 => /lib/x86_64-linux-gnu/libtinfo.so.5 (0x00007fb32ae94000)
libdl.so.2 => /lib/x86_64-linux-gnu/libdl.so.2 (0x00007fb32ac90000)
libc.so.6 => /lib/x86_64-linux-gnu/libc.so.6 (0x00007fb32a8d1000)
/lib64/ld-linux-x86-64.so.2 (0x0000563ddfa4e000)
sysadmin@localhost:~$
```

تنفيذ الأمر ldd /bin/bash

## المعمل الثاني عشر – تكوين الشبكة

# Network Configuration

## ١٢,١ مقدمة

هذا هو المعمل الثاني عشر: تكوين الشبكة. عن طريق أداء هذا التمرين، سوف تتعلم حول تكوين جهاز الحاسب الخاص بك على الشبكة.

في هذا التمرين، سوف نقوم بالمهام التالية:

- دراسة معلومات حول تكوين الشبكة.

## ١٢,٢ استكشاف الشبكة

في هذه المهمة، سوف تقوم بتنفيذ بعض الأوامر ودراسة بعض الملفات بهدف عرض تكوين configuration شبكة الاتصال.

### ١٢,٢,١ الخطوة ١

من أجل تحديد عنوان بروتوكول الإنترنت الخاص بجهازك (IP)، نفذ الأمر التالي:

```
ifconfig
```

وينبغي أن يكون التنفيذ مشابه لما يلي:

```

sysadmin@localhost:~$ ifconfig
eth0      Link
          encap:Ethernet  HWaddr 3a:3b:65:10:f6:43
          inet addr:192.168.1.2  Bcast:0.0.0.0  Mask:255.255.255.0
          inet6 addr: fe80::383b:65ff:fe10:f643/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:15 errors:0 dropped:0 overruns:0 frame:0
          TX packets:9 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:1206 (1.2 KB)  TX bytes:690 (690.0 B)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

sysadmin@localhost:~$

```

### تنفيذ الأمر ifconfig

يظهر في التنفيذ مجموعتين two blocks من المعلومات information؛ المجموعة الأولى تبدأ من ETH0 وتعطي معلومات عن أول بطاقة شبكة من نوع إيثرنت Ethernet network card. أما المجموعة الثانية فتبدأ من lo وتعكس معلومات حول الاسترجاع loopback أو واجهة الشبكة الداخلية internal network interface.

السطر الثاني في كل مجموعة يحتوي على معلومات ذات صلة pertinent بالإصدار الرابع version 4 من بروتوكول الانترنت (وتسمى عناوين IPv4) في حين أن السطر الثالث لديه معلومات عن الإصدار السادس من بروتوكول الانترنت (الإصدار IPv6). إن عناوين IPv4 هي وسيلة قديمة لتحديد الأجهزة identifying

machines عن طريق سلسلة من الأرقام series of numbers، وهو لا يزال يستخدم على نطاق واسع اليوم على الرغم من أن الإصدار السادس المحسن improved IPv6 متاح منذ سنوات.

يتم عرض عناوين IPv4 على شكل أربعة أرقام عشرية four decimal numbers تتراوح من ٠ إلى ٢٥٥ مفصولة بنقاط periods.

اما عناوين IPv6 فهي مكونة من ١٢٨ رقم يتم عرضها كخانات ست عشرية hexadecimal digits تتراوح من ٠ إلى f. الأرقام الست عشرية عموما مقسمة إلى مجموعات من أربعة أرقام groups of four digits مفصولة بنقاط colons. إذا كان العدد مكون من أرقام متتالية consecutive ست عشرية بقيمة صفر؛ فسيتم استبدالها بإثنين من النقاط two colons.

٢,٢,٢ الخطوة ٢

وجود عنوان IP يسمح للنظام الخاص بك your system من التواصل communicate مع الانظمة الأخرى other systems على نفس الشبكة same network. اما للتواصل مع أنظمة أخرى على شبكات مختلفة other networks فيم ذلك عن طريق أجهزة التوجيه routing devices. لعرض جدول معلومات التوجيه table of routing information؛ استخدم الأمر route:

route

route -n

وينبغي أن يكون التنفيذ مشابه لما يلي:

```

sysadmin@localhost:~$ route
Kernel IP routing table
Destination Gateway Genmask Flags Metric Ref U
se Iface
192.168.1.0 * 255.255.255.0 U 0 0
0 eth0

sysadmin@localhost:~$ route -n
Kernel IP routing table
Destination Gateway Genmask Flags Metric Ref U
se Iface
192.168.1.0 0.0.0.0 255.255.255.0 U 0 0
0 eth0

sysadmin@localhost:~$

```

#### تنفيذ الأمرين route -n ، route

لاحظ أن تنفيذ الأمر route الأول هو أرقام مع وجود بعض الرموز مثل الرمز " \* ". أما في التنفيذ الثاني للأمر route فنجد أرقام بدلا من الرموز السابقة وذلك نتيجة لاستخدام الخيار -n مع الأمر route.

عند الاتصال connecting to بأجهزة الحاسب الأخرى فيتم استخدام إما عنوان IP أو اسم المضيف hostname. يمكن استخدام أسماء المضيفين Hostnames إذا تم إدخالها في ملف /etc/hosts جنبا إلى جنب مع عناوين IP المرتبطة بها، أو إذا كان خادم اسم المجال (DNS) - Domain Name Server يعطي عنوان IP إلى برنامج ترجمة أسماء المضيفين host name translation.

زوجان من الأسماء couple of names التي تكون عادة موجودة في الملف /etc/hosts هما localhost و localhost.localdomain، وكلاهما يستخدم للإشارة refer إلى الجهاز الحالي current machine.

٣، ٢، ١٢ الخطوة ٣

تحقق من أن عنوان IP التالي 127.0.0.1 له إدخال entry في الملف /etc/hosts:

```
grep 127.0.0.1 /etc/hosts
```

يجب أن يظهر التنفيذ على النحو التالي، تحديد أسماء localhost:

```
sysadmin@localhost:~$ grep 127.0.0.1 /etc/hosts
127.0.0.1 localhost
sysadmin@localhost:~$
```

تنفيذ الأمر `grep 127.0.0.1 /etc/hosts`

يمكن استخدام الأمر `ping command` لمعرفة ما إذا كان النظام متصلاً `connected` في الوقت الراهن `presently` بالشبكة أو لا.

في بعض الأحيان، يتم تكوين `configured` النظام لعدم الاستجابة لطلبات الأمر `ping`. ولذلك فإن عدم وجود استجابة `lack of a response` للأمر `ping` لا يعني ان النظام غير متصل بالشبكة.

٤, ٢, ١٢ الخطوة ٤

حاول معرفة ما إذا كان الجهاز `localhost` سيستجيب لطلبات `requests` اختبار الاتصال `ping` الأربعة عن طريق الأمر التالي:

```
ping -c4 localhost
```

ويكون التنفيذ مشابه لما يلي:

```

sysadmin@localhost:~$ ping -c4 localhost
PING localhost (127.0.0.1) 56(84) bytes of data.
64 bytes from localhost (127.0.0.1): icmp_req=1 ttl=64 time=0.045 ms
64 bytes from localhost (127.0.0.1): icmp_req=2 ttl=64 time=0.028 ms
64 bytes from localhost (127.0.0.1): icmp_req=3 ttl=64 time=0.032 ms
64 bytes from localhost (127.0.0.1): icmp_req=4 ttl=64 time=0.040 ms

--- localhost ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 2998ms
rtt min/avg/max/mdev = 0.028/0.036/0.045/0.007 ms
sysadmin@localhost:~$

```

### تنفيذ الأمر ping -c4 localhost

على عكس الأمر ping المتوفر في نظام التشغيل مايكروسوفت ويندوز Microsoft Windows operating system، فإن الحالة الافتراضية للأمر ping في لينكس Linux هي عدم التوقف عن إرسال طلبات الاختبار. فإذا كنت قد نسيت تحديد الخيار -c؛ فيجب عليك وقف الأمر يدويا manually عن طريق الضغط على مفتاح التحكم والمفتاح c (CTRL + c).

ويمكن أيضاً الاستفادة من أسماء المضيفين Hostnames إذا كانت مسجلة registered في خادم اسم المجال (DNS). فإذا كان النظام الخاص بك متصلاً connected مع خوادم DNS؛ فعندها سيقوم مُدخل اسم الخادم nameserver في الملف /etc/resolv.conf بتكوين configures النظام الخاص بك لاستخدام هذه الخوادم لحل resolve أسماء المضيفين hostnames إلى عناوين IP.

١٢,٢,٥ الخطوة ٥

اعرض الملف /etc/resolv.conf لمعرفة هل توجد إدخلات entries exists لاسم الخادم  
:nameserver

```
cat /etc/resolv.conf
```

يجب أن يظهر في التنفيذ إدخال entry لاسم خادم nameserver واحد:

```
sysadmin@localhost:~$ cat /etc/resolv.conf
nameserver 127.0.0.1
sysadmin@localhost:~$
```

تنفيذ الأمر `cat /etc/resolv.conf`

٦,٢,١٢ الخطوة ٦

استخدم الأمر `dig` والخاص بالبحث لإعادة تحويل `resolve` الاسم `localhost.localdomain` إلى عنوان

:IP

```
dig localhost.localdomain
```

ويكون التنفيذ مشابه لما يلي:

```

sysadmin@localhost:~$ dig localhost.localdomain

; <<>> DiG 9.8.1-P1 <<>> localhost.localdomain
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 35943
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL:
0

;; QUESTION SECTION:
;localhost.localdomain.      IN      A

;; ANSWER SECTION:
localhost.localdomain.  86400  IN      A      127.0.0.1

;; AUTHORITY SECTION:
localhost.              86400  IN      NS     localhost.localdomain.

;; Query time: 803 msec
;; SERVER: 127.0.0.1#53(127.0.0.1)
;; WHEN: Mon Apr 11 13:33:03 2016
;; MSG SIZE rcvd: 69

sysadmin@localhost:~$

```

تنفيذ الأمر `dig localhost.localdomain`

لاحظ في التنفيذ السابق أن أسماء الخوادم nameserver الأولى التي تم عرضها في الملف `/etc/resolv.conf` هي التي استجابت `responded` كما هو واضح في قسم الاجوبة `ANSWER SECTION` في الشكل السابق.

## ٧,٢,١٢ الخطوة ٧

يمكنك استخدام الأمر `dig` والخاص بالبحث لإعادة تحويل `resolve` أسماء المجالات اللانقة الكاملة `fully qualified domain names` الأخرى. استخدم الأمر `dig` لإعادة تحويل اسم المضيف `cserver.example.com` إلى عنوان IP:

```
dig cserver.example.com
```

ويكون التنفيذ مشابهه لما يلي:

```
sysadmin@localhost:~$ dig cserver.example.com
; <<>> DiG 9.8.1-P1 <<>> cserver.example.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 57203
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL:
1
;; QUESTION SECTION:
;cserver.example.com.          IN      A
;; ANSWER SECTION:
cserver.example.com.          86400  IN      A      192.168.1.2
;; AUTHORITY SECTION:
example.com.                  86400  IN      NS     example.com.
;; ADDITIONAL SECTION:
example.com.                  86400  IN      A      192.168.1.2
;; Query time: 990 msec
;; SERVER: 127.0.0.1#53(127.0.0.1)
;; WHEN: Mon Apr 11 13:45:38 2016
;; MSG SIZE rcvd: 83
sysadmin@localhost:~$
```

تنفيذ الأمر `dig cserver.example.com`

اسم المجال اللائق بالكامل (FQDN) لا يشمل فقط اسم المضيف hostname؛ وإنما أيضاً المجال domain الذي يكون المضيف hostname فيه. اسم المجال اللائق (FQDN) للاسم cserver.example.com؛ فإن cserver هو اسم المضيف hostname و example.com هو المجال.

٨, ٢, ١٢ الخطوة ٨

أمر dig خاص بالبحث لإعادة تحويل الاسم إلى عنوان IP، ولكن الخيار -x يعكس العملية وتحويل عنوان IP إلى اسم للعنوان 192.168.1.2 كما يظهر في التنفيذ التالي:

```
dig -x 192.168.1.2
```

ويكون التنفيذ مشابه لما يلي:

```

sysadmin@localhost:~$ dig -x 192.168.1.2
; <<>> DiG 9.8.1-P1 <<>> -x 192.168.1.2
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 26791
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 1, ADDITIONAL:
1
;; QUESTION SECTION:
;2.1.168.192.in-addr.arpa.      IN      PTR
;; ANSWER SECTION:
2.1.168.192.in-addr.arpa. 86400 IN      PTR      cserver.example.com.
2.1.168.192.in-addr.arpa. 86400 IN      PTR      example.com.
;; AUTHORITY SECTION:
1.168.192.in-addr.arpa. 86400 IN      NS      example.com.
;; ADDITIONAL SECTION:
example.com.      86400 IN      A      192.168.1.2
;; Query time: 0 msec
;; SERVER: 127.0.0.1#53(127.0.0.1)
;; WHEN: Mon Apr 11 13:57:50 2016
;; MSG SIZE rcvd: 119
sysadmin@localhost:~$

```

تنفيذ الأمر dig -x 192.168.1.2

٩، ٢، ١٢ الخطوة ٩

ينفذ الأمر netstat مجموعة كبيرة ومتنوعة من المهام المتعلقة بالشبكات. للحصول على فكرة عن بعض من قدراته، نفذ الأمر مع خيار "--help":

```
netstat --help
```

وفيما يلي تنفيذ جزئي للأمر السابق:

```

sysadmin@localhost:~$ netstat --help

usage: netstat [-vWeenNcCF] [<Af>] -r

netstat {-V|--version|-h|--help}
netstat [-vWnNcaeol] [<Socket> ...]
netstat { [-vWeenNac] -i | [-cWnNe] -M | -s }

-r, --route                display routing table
-i, --interfaces           display interface table
-g, --groups               display multicast group memberships
-s, --statistics           display networking statistics (lik SNMP)
-M, --masquerade           display masqueraded connections
-v, --verbose              be verbose
-W, --wide                 don't truncate IP addresses
-n, --numeric              don't resolve names
--numeric-hosts            don't resolve host names
--numeric-ports            don't resolve port names
--numeric-users            don't resolve user names
-N, --symbolic             resolve hardware names
-e, --extend               display other/more information
-p, --programs             display PID/Program name for sockets

```

### تنفيذ جزئي للأمر netstat --help

أحد الاستخدامات الشائعة للأمر netstat هو تحديد الخدمات services التي تقوم بالاستماع listening إلى اتصال أو انتظار اتصال قادم incoming connection. على سبيل المثال، يتم استدعاء خدمة Secure Shell or SSH والتي تسمح للمستخدمين users بإجراء عمليات تسجيل الدخول logins عن بعد remote أو من شبكة network. عادة تقوم خدمة SSH بالاستماع listen إلى المنفذ رقم 22 في طبقة .TCP

الجدول التالي يظهر استخدامات الامر netstat مع عدد من الخيارات والتي ظهرت في التنفيذ السابق للأمر

:netstat --help

الأمر	المعنى
netstat -t	لعرض قنوات الاتصال sockets لطبقة tcp.
netstat -u	لعرض قنوات الاتصال sockets لطبقة udp.
netstat -w	لعرض قنوات الاتصال sockets لطبقة raw.
netstat -x	لعرض قنوات الاتصال sockets لطبقة unix.
netstat -a	لعرض جميع قنوات الاتصال sockets سواء في حالة استماع أو لا.
netstat -l	للاستماع الى احد المنافذ.
netstat -n	لعرض النتائج على شكل ارقام.
netstat --numeric-hosts	عرض عناوين المضيف على شكل ارقام، دون التأثير على اسم المنفذ أو المستخدم.
netstat --numeric-ports	عرض المنافذ على شكل ارقام، دون التأثير على اسم المضيف أو أسماء المستخدمين.
netstat --numeric-users	عرض معرفات المستخدم على شكل ارقام، دون التأثير على اسم المضيف أو أسماء المنفذ.
netstat -N	عرض أسماء العتاد.
netstat -e	عرض معلومات إضافية.
netstat -o	عرض المعلومات information والمتعلقة بالمؤقت الشبكي networking timers.
netstat -p	عرض (PID) process id وأيضا اسم البرنامج الذي ينتمي إليه كل مأخذ توصيل socket.
netstat -v	اعرض الحدث.

طباعة المعلومات المحددة كل ثانية بشكل متواصل.	netstat -c
لعرض معلومات جدول التوجيه.	netstat -r
لعرض الواجهات (interfaces)	netstat -i
عرض معلومات عضوية مجموعة البث المتعدد IPv4 multicast لـ و IPv6	netstat -g
عرض قائمة بالاتصالات masqueraded. (برنامج اتصال خاص بلينكس مشابه لنظام 1:many)	netstat -M
عرض تلخيص بإحصائيات كل بروتوكول.	netstat -s
عدم اقتطاع عناوين IP باستخدام تقنية إخراج واسع حسب الحاجة. (output as wide as needed)	netstat -W
طباعة جدول معلومات التوجيه وذلك من (Forwarding Information Base : FIB)	netstat -F
طباعة معلومات التوجيه من ذاكرة الطرق الموجودة في ذاكرة cache بدلاً من FIB.	netstat -C
الاصدار	netstat -V

جدول يوضح خيارات الأمر netstat

المنافذ المعروفة Well-known ports هي الموجودة في نطاق 0-1023، وتستخدم عادة من قبل عمليات النظام system processes وذلك بغرض لتوفير provide خدمات الشبكة network services. ويمكن الاطلاع على قائمة الخدمات service names وأسماء وأرقام المنافذ المرتبطة associated port numbers في الملف /etc/services.

استخدم الامر التالي لعرض قائمة بأسماء وارقام المنافذ ports:

```
cat /etc/services
```

التنفيذ التالي هو تنفيذ جزئي لأسماء وأرقام المنافذ ports:

```

sysadmin@localhost:~$ cat /etc/services
# Network services, Internet style
#
# Note that it is presently the policy of IANA to assign a single
well-known
# port number for both TCP and UDP; hence, officially ports have two
entries
# even if the protocol doesn't support UDP operations.
#
# Updated from http://www.iana.org/assignments/port-
numbers and other
# sources like http://www.freebsd.org/cgi/cvsweb.cgi/src/etc/services
.
# New ports will be added on request if they have been officially
assigned
# by IANA and used in the real-
world or are needed by a debian package.
# If you need a huge list of used numbers please install the nmap
package.
tcpmux      1/tcp          # TCP port service multiplexer
echo        7/tcp
echo        7/udp
discard     9/tcp      sink null
discard     9/udp      sink null
sysstat     11/tcp     users
daytime     13/tcp
daytime     13/udp
netstat     15/tcp
gotd        17/tcp     quote
msp         18/tcp     # message send protocol
msp         18/udp
chargen     19/tcp     ttytst source
chargen     19/udp     ttytst source
ftp-data    20/tcp
ftp         21/tcp
fsp         21/udp     fspd
ssh         22/tcp     # SSH Remote Login Protocol
ssh         22/udp
telnet      23/tcp
smtp        25/tcp     mail
time        37/tcp     timserver
time        37/udp     timserver

```

تنفيذ جزئي للأمر cat /etc/services

## ١٠, ٢, ١٢ الخطوة ١٠

استخدم الأمر netstat لمعرفة ما إذا كان منفذ TCP لخدمة ssh (منفذ 22) لديه عملية استماع:

```
netstat -tl
```

```
netstat -tln
```

ويكون التنفيذ مشابه لما يلي:

```
sysadmin@localhost:~$ netstat -tl
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 example.com:domain     *:*                     LISTEN
tcp        0      0 localhost:domain       *:*                     LISTEN
tcp        0      0 *:ssh                  *:*                     LISTEN
tcp        0      0 localhost:953          *:*                     LISTEN
tcp6       0      0 [::]:domain           [::]:*                  LISTEN
tcp6       0      0 [::]:ssh               [::]:*                  LISTEN
tcp6       0      0 localhost:953          [::]:*                  LISTEN
sysadmin@localhost:~$
```

تنفيذ الأمرين netstat -tln ، netstat -tl

## ١١, ٢, ١٢ الخطوة ١١

الخيار -t يضاف إلى الأمر netstat لعرض منافذ TCP فقط. اما الخيار -l يحد التنفيذ إلى المنافذ ذات خدمة الاستماع. واخيراً -n فهو يظهر عناوين شبكة عددياً:

```
sysadmin@localhost:~$ netstat -ltn
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 192.168.1.2:53          0.0.0.0:*               LISTEN
tcp        0      0 127.0.0.1:53           0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:22             0.0.0.0:*               LISTEN
tcp        0      0 127.0.0.1:953          0.0.0.0:*               LISTEN
tcp6       0      0 :::53                  :::*                    LISTEN
tcp6       0      0 :::22                  :::*                    LISTEN
tcp6       0      0 :::1:953               :::*                    LISTEN
sysadmin@localhost:~$
```

تنفيذ الأمر netstat -ltn

## المعمل الثالث عشر – النظام وحماية المستخدم

# System and User Security

## ١٣,١ مقدمة

هذا المعمل الثالث عشر: النظام وحماية المستخدم. يهتم هذا المعمل بمعرفة المستخدمين المتواجدين على النظام الخاص بك، فعن طريق أداء هذا التمرين سوف تكون قادر على رصد اي محاولة لتسجيل الدخول إلى النظام، وأيضاً عرض أي عملية من عمليات التشغيل الخاصة بأولئك المستخدمين.

في هذا التمرين، سوف نقوم بالمهام التالية:

- معرفة الفرق بين حساب المستخدم المسؤول وحساب أو حسابات المستخدمين العادية.
- عرض معلومات حساب المستخدم.

## ١٣,٢ حسابات المستخدم والمجموعة

في هذه المهمة، سوف نتعلم عن حسابات المستخدمين user accounts، والملفات files والأوامر commands التي عن طريقها يتم عرض display معلومات حساب المستخدم user account .information

### ١٣,٢,١ الخطوة ١

يتم تعريف حسابات المستخدم user والنظام system في الملفين /etc/passwd و /etc/shadow.

قم بعرض الاسطر العشرة الأولى من الملف /etc/passwd:

```
head /etc/passwd
```

ويكون التنفيذ مشابهه لما يلي:

```
sysadmin@localhost:~$ head /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/bin/sh
bin:x:2:2:bin:/bin:/bin/sh
sys:x:3:3:sys:/dev:/bin/sh
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/bin/sh
man:x:6:12:man:/var/cache/man:/bin/sh
lp:x:7:7:lp:/var/spool/lpd:/bin/sh
mail:x:8:8:mail:/var/mail:/bin/sh
news:x:9:9:news:/var/spool/news:/bin/sh
sysadmin@localhost:~$
```

تنفيذ الأمر `head /etc/passwd`

لاحظ أن هذا الملف يحتوي على نقاط colon لتدل على قاعدة بيانات database كافة حسابات المستخدمين والنظام، والتي هي متوفرة available على نظام التشغيل الحالي.

هناك ثلاثة أنواع من الحسابات سيتم التطرق إليها في هذه المذكرة وهي:

**الأول - حساب المستخدم User accounts:** حيث يتم إسناد حسابات المستخدمين user accounts وذلك للمستخدمين العاديين للسماح لهم للوصول access إلى نظام التشغيل operating system. ويعتبر حساب sysadmin الذي يُستخدم لتسجيل الدخول إلى النظام هو خير مثال على حساب المستخدم العادي.

**الثاني - الحساب الرئيسي أو الجذر The root account:** وهو حساب مستخدم خاص special user account، افتراضيا لديه وصول access وتحكم control كامل على النظام. ويشار إليه أحيانا باسم حساب المستخدم الرئيسي superuser account.

**الثالث - حسابات النظام System accounts:** وتستخدم حسابات النظام من قبل نظام التشغيل operating system أو الخدمات services التي تدير العمليات running processes عليه. هذه الخدمات لا يمكن تشغيلها حتى بواسطة المستخدم الجذر؛ وذلك بهدف أن يتم الحفاظ على النظام أكثر أمنا more secure من خلال الحد limiting من الضرر damage الذي يمكن ان تسببه حسابات خدمة التضمين comprised

service account. ولنفس السياق السابق فإن حسابات النظام System accounts لا تستخدم أبدا بشكل مباشر directly من قبل المستخدمين العاديين regular users.

٢,٢,١٣ الخطوة ٢

استخدم الأمر grep لعرض تسجيل record لحساب المستخدم sysadmin:

```
grep sysadmin /etc/passwd
```

ويكون التنفيذ مشابه لما يلي:

```
sysadmin@localhost:~$ grep sysadmin /etc/passwd
sysadmin:x:1001:1001:System Administrator,,,:/home/sysadmin:/bin/bash
sysadmin@localhost:~$
```

تنفيذ الأمر grep

بسبب استخدام الأمر grep فإن التنفيذ output يتضمن فقط معلومات حساب account information لاسم مستخدم واحد one username.

طريقة أخرى لاسترداد retrieve معلومات حساب مستخدم account information هي عن طريق تشغيل الأمر التالي getent passwd username. يتفوق الأمر getent على الأمر grep لأنه يستطيع الوصول إلى حسابات المستخدمين التي لم يتم تعريفها محليا locally. وبعبارة أخرى، الأمر getent قادر على الحصول على معلومات المستخدم user information وذلك للمستخدمين users المحتمل أنهم المعرفين defined في خدمة دليل الشبكة network directory servers مثل LDAP أو NIS أو مجال ويندوز Windows Domain أو خدمة مجال الدليل الفعال Active Directory Domain servers.

### ٣,٢,٣ الخطوة ٣

استخدم الأمر `getent` لاسترجاع معلومات حول الحساب `sysadmin`:

```
getent passwd sysadmin
```

وينبغي أن يكون التنفيذ كما يلي:

```
sysadmin@localhost:~$ getent passwd sysadmin
sysadmin:x:1001:1001:System Administrator,,,:/home/sysadmin:/bin/
bash
sysadmin@localhost:~$
```

تنفيذ الأمر `getent passwd sysadmin`

ملاحظة: في النتيجة السابقة لم يتم العثور على أي حسابات للشبكة `network accounts`؛ وبالتالي فإن التنفيذ `output` المعروض هو تماما مثل النظر في الملف `/etc/passwd`.

### ٤,٢,٤ الخطوة ٤

يمكنك الاطلاع على الوثائق `documentation` للحقول `fields` في الملف `/etc/passwd` مع الأمر التالي:

```
man 5 passwd
```

ويكون التنفيذ مشابه لما يلي:

```
PASSWD (5) File Formats and Conversions  
PASSWD (5)
```

**NAME**

```
passwd - the password file
```

**DESCRIPTION**

```
/etc/passwd contains one line for each user account, with  
seven fields
```

```
delimited by colons (":"). These fields are:
```

- o login name
- o optional encrypted password
- o numerical user ID
- o numerical group ID
- o user name or comment field
- o user home directory
- o optional user command interpreter

```
Manual page passwd(5) line 1 (press h for help or q to quit)
```

تنفيذ الأمر `man 5 passwd`

تذكر أثناء عرض دليل المستخدم `man page`، اضغط على مفتاح الإدخال `Enter` للانتقال الى السطر الذي يليه، أما للانتقال صفحة بعد أخرى فقم بالضغط على مفتاح مسافة `Spacepage`. وأخيراً قم بالضغط على حرف `q` للأنهاء.

الفاصلة تحدد ملف `/etc/passwd` بالمجالات التالية:

```
account:password:UID:GID:GECOS:directory:shell
```

وفيما يلي بتحليل لهذه الحقول الملف `/etc/passwd`:

١. **account**: حقل الحساب هو اسم المستخدم `username`.
٢. **password**: هذا الحقل أساساً يستعمل لتخزين كلمة المرور، والآن يستعمل أيضاً تخزين كلمات المرور ولكن في ملف `etc/shadow`. ويستعمل حرف `x` للدلالة على أن الخانة محجوزة لكلمة مرور وذلك بغرض إخفائها.
٣. **UID**: هو رقم هوية المستخدم.
٤. **GID**: هو رقم هوية المجموعة.
٥. **GECOS**: يستخدم لتخزين الاسم الكامل `full name` للمستخدم `user`.
٦. **directory**: هو الدليل الرئيسي `home directory` للمستخدم، حيث يتم وضعه عند التسجيل الأول `first logging` في النظام.
٧. **shell field**: حقل القشرة يحدد `defines shell` القشرة للمستخدم، فمثلاً `/bin/bash` تكون للمستخدم العادي `normal user`، أما `/sbin/nologin` فتكون للحساب المستخدم من قبل النظام `system` أو خدمة `service`.

١٣,٢,٥ الخطوة ٥

اعرض معلومات حسابك أو حساب مستخدم محدد، وذلك باستخدام الأمر `id`:

```
id  
  
id root
```

ويكون التنفيذ مشابه لما يلي:

```

sysadmin@localhost:~$ id
uid=1001(sysadmin) gid=1001(sysadmin) groups=1001(sysadmin),4(adm),27(sudo)

sysadmin@localhost:~$ id root
uid=0(root) gid=0(root) groups=0(root)

sysadmin@localhost:~$

```

### تنفيذ الأمرين id، id root

التنفيذ السابق يُظهر هوية المستخدم الخاص بك برقم وأيضاً باسم كما يلي: uid=1001(sysadmin). ويعرض أيضاً هوية المجموعة الرئيسية: gid=1001(sysadmin)، وجميع المجموعات التي ينتمي إليها المستخدم وهي groups=1001(sysadmin) و 4(adm) و 27(sudo). إذن فإن حساب المستخدم الخاص بك ينتمي إلى ثلاث مجموعات groups فقط.

ملف /etc/group مع الملف /etc/passwd يحددان determines عضوية مجموعتك group memberships. افتراضياً يتم تحديد مجموعتك الأساسية your primary group عن طريق مطابقة matching رقم مجموعتك your GID والموجود في /etc/passwd مع رقم المجموعة GID الموجود في الملف /etc/group. أيضاً يتم تحديد أي عضوية لمجموعة ثانوية secondary group memberships في الملف /etc/group.

شكل المدخلات في الملف /etc/group هي:

```
group_name:password:GID:user_list
```

٦، ٢، ١٣ الخطوة ٦

الأمر getent يمكن من خلاله استرجاع retrieve معلومات information حول مُدخلات entries قاعدة بيانات مثل الملف /etc/group. استخدم الأمر getent لاسترجاع المعلومات حول مجموعتي sysadmin وADM:

```
getent group sysadmin
```

```
getent group adm
```

ويكون التنفيذ مشابهه لما يلي:

```
sysadmin@localhost:~$ getent group sysadmin
sysadmin:x:1001:
sysadmin@localhost:~$ getent group adm
adm:x:4:sysadmin
sysadmin@localhost:~$
```

تنفيذ الأمرين 'getent group sysadmin'، 'getent group adm'

المجموعات Groups تستخدم من أجل التحكم في الوصول controlling access إلى الملفات files.

افتراضياً أي ملف جديد تقوم بإنشائه create يكون مملوك owned by من قبل مجموعتك الأساسية your primary group. لإسناد ملكية أي ملفات جديدة قمت بإنشائها لأحد المجموعات الثانوية الخاصة بك your secondary groups قم باستخدام الأمر newgrp. الشكل الأساسي basic format لهذا الأمر هو newgrp group\_name. يجب أن تكون عضواً في مجموعة من أجل استخدام الأمر newgrp. في حين أن الأمر id سيعرض اسم المجموعة GROUP\_NAME كمجموعتك الأساسية your primary group.

لتغيير مالك مجموعة group owner وذلك لمجموعة من الملفات الموجودة existing files التي يملكها المستخدم الخاص بك your user owns، يمكنك ذلك باستخدام الأمر chgrp. الشكل الأساسي basic format لهذا الأمر هو chgrp group\_name file\_name.

يجب أن تكون عضواً في مجموعة group member لتغيير ملكية الملف لمجموعة file group ownership لتلك المجموعة بالإضافة إلى ملكيتك للملف own the file.

## ٧,٢,١٣ الخطوة ٧

الأمران التاليان يوضحان الهوية الخاصة بك وهما `whoami` و `groups`. الأمر `groups` يمكن أيضا أن يُستخدم للحصول على قائمة المجموعات `list of groups` لمستخدم آخر وليس بالضرورة أن يكون المستخدم الحالي للنظام. استخدم الأمر `whoami` والأمر `groups` لعرض معلومات بحسابك (`sysadmin`)، ثم بعد ذلك استخدم الأمر `groups` فقط لعرض مجموعات المستخدم الرئيسي (`root`) وذلك من دون عملية تبديل للمستخدم الحالي:

```
whoami

groups

groups root
```

ويكون التنفيذ مشابه لما يلي:

```
sysadmin@localhost:~$ whoami
sysadmin
sysadmin@localhost:~$ groups
sysadmin adm sudo
sysadmin@localhost:~$ groups root
root : root
sysadmin@localhost:~$
```

تنفيذ الأوامر `whoami`، `groups`، `groups root`

## ١٣,٣ من في هذا النظام

في هذه المهمة، سوف نقوم بتنفيذ `execute` بعض الأوامر لمعرفة من المستخدم `user` الذي قام بعملية تسجيل الدخول إلى هذا النظام `logged into the system`.

### ١٣,٣,١ الخطوة ١

استخدم الأمر `who` للحصول على القائمة الحالية `current list` من المستخدمين `users` المتواجدين على النظام `on the system`:

```
who
```

ويكون التنفيذ مشابه لما يلي:

```
sysadmin@localhost:~$ who
sysadmin tty          Apr 11 14:32
sysadmin@localhost:~$
```

تنفيذ الأمر `who`

تنفيذ الأمر `who` هو أربعة أعمدة:

- اسم المستخدم **Username**: يظهر في العمود الأول اسم المستخدم، وهو `sysadmin` في المثال أعلاه.
- النافذة الطرفية **Terminal**: يظهر في العمود الثاني معرف لنوع النافذة الطرفية `Terminal`، وهي `tty` في المثال السابق. حيث أن لكل محطة `Terminal` اسم منفصل `separate name` يُستخدم من قبل المستخدم الرئيسي `superuser` للسيطرة على العمليات `control processes`.
- التاريخ/ الوقت **Date/Time**: العمود الثالث هو لإظهار تاريخ ووقت تسجيل الدخول، وهو ١١ أبريل ١٤:٣٢ كما يظهر في السطر الأول من التنفيذ أعلاه.

- استضافة Host: على الرغم من عدم وجود ناتج output بالعامود الرابع fourth column في الوقت الحالي، إلا انه يمكن أن يكون في ذلك العامود اسم أو عنوان IP لمضيف host محلي local أو بعيد remote. الأشكال forms التالية تشير إلى دخول محلي local logins (:#) أو (#:#). أما خلاف ذلك، فقد يظهر مضيف بعيد remote host بالاسم إذا كان يدعم ذلك resolvable أو من خلال عنوان IP.

٢, ٣, ١٣ الخطوة ٢

استخدم الأمر w للحصول على عرض أكثر تفصيلا more detailed للمستخدمين المتواجدين حاليا على النظام on your system:

w

تنفيذ الأمر w يكون مشابهه لما يلي:

```
sysadmin@localhost:~$ w
 15:17:08 up 6 days, 15 min,  1 user,  load average: 0.39, 0.34, 0.37
USER      TTY      FROM            LOGIN@   IDLE   JCPU   PCPU WHAT
sysadmin  console                14:32    4.00s  0.16s  0.00s w
sysadmin@localhost:~$
```

تنفيذ الأمر w

التنفيذ السابق هو ملخص summary لمدة عمل النظام وعدد المستخدمين الذين قاموا بتسجيل الدخول logged in ومعدلات تحميل النظام system load averages للفترات ١ و ٥ و ١٥ دقيقة الماضية.

ونرى في المثال أعلاه إدخال لكل مستخدم مع اسم تسجيل الدخول login name، واسم النافذة tty name والمضيف host ووقت تسجيل الدخول login time ووقت الخمول idle time و JCPU (وقت وحدة المعالجة المركزية المستخدمة من قبل الوظائف الخلفية) و PCPU (وقت وحدة المعالجة المركزية المستخدمة من قبل العملية الحالية) وما الأوامر المنفذة على سطر الأوامر الحالي.

## ١٣,٤ المستخدم الرئيسي أو الجذر

في هذه المهمة سوف تتعلم كيفية الوصول إلى حساب المستخدم الجذر بعدد من طرق مختلفة.

حساب المستخدم الجذر root هو حساب مميز special في نظام التشغيل لينكس؛ حيث أن له صلاحيات غير محددة للسيطرة control والوصول access على كامل النظام. الجدير بالذكر أن قيمة رقم هوية المستخدم UID والتي تكون بقيمة الصفر؛ هي التي تمنح هذه السلطة authority وليس اسم المستخدم.

بشكل عام، لا يعتبر تسجيل الدخول login إلى النظام بصفة مستخدم جذر root user تدريب جيد. وبشكل خاص ينطبق هذا بالكامل على البيئة الرسومية graphical environment.

تستطيع الوصول access الى حساب الجذر root user باستخدام أحد الأمرين su أو sudo.

الأمر su عادة ما يستخدم لتبديل المستخدمين switch users وذلك لبدء قشرة جديدة new shell كمستخدم آخر another user. بشكل افتراضي يكون التحويل إلى المستخدم الجذر root user. بالإضافة إلى تبديل المستخدمين؛ يستخدم ايضا الأمر su عندما نحتاج إلى تنفيذ execute سلسلة من الأوامر series of commands باسم المستخدم الجذر.

في المقابل يستخدم الأمر sudo لتنفيذ execute أمر واحد single command باسم المستخدم الجذر، وذلك بكتابة الأمر مع sudo. كما يجب أن يتم تكوين الأمر sudo من قبل المستخدم الجذر قبل ان يتمكن المستخدم العادي ordinary user من استخدامه.

### ١,٤,١٣ الخطوة ١

لإعداد set up الأمر sudo للمستخدمين العاديين regular users؛ استخدم الأمر visudo. والذي يتطلب requires وصول الجذر root access، حيث سيفشل الأمر السابق عند تنفيذه بواسطة المستخدمين العاديين. حاول attempt تكوين الوصول configure access إلى الأمر sudo:

visudo

ويكون التنفيذ مشابهه لما يلي:

```
sysadmin@localhost:~$ visudo
visudo: /etc/sudoers: Permission denied
visudo: /etc/sudoers: Permission denied
sysadmin@localhost:~$
```

تنفيذ الأمر visudo

يظهر في التنفيذ output أن الأمر visudo فشل لأنه حاول تعديل الملف /etc/sudoers، وهو ملف لا يستطيع المستخدمين العاديين regular users تعديله بسبب الأذونات permissions. حيث يتحكم controls هذا الملف بالوصول access إلى الأمر sudo وينبغي ألا يتم تعديله مباشرة ولكن باستخدام الأمر visudo.

٢,٤,١٣ الخطوة ٢

بدل المستخدمين switch users إلى حساب المستخدم الجذر root user، واعطي كلمة سر الجذر netlab123 عند الطلب:

su -

{Enter the password: netlab123}

ويكون التنفيذ مشابهه لما يلي:

```
sysadmin@localhost:~$ su -
Password:
root@localhost:~#
```

تنفيذ الأمر sudo

رمز الشرطة أو الواصلة " - " الذي يُكتب بعد الأمر su هو اختصار abbreviation إلى الخيار -l ، وهو الأمر الذي يجعل من تسجيل الدخول login والذي يتم بواسطة الأمر su هو تسجيل دخول كامل complete login، وذلك بتنفيذ execute برامج تسجيل دخول login scripts ينتمي belong إلى المستخدم الجذر .root user

بدون رمز الشرطة dash " - " أو الخيار -l فإن الأمر su سيبدل switch هوية المستخدم الخاص بك your user identity، ولكنه لن ينفذ execute البرامج النصية لتسجيل الدخول login scripts لمستخدم جديد .new user والذي يمكن أن يُسبب بعض المشاكل وخصوصاً عند التبديل إلى حساب الجذر .root account في معظم الحالات والخاصة بتبديل المستخدمين switch users فأنت ترغب بإتمام عملية performed environment variables تسجيل دخول كامل complete login بحيث يتم تهيئة بيئة المتغيرات environment variables والوظائف functions والأسماء المستعارة aliases لإنشاء مستخدم جديد ومن ثم إطلاقه initialize .

٣,٤,١٣ الخطوة ٣

والآن بعد أن قمت بتسجيل الدخول logged in باسم المستخدم الجذر root user، يجب أن تكون قادر على تنفيذ execute أوامر التكوين configure والخاصة بالوصول access إلى الأمر sudo:

```
visudo
```

ويكون التنفيذ مشابه لما يلي:

```
root@localhost:~# visudo
```

تنفيذ الأمر visudo

## ٤, ٤, ١٣ الخطوة ٤

لتمكين enable عملية الوصول الأساسي basic access لاستخدام الأمر sudo؛ يجب إضافة add السطر التالي (sysadmin ALL=(ALL) ALL) في الملف /etc/sudoers والذي تقوم بتحريره الآن. بما أنك في المحرر vi افتراضياً، اكتب ما يلي:

ماذا اكتب	المعنى
G	وضع الأوامر: الذهاب إلى السطر الأخير.
o	وضع الإدراج: لإدراج سطر جديد تحت السطر الحالي.
sysadmin ALL=(ALL) ALL	الدخول مسموح للوصول إلى نظام sudo
ESC	للخروج والعودة إلى وضع الأوامر.
:wq	لحفظ التغييرات والخروج من الملف بالكامل.

### جدول لتعديل sudo

```
#includedir /etc/sudoers.d

sysadmin ALL=(ALL) ALL

-- INSERT --
```

### تنفيذ أمر sudo

إذا قمت بكتابة ما سبق من بيانات data في الملف بشكل صحيح correctly؛ فستعود مرة أخرى إلى موجه الأوامر prompt في قشرة shell.

ولكن وفي بعض الأحيان عند الانتهاء التحرير تظهر رسالة ماذا الآن؟ عندها اضغط على مفتاح الإدخال Enter للاطلاع على الخيارات الخاصة بك. ومن المستحسن أن تكتب حرف x للخروج دون حفظ التغييرات على الملف sudoers، ثم قم بتكرار هذه الخطوة مرة أخرى.

## ٥, ٤, ١٣ الخطوة ٥

قم بالعودة إلى حساب sysadmin للتحقق من أن الأمر sudo يوفر الوصول للجذر root access، وذلك بالخروج من حساب الجذر root عن طريق كتابة الأمر التالي:

```
exit
```

ويكون التنفيذ مشابه لما يلي:

```
root@localhost:~# exit
logout
sysadmin@localhost:~$
```

تنفيذ الأمر exit

## ٦, ٤, ١٣ الخطوة ٦

حاول عرض الأسطر الثلاثة الأولى من الملف /etc/shadow، وهو الملف الذي يحتوي على كلمات السر المشفرة encrypted passwords ومعلومات حول عمر كلمة السر password aging (إلى متى تكون كلمة مرور صالحة):

```
head -3 /etc/shadow
```

ويكون التنفيذ مشابه لما يلي:

```
sysadmin@localhost:~$ head -3 /etc/shadow
head: cannot open `/etc/shadow' for reading:Permission denied
sysadmin@localhost:~$
```

تنفيذ الأمر head -3 /etc/shadow

لاحظ ظهور رسالة خطأ `error message` عند استخدامنا للأمر السابق؛ وذلك لأن حساب المستخدم `sysadmin` ليس لديه الصلاحية لعرض `view` هذا الملف، حيث يمكن عرض هذا الملف من قبل المستخدم الجذر `root user`.

٧، ٤، ١٣ الخطوة

لاحظ أن الأذونات `permissions` في ملف `/etc/shadow` تشير إلى أن أعضاء مجموعة `shadow` `group` فقط يكون لديهم الإذن `permission` لرؤية الملف `view the file`:

```
ls -l /etc/shadow
```

ضع في اعتبارك أنه يمكن للمستخدم الجذر `root user` عرض أي ملف. ويرجع ذلك إلى أن حساب الجذر لديه امتيازات خاصة `special privileges` تتجاوز `transcend` أذونات الملفات العادية `regular file` `permissions`.

```
sysadmin@localhost:~$ ls -l /etc/shadow
-rw-r----- 1 root shadow 838 Mar 14 2016 /etc/shadow
sysadmin@localhost:~$
```

تنفيذ الأمر `ls -l /etc/shadow`

٨، ٤، ١٣ الخطوة

استخدم الأمر `sudo` لعرض الأسطر الثلاثة الأولى من ملف `/etc/shadow`:

```
sudo head -3 /etc/shadow

{Enter the password: netlab123}
```

ويكون التنفيذ مشابه لما يلي:

```
sysadmin@localhost:~$ sudo head -3 /etc/shadow
[sudo] password for sysadmin:
root:$6$T3W2rbrt$N/2Jrt1EzQ8TqOvxWkYjEpIf3tCbPOyFwU7ZYkToosXB4AGmtb0.
W6f8Gb7Vmihnj76yZezNPwMbTGoQFs5Kx1:16874:0:99999:7:::
daemon:*:16863:0:99999:7:::
bin:*:16863:0:99999:7:::
sysadmin@localhost:~$
```

تنفيذ الأمر `sudo head -3 /etc/shadow` ، {Enter the password: netlab123}

ملاحظة هامة: كلمة السر التي قمت بكتابتها هي لحساب المستخدم `sysadmin` الخاص بك وليست لحساب الجذر `root account`. عندما يتم عمل التكوين `configured` للأمر `sudo` للحساب الخاص بك فإنك لا تحتاج إلى معرفة كلمة سر الجذر `root password` لتشغيل الأمر `sudo` كالمستخدم الجذر `root user`.

## المعمل الرابع عشر – إنشاء مستخدم

### Creating a User

## ١٤,١ مقدمة

هذا هو المعمل الرابع عشر: إنشاء مستخدم. عن طريق أداء هذا المعمل، سوف نتعلم كيفية إنشاء حساب لمستخدم جديد، وإنشاء كلمة مرور أولية لهذا الحساب، وإجراء تعديلات أخرى على المستخدم مثل جعله عضوا في مجموعة ثانوية.

في هذا التمرين، سوف نقوم بالمهام التالية:

- إنشاء مستخدم جديد مع الأمر `useradd`.
- تعيين وإعادة تعيين كلمة مرور المستخدم مع الأمر `passwd`.
- إجراء تغييرات على حساب المستخدم مع الأمر `usermod`.

## ١٤,٢ إنشاء المستخدمين والمجموعات

في هذه المهمة، سنقوم بإنشاء حسابات `create an accounts` لمجموعة `group` وللمستخدم `user`. حسابات المجموعة `group accounts` مفيدة وذلك لتعيين الأذونات `assign permissions` على الملفات المشتركة `shared files` من قبل مجموعة من المستخدمين `group of users`.

حسابات المستخدمين `user accounts` في توزيعات لينكس مبنية على `based upon` ريد هات Red Hat مثل توزيع CentOS، حيث تبدأ مع أول معرف للمستخدم (UID) بالرقم ٥٠٠، رمز المستخدم `User ID` التالي هو ٥٠١ وهلم جرا. العديد من التوزيعات الأخرى الحديثة تميل إلى أن يكون رمز المستخدم الأول `first UID` هو الرقم ١٠٠٠، والثاني ١٠٠١ وهلم جرا.

عند إدارة حسابات `managing accounts` لأنظمة متعددة `multiple systems`؛ فمن المستحسن أن تكون مصادقة الخادم مستندة إلى الشبكة `network-based authentication server`، حيث يمكن إنشاء الحسابات `created an accounts` دفعة واحدة ولكن تستخدم على أجهزة متعددة `many machines`. وإلا فستكون إدارة الحسابات المتعددة `managing multiple accounts` على آلات متعددة `on multiple`

machines تحدياً؛ حيث سيكون من الصعب التأكد ensure من ان المستخدم user وجميع المجموعات all the groups التي تنتمي إلى الكل لديها نفس قيمةUIDs وGIDs على جميع الآلات on all machines.

وثمة مسألة أخرى مع حسابات الجهاز المتعددة multiple machine accounts وهي انه من الصعب difficult محاولة الحفاظ على كلمات السر passwords لكل حساب متزامن account synchronized عبر كافة الأجهزة across all machines.

إدارة الحسابات managing accounts للمستخدمين المحليين local users لا تزال مفيدة من قبل الآلات الفردية individual machines، حتى إذا كان لديهم إمكانية الوصول access إلى مصادقة الخادم المستندة إلى الشبكة network-based authentication server. في هذا التمرين، سوف نقوم بإدارة المجموعة المحلية local group وحسابات المستخدمين user accounts.

١٤,٢,١ الخطوة ١

من أجل إدارة administer حسابات المستخدمين والمجموعات user and group accounts، سوف تحتاج إلى تبديل المستخدمين switch users إلى حساب الجذر root account باستخدام الأمر التالي:

```
su -  
{Enter the password: netlab123}
```

ويكون التنفيذ مشابه لما يلي:

```
sysadmin@localhost:~$ su -  
Password:  
root@localhost:~#
```

تنفيذ الأمر - su

## ٢,٢,١٤ الخطوة ٢

استخدم الأمر `groupadd` لإنشاء مجموعة `create a group` تسمى `research`:

```
groupadd -r research
```

ويكون التنفيذ مشابه لما يلي:

```
root@localhost:~# groupadd -r research
root@localhost:~#
```

تنفيذ الأمر `groupadd -r research`

إن استخدام الخيار `-r` مع الأمر السابق للمجموعة التي تم إضافتها وهي `research` يجعل معرف المجموعة `GID` في النطاق المحجوز `reserved range` بين `1-999`. يتم تعيين معرفات مجموعة (`GIDs`) تلقائياً `automatically` مع قيمة أقل من أقل قيمة المعرف `UID` للمستخدم العادي مع هذا الخيار. نقطة أخرى وهي أن الأمر `groupadd` يقوم بتعديل الملف `/etc/group` والذي يتم تخزين `store` المعلومات حساب المجموعة `group account information` فيه.

يمكن استخدام الأمر `groupmod` مع خيار `-n` لتغيير اسم هذه المجموعة `group name` أو الخيار `-g` من أجل تغيير المعرف `GID` لهذه المجموعة. أما لحذف المجموعة فاستخدم الأمر `groupdel` طالما أنه لم يتم جعلها المجموعة الأساسية `primary group` لمستخدم `user`.

## ٣,٢,١٤ الخطوة ٣

استخدم الأمر `getent` لاسترجاع `retrieve` معلومات من قاعدة بيانات أو ملف حول المجموعة الجديدة:

```
getent group research
```

يجب أن يظهر التنفيذ مشابه لما يلي، على الرغم من أن معرف المجموعة GID الذي تم تعيينه assigned قد يكون مختلف في الجهاز الذي تستخدمه:

```
root@localhost:~# getent group research
research:x:999:
root@localhost:~#
```

تنفيذ الأمر getent group research

والآن بعد أن تم إنشاء مجموعة research، فيمكنك إضافة مستخدمين حاليين existing or users أو جدد new كأعضاء member في هذه المجموعة. لإضافة مستخدم إلى مجموعة ثانوية يجب استخدام أمر التعديل على مستخدم usermod ثم الخيار a للإضافة وأخيراً الخيار -G والخاص بالمجموعات الثانوية. لاحظ أنه في حالة وجود أكثر من مجموعة ثانوية نريد إضافة المستخدم لها فيجب إضافة فاصلة " , " وذلك للفصل separated بين تلك المجموعات. أما عند استخدام الخيار -G بدون الخيار a فسيتم حذف جميع المجموعات الثانوية السابقة وإضافة الجديدة.

يجب أن يحتوي خيار -G مع الأمر usermod على قائمة مفصولة بفواصل لجميع المجموعات الثانوية التي ينتمي إليها المستخدم. وعند استخدام usermod مع الخيارات -a و -G عندها فقط المجموعة الجديدة سوف تحتاج إلى أن تكون محددة specified وسوف تضاف إلى عضوية مجموعة group memberships ثانوية موجودة existing secondary.

٤, ٢, ١٤ الخطوة ٤

استخدم الأمر usermod لإضافة المجموعة research كمجموعة ثانوية secondary group للمستخدم  
:sysadmin

```
usermod -aG research sysadmin
```

ويكون التنفيذ مشابه لما يلي:

```
root@localhost:~# usermod -aG research sysadmin
root@localhost:~#
```

تنفيذ الأمر usermod -aG research sysadmin

المستخدمين الذين قاموا بتسجيل دخول logged into تفاعلي actively في النظام system لن يكونوا قادرين على استخدام عضوية memberships أي مجموعة جديدة new group؛ إلا في المرة القادمة التي يقومون فيها بتسجيل الدخول إلى النظام.

٥، ٢، ١٤ الخطوة ٥

هناك العديد من الأوامر التي يمكن استخدامها للتحقق verify من عضوية membership المجموعة الجديدة new group. استخدم الأوامر التالية groups، id، getent للتحقق من عضويات المستخدم sysadmin's:

```
groups sysadmin
```

```
id sysadmin
```

```
getent group research
```

يجب أن يظهر إخراج الأوامر السابقة بالشكل التالي:

```
root@localhost:~# groups sysadmin
sysadmin : sysadmin adm sudo research

root@localhost:~# id sysadmin
uid=1001(sysadmin) gid=1001(sysadmin) groups=1001(sysadmin),4(adm),27(sudo),999(research)

root@localhost:~# getent group research
research:x:999:sysadmin

root@localhost:~#
```

تنفيذ الأوامر `groups sysadmin`، `id sysadmin`، `getent group research`

الأمر `useradd` يقوم بإنشاء `create` حساب مستخدم جديد `new user account`، وفي التوزيعات المبنية على ريد هات Red Hat يُستخدم للمجموعة الجديدة `new group` لهذا المستخدم `user`. وسيتم تسمية هذه المجموعة الجديدة باسم المستخدم وستكون المجموعة الأساسية `primary group` لذلك المستخدم.

أيضاً تستخدم التوزيعات المبنية على ريد هات Red Hat-based distributions ما يعرف باسم عضو المجموعات الخاصة `User Private Groups – UPG`، والتي تعني أن كل مستخدم هو عضو أساسي `primary member` في المجموعة الخاصة به.

أما التوزيعات `distributions` التي لا تستخدم التقنية السابقة `UPG`، فإن كل المستخدمين الجدد `all new users` ينتمون إلى `belong to` مجموعة المستخدمين `users group` كمجموعتهم الرئيسية `primary group`.

٦، ٢، ١٤ الخطوة ٦

أنشئ مستخدم جديد `new user` يسمى `student`، بحيث يكون عضو ثانوي `econdary member` في مجموعة `research` وعضو أساسي `primary member` في مجموعة خاصة به. استخدم التعليق `Linux Student` ليبدو وكأنه الاسم الكامل `full name` للمستخدم عند القيام بتسجيل الدخول في الواجهة الرسومية `graphical login`. تأكد من أن الدليل `home` الخاص به سيتم إنشاؤه بواسطة خيار `-m`:

```
useradd -G research -c 'Linux Student' -m student
```

ويكون التنفيذ مشابه لما يلي:

```
root@localhost:~# useradd -G research -c 'Linux Student' -m student
root@localhost:~#
```

تنفيذ الأمر `useradd -G research -c 'Linux Student' -m student`

يتم تخزين stored معلومات حساب المستخدم user's account information في الملفين `/etc/passwd` و `/etc/shadow`. ويمكن الاطلاع على معلومات المجموعة لمستخدم user's group information في الملفين `/etc/passwd` و `/etc/group`.

٧،٢،١٤ الخطوة ٧

باستخدام الأمر `getent`، اعرض أعضاء مجموعة `group members` والمسماة باسم `research` مرة أخرى، ثم استخدم نفس الأمر السابق `getent` ولكن لعرض أعضاء مجموعة `student`، وأخيراً قم بعرض قاعدتي البيانات `passwd` و `shadow` للمستخدم `student`:

```
getent group research
getent group student
getent passwd student
getent shadow student
```

ويكون التنفيذ مشابه لما يلي:

```
root@localhost:~# getent group research
research:x:999:sysadmin,student
root@localhost:~# getent group student
student:x:1002:
root@localhost:~# getent passwd student
student:x:1002:1002:Linux Student:/home/student:/bin/sh
root@localhost:~# getent shadow student
student:!:16902:0:99999:7:::
root@localhost:~#
```

تنفيذ الأوامر 'getent passwd student'، 'getent group student'، 'getent group research'،  
getent shadow student

يظهر في التنفيذ السابق أن كلا المستخدمين sysadmin و student هما عضوان ثانويان secondary members في مجموعة research.

معرف المجموعة GID لمجموعة student يطابق matches الحقل الرابع fourth field من المعلومات information الموجودة في الملف passwd. وهذا هو ما يجعل student عضو رئيسي primary member لمجموعة student.

وأخيراً، علامة التعجب " ! " والتي تظهر appear عند تنفيذ الأمر getent shadow student وتحديداً في حقل كلمة المرور password field وهو الحقل الثاني من الملف shadow؛ فهي للتوضيح أنه لم يتم تعيين set up كلمة المرور password للمستخدم student.

٨، ٢، ١٤ الخطوة ٨

استخدم الأمر passwd من أجل تعيين set كلمة المرور password للمستخدم student وهي {netlab123}، ثم بعد ذلك قم بعرض تلك الكلمة والتي من المفترض أن تكون قد أدخلت في الملف shadow للمستخدم student:

```
passwd student
```

```
Enter new UNIX password: netlab123
```

```
Retype new UNIX password: netlab123
```

```
getent shadow student
```

تنفيذ الأمر `getent shadow student` يُظهر كلمة المرور للمستخدم `student` والموجودة في قاعدة البيانات `/etc/shadow` ولكن بشكل مشفر، كما هو واضح في الحقل الثاني وتحديدًا بعد اسم المستخدم:

```
root@localhost:~# passwd student
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
root@localhost:~# getent shadow student
student:$6$pIIEdvAX$GBo0beYhojL3/vDrOP2UAQR6uVCWMZXxMPqImREJWw/5oR2WT
tM6dH3H83VjrmG6hGd9ux2I9FQFWQLtg3/:16902:0:99999:7:::
root@localhost:~#
```

تنفيذ الأوامر `passwd student`، `Enter new UNIX password: netlab123`، `Enter new UNIX password: netlab123`

`getent shadow student`، `UNIX password: netlab123`

٩، ٢، ١٤ الخطوة ٩

وجود كلمة مرور `password` للمستخدم `user` لا يعني بالضرورة أنه قام بأي تسجيل الدخول `logged into` للنظام. استخدم الأمر `last` لمعرفة ما إذا كان المستخدم `student` قد قام بأي تسجيل دخول في النظام:

```
Last
```

```
last student
```

يجب أن يظهر في التنفيذ أن المستخدم `sysadmin` فقط هو الذي قام بتسجيل الدخول `logged in` للنظام، أما المستخدم `student` فيجب ألا يظهر له أي تسجيل دخول للنظام:

```
root@localhost:~# last
sysadmin console          Mon Apr 11 19:08   still logged in
sysadmin console          Mon Apr 11 19:08 - 19:08   (00:00)

wtmp begins Mon Apr 11 18:25:26 2016
root@localhost:~# last student
wtmp begins Mon Apr 11 18:25:26 2016
root@localhost:~#
```

تنفيذ الأمرين `last student`،

وهناك أيضاً أمر `lastb`، والذي يعمل بشكل مشابه للأمر `last`؛ إلا أنه يُظهر محاولات تسجيل الدخول `log in` الفاشلة أو الغير صحيحة `bad or failed`.

إذا لم تعد ترغب في وصول المستخدم `student` ودخوله على النظام `system`؛ استخدم الأمر `usermod` والخاص بالتعديل على المستخدمين ثم الخيار `-L` والذي يستعمل لحجب أو قفل الحساب `lock the account` وأخيراً اسم المستخدم `student`. ويمكن فك الحجب `unlocked` بنفس الأمر السابق ولكن مع الخيار `-U` كما يلي `usermod -U student`.

الحل الدائم `permanent solution` لمنع الوصول `preventing access` للنظام من قبل الحساب الجديد والذي قمنا بإنشائه وهو `student` هو حذف `delete` ذلك الحساب عن طريق أحد الطريقتين التاليتين؛ الأولى مع الأمر `userdel student` والذي يقوم بحذف المستخدم فقط، أو الثانية وهي باستخدام نفس الأمر السابق ولكن بإضافة الخيار `-r` كما يلي `userdel -r student`. حيث أن إضافة الخيار `-r` مع الأمر `userdel` يقوم بحذف حساب المستخدم `deleting the user's account` بجانب إزالة الدليل الرئيسي للمستخدم `removes` `user's home directory` وأيضاً البريد `mail`.

١٠, ٢, ١٤ الخطوة ١٠

احذف الحساب student مع إزالة الدليل الرئيسي للمستخدم وايضاً البريد:

```
userdel -r student
```

تنفيذ الأمر ينبغي أن يبدو كما يلي:

```
root@localhost:~# userdel -r student  
root@localhost:~#
```

تنفيذ الأمر userdel -r student

## المعمل الخامس عشر – الملكيات والاذونات

# Ownership and Permissions

## ١٥,١ مقدمة

هذا هو المعمل الخامس عشر: الملكيات والاذونات. عن طريق أداء هذا التمرين سوف نتعلم كيفية إعداد وعرض الملكيات `ownerships` وأذونات `permissions` الملفات والدلائل.

في هذا التمرين، سوف نقوم بالمهام التالية:

- عرض وفهم الأذونات للملفات والمراجع.
- استخدام الأمر `chmod` لتغيير الأذونات.
- تغيير الملكيات مع الأوامر `chown` و `chgrp`.

## ١٥,٢ أذونات الملفات والملكيات

في هذه المهمة، سوف تقوم بإنشاء `create` الملفات والدلائل `files and directories`، ثم عرض `view` وتعيين `set` الأذونات والملكيات `permissions and ownerships`.

### ١٥,٢,١ الخطوة ١

انشئ اثنين دليلين وملفين في الدليل `/tmp` كما هو موضح:

```
cd /tmp  
  
mkdir priv-dir pub-dir  
  
touch priv-dir/priv-file  
  
touch pub-dir/pub-file
```

ويكون التنفيذ مشابه لما يلي:

```
sysadmin@localhost:~$ cd /tmp
sysadmin@localhost:/tmp$ mkdir priv-dir pub-dir
sysadmin@localhost:/tmp$ touch priv-dir/priv-file
sysadmin@localhost:/tmp$ touch pub-dir/pub-file
sysadmin@localhost:/tmp$
```

تنفيذ الأوامر 'cd /tmp' ، 'mkdir priv-dir pub-dir' ، 'touch priv-dir/priv-file' ، 'touch pub-dir/pub-file'

٢،٢،١٥ الخطوة ٢

عرض view محتويات الدلائل :directories contents

```
ls -l priv-dir
```

```
ls -l pub-dir
```

ويكون التنفيذ مشابه لما يلي:

```
sysadmin@localhost:/tmp$ ls -l priv-dir
total 0
-rw-rw-r-- 1 sysadmin sysadmin 0 Apr 11 21:26 priv-file
sysadmin@localhost:/tmp$ ls -l pub-dir
total 0
-rw-rw-r-- 1 sysadmin sysadmin 0 Apr 11 21:27 pub-file
sysadmin@localhost:/tmp$
```

تنفيذ الأمرين 'ls -l priv-dir' ، 'ls -l pub-dir'

الآن سنشرح النتيجة `1 sysadmin sysadmin 0 Apr 11 21:27 pub-file -rw-rw-r--` كما يلي:

ان الخانة الأولى `first character` من السطر تشير إلى نوع `type` هذا الكائن. ولنبداً بعلامة الشرطة أو الواصلة ( - ) `hyphen` وهي موجودة في بداية تفاصيل الملف السابق وتعني أن هذا الكائن هو ملف عادي `regular file`. اما إذا كانت الخانة الأولى هي الحرف `d` فهذا يشير إلى أن الكائن هو من نوع دليل `directory`، أما الحرف `l` فهو للدلالة على ارتباط رمزي `symbolic link`، وحرف `b` فهو لملف كتلة الجهاز `block device file`، ونجد ان الحرف `c` هو ملف خانة الجهاز `character device file`، وحرف `p` هو ملف الأنابيب `pipe file` و `s` هو ملف برنامج الاستماع `socket file`.

العلامة	نوع الكائن
علامة الشرطة أو الواصلة ( - ) <code>hyphen</code> .	ملف عادي.
الحرف <code>d</code> .	دليل <code>directory</code> .
الحرف <code>l</code> .	ارتباط رمزي <code>symbolic link</code> .
الحرف <code>b</code> .	ملف كتلة الجهاز <code>block device file</code> .
الحرف <code>c</code> .	ملف خانة الجهاز <code>character device file</code> .
الحرف <code>p</code> .	ملف الأنابيب <code>pipe file</code> .
الحرف <code>s</code> .	ملف برنامج الاستماع <code>socket file</code> .

#### جدول يوضح نوع الكائن

بعد الخانة الأولى لننتقل إلى الخانات التسعة `nine characters` التالية وهي `-rw-rw-r--`، تصنف تلك الخانات إلى ثلاث مجموعات `three groups`، كل مجموعة تتكون من ثلاثة خانات `three characters`، أول مجموعة في المثال السابق هي `(rw-)` وتعني أذونات المستخدم المالك `user owner's permissions` للكائن، أما المجموعة التي تليها وهي `(rw-)` فتعني أذونات مجموعة المالكين `group owner's permissions` للكائن (المجموعة التي ينتمي إليها المستخدم المالك للملف)، وأخيراً المجموعة الثالثة وهي `(-)`

(r- وتمثل أذونات الآخرين others permissions ) أذونات أي شخص آخر everyone else's permissions).

ولكن ما معنى تلك الرموز؟ يدل الحرف r على إذن القراءة read permission، ثم بعد ذلك الحرف w والذي يدل على إذن الكتابة write permission، واخيراً الحرف x والذي يشير إلى التنفيذ execute permission، أما إذا وجدنا علامة شرطة - موجودة في أحد تلك الخانات التسعة السابقة وهي rw-rw-r-- فهذا يشير إلى سحب تلك الأذونات السابقة منه (سواءً كان للمالك ورمزه u أو للمجموعة ورمزها g أو للآخرين ورمزهم o).

الآن نتطرق في الحديث عن الحقل الثاني وهو رقم حساب الارتباط link count number، وكما نلاحظ في المثال السابق أنه رقم (1) وهو يشير إلى عدد الملفات files numbers المرتبطة linked بهذا الملف file. بعد ملفات الارتباط نرى المستخدم المالك user owner للملف file وهو في المثال السابق sysadmin. بعد ذلك نجد المجموعة التي ينتمي إليها المستخدم المالك وهي مجموعة المالكين group owner للملف file؛ والموجودة في المثال السابق هي sysadmin.

أما النقطة التالية فهي حجم الملف size of the file وهو في المثال السابق (0).

بعد الحجم نشاهد التاريخ والوقت date/time لآخر تعديل last modified تم على الملف.

واخيراً نجد اسم الملف file name.

٣, ٢, ١٥ الخطوة ٣

إذا كنت تريد أن تجعل دليل ما أكثر خصوصية more private؛ استعمال الأمر chmod لإزالة الأذونات remove permissions عن ذلك الدليل. استخدم الأمر chmod لإزالة أذونات الآخرين من قراءة وتنفيذ:

```
ls -ld priv-dir/
```

```
chmod o-rx priv-dir/
```

```
ls -ld priv-dir/
```

يُظهر الأمر الأخير أن الآخرين ليس لديهم إذن permission أو الوصول access إلى الدليل priv-dir:

```
sysadmin@localhost:/tmp$ ls -ld priv-dir/
drwxrwxr-x 2 sysadmin sysadmin 4096 Apr 11 21:26 priv-dir/
sysadmin@localhost:/tmp$ chmod o-rx priv-dir/
sysadmin@localhost:/tmp$ ls -ld priv-dir/
drwxrwx--- 2 sysadmin sysadmin 4096 Apr 11 21:26 priv-dir/
sysadmin@localhost:/tmp$
```

تنفيذ الأوامر 'ls -ld priv-dir/'، 'chmod o-rx priv-dir/'، 'ls -ld priv-dir/'

يمكنك استخدام الأمر chmod لتعديل أذونات الآخرين others باستخدام الرمز الذي يمثلهم وهو الحرف o؛ متبوع بأحد العلامات الثلاث وهي علامة زائد + لإضافة الأذونات، أو علامة ناقص - لإزالة الأذونات، وأخيراً علامة يساوي = والتي لها طريقتين في العمل، الأولى مع التحديد للجميع a حيث تقوم بإعطاء الصلاحيات المذكورة مع الأمر من قراءة وكتابة وتنفيذ وإزالة جميع الأذونات السابقة التي لم تُذكر ووضع علامة الشرطة مكان الأذن للدلالة على إزالته. أما الاستخدام الثاني فهو مع أحد التحديدات التالية u أو g أو o حيث تقوم بإعطاء الصلاحيات المذكورة مع الأمر من قراءة وكتابة وتنفيذ وإزالة جميع الأذونات السابقة التي لم تُذكر وبالتالي وضع علامة الشرطة مكان الأذن للدلالة على إزالته ولكن فقط للفئة المحددة (المالك أو المجموعة أو الآخرين).

أما الرمز الذي يمثل المستخدم المالك user owner للملف فهو الحرف u والمستخدم مع الأمر chmod الخاص بتعديل modify الأذونات permissions. وأخيراً الحرف g والذي يمثل مجموعة المالكين group .owner

لتعديل أذونات الجميع استخدم الحرف a بدلا من الحروف التالية o ، u ، g.

بعض الأمثلة على تعديل الأذونات (قراءة، كتابة، تنفيذ) وذلك للمستخدم (المالك، المجموعة، الآخرين):

المعنى	الأمر chmod مع حرف
يعطي الجميع إذن التنفيذ.	<code>chmod a+x file name</code>
يزيل من المجموعة إذن الكتابة.	<code>chmod g-w file name</code>
إضافة للمجموعة والآخرين إذن القراءة.	<code>chmod go+r file name</code>
إعطاء للآخرين اذونات قراءة وكتابة وتنفيذ. وعدم تغيير الأذونات للفئتين الباقية وهي المالك والمجموعة.	<code>chmod o=rwx file name</code>

جدول أمثلة على تعديل الأذونات باستخدام الأمر `chmod`

٤, ٢, ١٥ الخطوة ٤

إذا كنت تريد أن تجعل دليل ما أكثر عمومية `more public`؛ استخدم الأمر `chmod` لإضافة إذن الكتابة للآخرين `others`:

```
ls -ld pub-dir/
```

```
chmod o+w pub-dir/
```

```
ls -ld pub-dir/
```

يظهر في التنفيذ `output` الآن أن الآخرين `others` لديهم إذن للكتابة `write permission` على الدليل `directory` (ونقصد بالكتابة القدرة `ability` على إضافة `add` أو حذف `delete` الملفات داخل الدليل):

```
sysadmin@localhost:/tmp$ ls -ld pub-dir/
drwxrwxr-x 2 sysadmin sysadmin 4096 Apr 11 21:27 pub-dir/
sysadmin@localhost:/tmp$ chmod o+w pub-dir/
sysadmin@localhost:/tmp$ ls -ld pub-dir/
drwxrwxrwx 2 sysadmin sysadmin 4096 Apr 11 21:27 pub-dir/
sysadmin@localhost:/tmp$
```

تنفيذ الأوامر 'ls -ld pub-dir/' ، 'chmod o+w pub-dir/' ، 'ls -ld pub-dir/'

١٥,٢,٥ الخطوة ٥

استخدم الأمر 'chmod لإزالة remove أي إذن من المجموعة أو من الآخرين على الملف priv-file:

```
ls -l priv-dir/priv-file

chmod g-rw,o-r priv-dir/priv-file

ls -l priv-dir/priv-file
```

يجب أن تظهر التنفيذ مشابه لما يلي:

```
sysadmin@localhost:/tmp$ ls -l priv-dir/priv-file
-rw-rw-r-- 1 sysadmin sysadmin 0 Apr 11 21:26 priv-dir/priv-file
sysadmin@localhost:/tmp$ chmod g-rw,o-r priv-dir/priv-file
sysadmin@localhost:/tmp$ ls -l priv-dir/priv-file
-rw----- 1 sysadmin sysadmin 0 Apr 11 21:26 priv-dir/priv-file
sysadmin@localhost:/tmp$
```

تنفيذ الأوامر 'ls -l priv-dir/priv' ، 'o-r priv-dir/priv-file' ، 'chmod g-rw' ، 'ls -l priv-dir/priv-file'

-file

## ٦، ٢، ١٥ الخطوة ٦

اعطي grant كافة المستخدمين all users أذونات القراءة والكتابة للملف pub-file (وسيم إزالة الأذن الذي لم يُذكر وهو التنفيذ من الجميع):

```
ls -l pub-dir/pub-file

chmod a=rw pub-dir/pub-file

ls -l pub-dir/pub-file
```

يجب أن تظهر التنفيذ مشابه لما يلي:

```
sysadmin@localhost:~/tmp$ ls -l pub-dir/pub-file
-rw-rw-r-- 1 sysadmin sysadmin 0 Apr 11 21:27 pub-dir/pub-file
sysadmin@localhost:~/tmp$ chmod a=rw pub-dir/pub-file
sysadmin@localhost:~/tmp$ ls -l pub-dir/pub-file
-rw-rw-rw- 1 sysadmin sysadmin 0 Apr 11 21:27 pub-dir/pub-file
sysadmin@localhost:~/tmp$
```

تنفيذ الأوامر الثلاثة ls -l pub-dir/pub-file ، chmod a=rw pub-dir/pub-file ، ls -l pub-dir/pub-file  
ub-file

## ٧، ٢، ١٥ الخطوة ٧

أنشأ الملف test.sh في /tmp والذي يحتوي على أمر التاريخ "date":

```
echo "date" > test.sh
```

ويكون التنفيذ مشابهه لما يلي:

```
sysadmin@localhost:/tmp$ echo "date" > test.sh
sysadmin@localhost:/tmp$
```

تنفيذ الأمر `echo "date" > test.sh`

إذا كان ملف ما يحتوي على أوامر `commands`؛ فإن تلك الأوامر يمكن تشغيلها `run` إذا تم منح إذن التنفيذ `execute permission` عليها المستخدم (مالك، مجموعة، آخرين). وبالتالي يستطيع المستخدم التشغيل ويصبح الملف قابل للتنفيذ `executable file`. أما في حالة عدم وجود هذا الإذن؛ فإن الملف لا يمكن أن تتم معاملته كبرنامج `program`.

٨، ٢، ١٥ الخطوة ٨

حاول تنفيذ الملف `test.sh` (ويجب أن تبوء المحاولة بالفشل). ثم اعرض الأذونات `permissions` على الملف لترى سبب الفشل:

```
./test.sh
```

```
ls -l test.sh
```

ويكون التنفيذ مشابهه لما يلي:

```
sysadmin@localhost:/tmp$ ./test.sh
-bash: ./test.sh: Permission denied
sysadmin@localhost:/tmp$ ls -l test.sh
-rw-rw-r-- 1 sysadmin sysadmin 5 Apr 11 22:27 test.sh
sysadmin@localhost:/tmp$
```

تنفيذ الأمرين `./test.sh` ، `ls -l test.sh`

## ٩, ٢, ١٥ الخطوة ٩

يُسمح فقط للمستخدم المالك user owner للملف (أو المستخدم الجذر root) بتغيير الأذونات على الملف.  
اعطي المستخدم المالك إذن التنفيذ execute permission، ثم نفذ الملف test.sh:

```
chmod u+x test.sh
```

```
ls -l test.sh
```

```
./test.sh
```

التنفيذ يُظهر إضافة إذن التنفيذ للمستخدم المالك user owner للملف وأيضاً التاريخ والوقت الحالي لتنفيذ الأمر  
date داخل الملف النصي script file السابق test.sh:

```
sysadmin@localhost:~/tmp$ chmod u+x test.sh
sysadmin@localhost:~/tmp$ ls -l test.sh
-rwxrw-r-- 1 sysadmin sysadmin 5 Apr 11 22:27 test.sh
sysadmin@localhost:~/tmp$ ./test.sh
Mon Apr 11 22:35:23 UTC 2016
sysadmin@localhost:~/tmp$
```

تنفيذ الأوامر chmod u+x test.sh، ls -l test.sh، ./test.sh

في الفقرة السابقة تعلمت كيفية استخدام الأمر chmod مع التدوين الرمزي symbolic notation، حيث يتم استخدام رموز symbols للتعبير عن الفئة وهي (u، g، o، a)، وللتعبير عن عملية المنح أو الإزالة تم استخدام الرموز (+، -، =)، وأخيراً وللتعبير عن الأذن تم استخدام الرموز (r، w، x). ولكن؛ وبجانب استخدام التدوين الرمزي، فإن الأمر chmod يمكن استخدامه مع قيمة رقمية (ليست رمزية) تمثل أيضاً أذونات المستخدم المالك والمجموعة والآخرين وتعرف باسم التدوين الثماني octal notation.

لإستخدام الأمر chmod مع التدوين الثماني، يجب عليك أولاً فهم قيمة الثمانية من الأذونات:

الإذن	القيمة
قراءة (r) Read (r)	٤
كتابة (w) Write (w)	٢
تنفيذ (x) Execute (x)	١

#### جدول القيمة الثمانية

من تفاصيل الملف السابق `test.sh`، نرى الأذونات كالتالي `rw-rw-r--`، حيث أن `rw` للمستخدم المالك، و `rw` للمجموعة، و `r` للآخرين. وللتعبير عن هذه الأذونات بالتدوين ثماني `octal notation`؛ سيتم جمع القيمة العددية لكل إذن.

ولنبدأ أولاً بأذونات المستخدم المالك `user's permission`؛ حيث أن الأذونات التي لديه `rw` تكون قيمتها `٤،٢،١` على التوالي (٤ للقراءة، ٢ للكتابة و ١ للتنفيذ)، نقوم الآن بعملية جمع لتلك القيم كما يلي  $٤ + ٢ + ١ = ٧$ ، فتصبح قيمة الأذونات بالتدوين الثماني للمالك هي `٧`.

ثانياً، لننتقل إلى أذونات المجموعة `group's permission`، حيث أن للمجموعة الأذونات التالية `rw` وتكون قيمتها `٤،٢` على التوالي (٤ للقراءة، ٢ للكتابة)، نقوم الآن بعملية جمع لتلك القيم كما يلي  $٤ + ٢ = ٦$ . إذن القيمة الثمانية لأذونات المجموعة هي `٦`.

وأخيراً، أذونات ملكية الآخرين `others' ownership permission`، ببساطة الآخرين لديهم إذن القراءة `r` وقيمه `٤`، وبالتالي فإن القيمة الثمانية لأذونات الآخرين هي `٤`.

الخطوة الأخيرة هي وضع تلك النتائج معاً (مجموع المالك، مجموع المجموعة، مجموع الآخرين)؛ فتصبح القيمة الثمانية `octal value` للأذونات الملف هي `٧٦٤`.

١٠،٢،١٠ الخطوة ١٠

استخدم الأمر `stat` لرؤية القيمة الثمانية `octal value` للأذونات `permissions` الملف `test.sh`، وبالتالي التحقق `verify` من حسابات الخطوة السابقة:

stat test.sh

ويكون التنفيذ مشابه لما يلي:

```
sysadmin@localhost:~/tmp$ stat test.sh
  File: `test.sh'
  Size: 5             Blocks: 8           IO Block: 4096   regular file
Device: fc00h/64512d  Inode: 3540004      Links: 1
Access: (0764/-rwxrw-r--)  Uid: ( 1001/sysadmin)  Gid: ( 1001/sysadmin)
Access: 2016-04-11 22:27:24.987740243 +0000
Modify: 2016-04-11 22:27:24.987740243 +0000
Change: 2016-04-11 22:35:08.335757863 +0000
 Birth: -
sysadmin@localhost:~/tmp$
```

تنفيذ الأمر stat test.sh

إذا كنت تريد تغيير هذه الأذونات permissions باستخدام الترميز الثماني octal notation وذلك لإعطاء

المجموعة group والآخرين others إذن التنفيذ، فإنك سوف تستخدم الأرقام الثلاثة التالية:

٧ (قراءة، كتابة وتنفيذ) للمالك.

٧ (قراءة، كتابة وتنفيذ) للمجموعة.

٥ (قراءة وتنفيذ) للآخرين.

إذن يصبح الوضع الجديد new mode، أو القيمة الثمانية للأذونات الملف test.sh هي ٧٧٥.

١١، ٢، ١٥ الخطوة ١١

باستخدام الترميز الثماني octal notation قم بتعديل modify أذونات الملف test.sh، ليكون جميع

المستخدمين قادرين على تنفيذ الملف:

```
chmod 775 test.sh
```

```
ls -l test.sh
```

ويكون التنفيذ مشابهه لما يلي:

```
sysadmin@localhost:~/tmp$ chmod 775 test.sh
sysadmin@localhost:~/tmp$ ls -l test.sh
-rwxrwxr-x 1 sysadmin sysadmin 5 Apr 11 22:27 test.sh
sysadmin@localhost:~/tmp$
```

تنفيذ الأمر 'chmod 775 test.sh'، 'ls -l test.sh'

هناك نوعان من الأوامر التي يمكن أن تؤثر affect على ملكية الملفات files ownership، الأمر الأول هو chown ومن خلاله يمكن تغيير مالك الملف أو تغيير المالك والمجموعة اللذان يملكان الملف. ولكن لا يمكن تنفيذ ذلك الأمر إلا من قبل المستخدم الجذر root.

الأمر الثاني والذي يؤثر على ملكية الملف هو الأمر chgrp والذي يمكن استخدامه من قبل المستخدم الذي يملك الملف أو من قبل المستخدم الجذر root.

حيث يقوم الأمر chgrp بتغيير المجموعة التي تملك الملف فقط وليس مالك الملف.

عندما يُستعمل الأمر chgrp من قبل مستخدم غير رئيسي non-root user يمكنه فقط تغيير ملكية المجموعة إلى مجموعة يكون ذلك المستخدم عضواً بها. بينما يستطيع المستخدم الرئيسي root user استخدام الأمر chgrp وذلك لتغيير ملكية المجموعة إلى أي مجموعة أخرى.

١٢، ٢، ١٥ الخطوة ١٢

قم بتبديل المستخدم والدخول الى حساب المستخدم الرئيسي root user لتكون قادراً على تنفيذ كل من الأمرين chown و chgrp لتغيير ملكية المجموعة إلى أي مجموعة:

su -

{Enter the password: netlab123}

ويكون التنفيذ مشابه لما يلي:

```
sysadmin@localhost:~/tmp$ su -
Password:
root@localhost:~#
```

تنفيذ الأمر - su

١٣، ٢، ١٥ الخطوة

انتقل إلى الدليل /tmp، ثم قم بسرد تفاصيل الدليل pub-dir، ثم قم بسرد تفاصيل ما بداخله:

```
cd /tmp
```

```
ls -ld pub-dir
```

```
ls -l pub-dir/pub-file
```

لاحظ يظهر في التنفيذ أن الدليل pub-dir والملف الذي بداخله pub-file مملوكان للمستخدم sysadmin،

ولمجموعة sysadmin:

```
root@localhost:~# cd /tmp
root@localhost:~/tmp# ls -ld pub-dir/
drwxrwxrwx 2 sysadmin sysadmin 4096 Apr 11 22:48 pub-dir/
root@localhost:~/tmp# ls -l pub-dir/pub-file
-rw-rw-rw- 1 sysadmin sysadmin 0 Apr 11 22:48 pub-dir/pub-file
root@localhost:~/tmp#
```

تنفيذ الأوامر cd /tmp، ls -ld pub-dir، ls -l pub-dir/pub-file

## ١٤, ٢, ١٥ الخطوة ١٤

استخدم الأمر `chown` لتغيير المالك والمجموعة للدليل `pub-dir` إلى المستخدم الرئيسي `root user` والمجموعة الرئيسية `root group`، ثم قم بعرض تفاصيل الدليل:

```
chown root:root pub-dir
```

```
ls -ld pub-dir
```

و ينبغي أن يظهر في التنفيذ تغيير المالك والمجموعة:

```
root@localhost:~/tmp# chown root:root pub-dir
root@localhost:~/tmp# ls -ld pub-dir/
drwxrwxrwx 2 root root 4096 Apr 11 22:48 pub-dir/
root@localhost:~/tmp#
```

تنفيذ الأمرين `chown root:root pub-dir` و `ls -ld pub-dir`

## ١٥, ٢, ١٥ الخطوة ١٥

استخدم الأمر `chown` لتغيير مالك الملف `pub-file` إلى المستخدم `bin`:

```
chown bin pub-dir/pub-file
```

```
ls -l pub-dir/pub-file
```

يظهر التنفيذ الآن أن المالك للملف هو المستخدم `bin`:

```
root@localhost:~/tmp# chown bin pub-dir/pub-file
root@localhost:~/tmp# ls -l pub-dir/pub-file
-rw-rw-rw- 1 bin sysadmin 0 Apr 11 22:48 pub-dir/pub-file
root@localhost:~/tmp#
```

تنفيذ الأمرين `chown bin pub-dir/pub-file` ، `ls -l pub-dir/pub-file`

١٦، ٢، ١٥ الخطوة ١٦

اعرض تفاصيل الدليل `priv-dir`، ثم قم بعرض تفاصيل محتوياته:

```
ls -ld priv-dir
```

```
ls -l priv-dir/priv-file
```

التنفيذ `output` ينبغي أن يُظهر أن الدليل `priv-dir` تعود ملكيته إلى المستخدم `sysadmin user` وإلى مجموعة `sysadmin`:

```
root@localhost:~/tmp# ls -ld priv-dir
drwxrwx--- 2 sysadmin sysadmin 4096 Apr 11 22:48 priv-dir
root@localhost:~/tmp# ls -l priv-dir/priv-file
-rw----- 1 sysadmin sysadmin 0 Apr 11 22:48 priv-dir/priv-file
root@localhost:~/tmp#
```

تنفيذ الأمرين `ls -ld priv-dir` ، `ls -l priv-dir/priv-file`

١٧، ٢، ١٥ الخطوة ١٧

يستخدم الخيار `-R` مع الأمر `chgrp` لتغيير المجموعة للدليل ولكل الملفات التي بداخله، غير المجموعة للدليل `priv-dir` وما بداخله (الملف `priv-file`) إلى مجموعة `users` باستخدام الأمر `chgrp` بشكل متكرر `recursively` باستخدام الخيار `-R` ؛ ثم قم بعرض التفاصيل للدليل السابق:

```
ls -ld priv-dir
```

```
ls -l priv-dir/priv-file
```

```
chgrp -R users priv-dir
```

```
ls -ld priv-dir
```

```
ls -l priv-dir/priv-file
```

ويكون التنفيذ مشابه لما يلي:

```
root@localhost:~/tmp# ls -ld priv-dir
drwxrwx--- 2 sysadmin sysadmin 4096 Apr 11 22:48 priv-dir
root@localhost:~/tmp# ls -l priv-dir/priv-file
-rw----- 1 sysadmin sysadmin 0 Apr 11 22:48 priv-dir/priv-file
root@localhost:~/tmp# chgrp -R users priv-dir
root@localhost:~/tmp# ls -ld priv-dir
drwxrwx--- 2 sysadmin users 4096 Apr 11 22:48 priv-dir
root@localhost:~/tmp# ls -l priv-dir/priv-file
-rw----- 1 sysadmin users 0 Apr 11 22:48 priv-dir/priv-file
root@localhost:~/tmp#
```

تنفيذ الأوامر `ls -ld priv-dir`، `ls -l priv-dir/priv-file`، `chgrp -R users priv-dir`،

`ls -l priv-dir/priv-file`، `dir`

يظهر في النتيجة السابقة أنه عند تنفيذ تغييرات على دليل بشكل متكرر `recursively`، فإن التغييرات تطبق على الدليل وكل محتوياته. بمعنى أن كل الملفات والأدلة الفرعية `subdirectories` التي تحت الدليل `priv-dir` وكل محتوياتها من ملفات وأدلة فرعية سيطبق عليها هذا التغيير.

المعمل السادس عشر – أذونات خاصة، روابط

ومواقع الملفات

Special Permissions, Links and File  
Locations

## ١٦,١ مقدمة

هذا هو المعمل السادس عشر: أذونات خاصة وروابط ومواقع الملفات. عن طريق أداء هذا التمرين، سوف نتعلم عن الأذونات الخاصة `special permissions` وأنواع مختلفة من ملفات الارتباط `link files`.

في هذا التمرين، سوف تقوم بالمهام التالية:

- عرض ملفات ذات أذونات خاصة.
- إنشاء الروابط الصلبة `hard` واللينة `soft`.

## ١٦,٢ الاطلاع على الأذونات الخاصة

في هذه المهمة سوف تفهم ما هو الغرض من الأذونات الخاصة `special permissions` غير القراءة والكتابة والتنفيذ.

### ١٦,٢,١ الخطوة ١

اسرد تفاصيل الدليل `/tmp` والدليل `/var/tmp`:

```
ls -ld /tmp
```

```
ls -ld /var/tmp
```

يظهر في التنفيذ `output` أن الأذونات `permissions` على الدليلين السابقين متماثلة كما هو واضح:

```
sysadmin@localhost:~$ ls -ld /tmp
drwxrwxrwt 2 root root 4096 Mar 14 17:34 /tmp
sysadmin@localhost:~$ ls -ld /var/tmp
drwxrwxrwt 2 root root 4096 Apr 19 2012 /var/tmp
sysadmin@localhost:~$
```

تنفيذ الأمرين `ls -ld /tmp`، `ls -ld /var/tmp`

الأذونات في الدليل `/tmp` والدليل `/var/tmp` من قراءة وكتابة وتنفيذ معطاء لجميع المستخدمين. بجانب دلائل المستخدمين الرئيسية `users' home directories`؛ يوجد هذان الدليلان المؤقتان `temporary directories` في نظام الملفات `filesystem`، حيث يمكن للمستخدمين العاديين `ordinary users` إنشاء ملفات جديدة أو دلائل.

وهذا يشكل مشكلة `problem`؛ فإذا كان لكافة المستخدمين `all users` أذونات لإنشاء ملفات جديدة، سيكون لديهم أيضا إمكانية حذف `delete` الملفات الموجودة `existing files`. وذلك لأن إذن الكتابة على دليل ما يمنح المستخدمين القدرة على إضافة وحذف الملفات في ذلك الدليل.

لاحظ وجود حرف `t` في مكان خانة التنفيذ `execute column` لأذونات الآخرين `others permissions` في المثال السابق (`drwxrwxrwt`)، حيث يشير إلى أن هذا الدليل لديه مجموعة أذونات من نوع البت اللاصق `sticky bit`. هذا الإذن الخاص يعني أنه على الرغم من أن الجميع قادر على إضافة ملفات إلى دليل ما؛ إلا أن المستخدم الذي أنشأ الملف ذلك الملف هو الذي يمكنه حذفه.

وتجدر الإشارة إلى أن المستخدم الرئيسي الجذر `root user` لا يتأثر بهذا الإذن؛ حيث يستطيع حذف جميع الملفات في أي دليل بغض النظر عن ملكيتها `ownership`.

٢,٢,١٦ الخطوة ٢

اعرض الأذونات `permissions` على الملف `/etc/shadow`:

ls -l /etc/shadow

نرى في التنفيذ التالي (rw-r----- 1 root shadow) ونجد أن المالك هو المستخدم الجذر root user ولديه إذن قراءة r وكتابة w، كما يظهر في التنفيذ المجموعة وهي shadow ولأعضائها إذن القراءة r، وأخيراً نجد الآخرين others بدون اي أدونات على الملف:

```
sysadmin@localhost:~$ ls -l /etc/shadow
-rw-r----- 1 root shadow 838 Mar 14 17:34 /etc/shadow
sysadmin@localhost:~$
```

تنفيذ الأمر ls -l /etc/shadow

مثل الملفات الأخرى الموجودة في دليل /etc، الملف /etc/shadow يحتوي على معلومات معينة لتكوين المضيف host-specific configuration information. وتحديداً، يحتوي ملف /etc/shadow على كلمات السر المشفرة encrypted passwords لجميع حسابات all accounts المستخدمين المحليين local user ومعلومات عن عمر كلمة السر password aging (إلى متى تكون كلمة مرور صالحة). وبما أن هذه المعلومات حساسة جداً very sensitive information؛ فإن الوصول access إليها محدد limited بالمستخدم الجذر root user؛ وبالأوامر التي تُنفذ بصلاحياته as root؛ وأيضاً أعضاء members مجموعة shadow.

يتم تحديث updates كلمة المرور password الخاصة بالمستخدمين عن طريق الأمر passwd الذي لديه إذن خاص يسمى setuid. إذن setuid يعطي الصلاحية للتنفيذ كأنك المستخدم الذي يملك الملف shadow (وهو المستخدم root)، وذلك للمستخدم الفعلي الذي أصدر الأمر.

إذا كانت لديك خبرة في نظام التشغيل مايكروسوفت ويندوز Microsoft Windows؛ فيمكنك اعتبار الإذن setuid كأنه مماثل للأمر " تشغيل كمسؤول النظام Run as administrator".

### ٣, ٢, ١٦ الخطوة ٣

اعرض أذونات permissions للأمر `:/usr/bin/passwd`:

```
ls -l /usr/bin/passwd
```

وينبغي أن يظهر التنفيذ كما يلي:

```
sysadmin@localhost:~$ ls -l /usr/bin/passwd
-rwsr-xr-x 5 root root 42824 Sep 12 2012 /usr/bin/passwd
sysadmin@localhost:~$
```

تنفيذ الأمر `ls -l /usr/bin/passwd`

لاحظ وجود حرف `s` في مكان خانة التنفيذ للمستخدم المالك `rws`، ويدل هذا الحرف على أن هذا الأمر يحتوي على مجموعة أذونات من نوع `setuid`، وبالتالي سيتم تنفيذ الأمر `passwd` من قبل المستخدم الفعلي المُشغل للأمر وكأنه المستخدم الذي يملكه (وهو المستخدم الجذر).

ولهذا فإن الأمر `passwd` قادر على تحديث `update` الملف `/etc/shadow` لأنه ينفذ باسم المستخدم `root`. (تذكر أن المستخدم الجذر `root user` يمكنه تحرير `edit` أي ملف، بغض النظر عن الأذونات `permissions` الموجودة عليه).

### ٤, ٢, ١٦ الخطوة ٤

اعرض الأذونات permissions للأمر `:/usr/bin/wall`:

```
ls -l /usr/bin/wall
```

يظهر في التنفيذ حرف s؛ وتحديداً في مكان خانة التنفيذ لأذونات المجموعة r-s كما يلي:

```
sysadmin@localhost:~$ ls -l /usr/bin/wall
-rwxr-sr-x 5 root tty 18976 Jun 18 2014 /usr/bin/wall
sysadmin@localhost:~$
```

تنفيذ الأمر `ls -l /usr/bin/wall`

حرف s الموجود في عمود التنفيذ للمجموعة يشير إلى أن هذا الأمر يحتوي على مجموعة أذونات من نوع `setgid`، لذلك فإنه سيُنفذ من قبل أعضاء المجموعة المُشغلين للأمر وكأنهم من مجموعة المالكين (`tty`). وهكذا فإن الأمر `wall` يكون قادراً `able` على الكتابة `write` إلى جميع النوافذ الطرفية (`terminals (ttys)` حيث أنه يُنفذ كمجموعة `tty`.

ملاحظة: الإذن `setgid` مشابه جداً للإذن `setuid`؛ ولكن بدلاً من تشغيل الأمر بصفة المستخدم مالك البرنامج، يُشغل الأمر بصفة المجموعة المالكة للبرنامج.

حتى الآن، شاهدنا ثلاثة أنواع من الأذونات الخاصة: `sticky bit` على دليل، `setuid` على ملف قابل للتنفيذ `executable file` والإذن `setgid` على ملف قابل للتنفيذ `executable file` أيضاً. وكما شاهدنا؛ فإن جميع تلك الأذونات موجودة `exist` على أي نظام نموذجي `typical system`.

بجانب ما سبق فإنه يوجد إذن خاص `special permission` آخر يمكن استخدامه وهو إذن `setgid` والذي يمكن تطبيقه `applied` على دليل `directory`.

فإذا رأيت حرف s في مكان خانة التنفيذ لأذونات المجموعة المالكة للدليل؛ فهذا يشير إلى أن الدليل لديه إذن `setgid`. وبالتالي فإن أي ملف أو دليل جديد يتم إنشاؤه في هذا الدليل سيكون تلقائياً مملوكاً من قبل المجموعة التي تملك الدليل، وليس من قبل أعضاء المجموعة الذين أنشأوا الملف.

## ١٦,٣ الروابط الصلبة واللينة

في هذه المهمة، سنقوم بإنشاء واستخدام الروابط links الصلبة واللينة hard and soft.

### ١٦,٣,١ الخطوة ١

انتقل الى الدليل الرئيسي home directory:

```
cd
```

### ١٦,٣,٢ الخطوة ٢

قم بإنشاء ملف يسمى source يحتوي على النص التالي "data" وذلك باستخدام أمر إعادة التوجيه:

```
echo "data" > source
```

### ١٦,٣,٣ الخطوة ٣

اعرض تفاصيل وأيضاً معلومات inode للملف source:

```
ls -li source
```

نرى في التنفيذ جزئيين مظللين باللون الأصفر، الأول في بداية التنفيذ ويشير إلى مؤشر الفهرسة inode للملف وهو ٢٠٧٦ والذي يختلف من نظام إلى آخر، والثاني نراه بعد أذونات الملف وهو يشير الى عدد الروابط link count لهذا الملف وهو كما نرى الرقم ١:

```
sysadmin@localhost:~$ ls -li source
2076 -rw-rw-r-- 1 sysadmin sysadmin 5 Apr 12 13:27 source
sysadmin@localhost:~$
```

### تنفيذ الأمر ls -li source

يستخدم نظام التشغيل لينكس مؤشر الفهرسة inodes لمواصلة تتبع keep track المعلومات حول الملف. نظام مُدخلات الدليل directory entry يربط associates مؤشر الفهرسة inode مع اسم الملف. أن إنشاء ارتباط صلب hard link بين كائنين أو أكثر يجعل نظام مُدخلات الدليل directory entry يقوم بربط الكائن الجديد الذي نريد عمل ارتباط صلب له مع مؤشر الفهرسة inode للكائن القديم (وبالتالي يُصبحان بنفس رقم inode)، وأيضاً سوف يزداد رقم عداد الارتباطات link count number.

تسمح الروابط الصلبة hard link بوجود أسماء متعددة multiple names للإشارة refer إلى نفس الملف same file. حتى إذا تمت إزالة removed أحد تلك الأسماء؛ فإن الأسماء الأخرى يمكن استخدامها للإشارة إلى ذلك الملف.

في الواقع، يمكن حتى استخدام تلك الأسماء الأخرى other names لإنشاء ارتباطات إضافية additional links. وتعتبر كل تلك الأسماء متكافئة equivalent بما أنها جميعاً تشير إلى مؤشر فهرسي موجود existing inode.

ملاحظة: لا يمكنك إنشاء ارتباطات صلبة hard links بأي من الدلائل ولكن فقط بالملفات. أيضا الارتباط الصلب بملف يجب أن يكون موجوداً داخل نفس نظام الملفات (التقسيم partition) بأنه مرتبط إليه.

٤, ٣, ١٦ الخطوة ٤

استخدم الأمر ln لإنشاء create ارتباط صلب hard link. ثم اعرض تفاصيل details ومعلومات inode للملف source وملف الارتباط الصلب الجديد:

ln source hardlink

ls -li source hardlink

لاحظ أن الملف الجديد hardlink والملف الأصلي source يتشاركان نفس inode:

```
sysadmin@localhost:~$ ln source hardlink
sysadmin@localhost:~$ ls -li source hardlink
2076 -rw-rw-r-- 2 sysadmin sysadmin 5 Apr 12 13:27 hardlink
2076 -rw-rw-r-- 2 sysadmin sysadmin 5 Apr 12 13:27 source
sysadmin@localhost:~$
```

تنفيذ الأمرين ln source hardlink ، ls -li source hardlink

٥،٣،١٦ الخطوة ٥

يستخدم الأمر ln لإنشاء ارتباط صلب آخر مع الملف source، ثم اعرض تفاصيل وأيضاً معلومات مؤشر الأرشفة inode للملفات التي تم عمل ارتباط لها:

ln hardlink hardtoo

ls -li hardlink hardtoo source

لاحظ في التنفيذ أن الروابط الصلبة hardlinks تتشارك نفس inode، وعداد الإرتباطات link count يزداد لجميع الملفات المرتبطة عند إضافة الإرتباط الجديد:

```
sysadmin@localhost:~$ ln source hardtoo
sysadmin@localhost:~$ ls -li hardlink hardtoo source
2076 -rw-rw-r-- 3 sysadmin sysadmin 5 Apr 12 13:27 hardlink
2076 -rw-rw-r-- 3 sysadmin sysadmin 5 Apr 12 13:27 hardtoo
2076 -rw-rw-r-- 3 sysadmin sysadmin 5 Apr 12 13:27 source
sysadmin@localhost:~$
```

تنفيذ الأمرين `ln hardlink hardtoo` و `ls -li hardlink hardtoo source`

٦،٣،٦ الخطوة ٦

أزل `remove` الارتباط الأخير التي تم إنشاؤه، ثم قم بسرد تفاصيل الملفين `source` و `hardlink` :

```
rm hardtoo
```

```
ls -li source hardlink
```

لاحظ أن عدد الارتباطات `link count` تناقص `decreases` عند إزالة `removed` أحد الملفات المرتبطة

الصلبة `hard linked file`:

```
sysadmin@localhost:~$ rm hardtoo
sysadmin@localhost:~$ ls -li source hardlink
2076 -rw-rw-r-- 2 sysadmin sysadmin 5 Apr 12 13:27 hardlink
2076 -rw-rw-r-- 2 sysadmin sysadmin 5 Apr 12 13:27 source
sysadmin@localhost:~$
```

تنفيذ الأمرين `rm hardtoo` و `ls -li source hardlink`

٧،٣،٦ الخطوة ٧

أزل ملف الارتباط الصلب `hardlink`، ثم قم بسرد تفاصيل الملف `source`:

rm hardlink

ls -li source

وينبغي ان يكون التنفيذ كالتالي:

```
sysadmin@localhost:~$ rm hardlink
sysadmin@localhost:~$ ls -li source
2076 -rw-rw-r-- 1 sysadmin sysadmin 5 Apr 12 13:27 source
sysadmin@localhost:~$
```

تنفيذ الأمرين rm hardlink ،ls -li source

الآن ننتقل الى النوع الآخر من الارتباطات التي يمكن إنشاؤها وهو ما يعرف باسم الارتباط الرمزي symbolic link أو اللين soft link. وفي هذا النوع من الروابط لا يزيد عدد الارتباطات link count للملفات المرتبطة linked files.

ملفات الارتباط الرمزية symbolic link files لها مؤشر أرشفة inode ونوع ملف file type مستقل وخاص بها. فبدلاً من ربط ومشاركة inode فإنها ترتبط مع اسم الملف file name. فعلى عكس الارتباطات الصلبة hard links؛ يمكن للروابط اللينة soft links أن ترتبط بالدلائل directories ويمكنها أن تعبر cross الأجهزة والأقسام devices and partitions لتصل إلى وجهتها.

٨,٣,١٦ الخطوة ٨

انشئ ارتباط رمزي symbolic link مع الملف source، ثم قم بعرض التفاصيل لكلا الملفين المرتبطين:

ln -s source softlink

ls -li source softlink

يُبين التنفيذ المظلل باللون الأصفر أن الملف `softlink` والملف `source` لديهما `inodes` مختلف. لاحظ انه يتم إنشاء نوع ارتباط `link type` للملف عند عمل ارتباط رمزي `symbolic link`. أما أذونات الارتباط `link permissions` فهي ليست ذات صلة `irrelevant`؛ حيث أن أذونات الملف الهدف `target file` هي التي تحدد إمكانية الوصول `access`:

```
sysadmin@localhost:~$ ln -s source softlink
sysadmin@localhost:~$ ls -li source softlink
2086 lrwxrwxrwx 1 sysadmin sysadmin 6 Apr 12 13:56 softlink -> source
2076 -rw-rw-r-- 1 sysadmin sysadmin 5 Apr 12 13:27 source
sysadmin@localhost:~$
```

تنفيذ الأمرين `ln -s source softlink` ، `ls -li source softlink`

٩،٣،١٦ الخطوة ٩

انشئ ارتباط رمزي `symbolic link` مع الدليل `/proc` ، ثم قم بعرض تفاصيل الرابط:

```
ln -s /proc crossdir
```

```
ls -l crossdir
```

نجاح هذه الأوامر دليل على أن الروابط اللينة `soft links` يمكن أن تشير إلى الدلائل `directories`، وأنها يمكن تعبير `cross` من ملف نظام `filesystem` إلى آخر، وهما المهمتان التي لا تستطيع الروابط الصلبة `hard links` أن تقوم بها:

```
sysadmin@localhost:~$ ln -s /proc crossdir
sysadmin@localhost:~$ ls -l crossdir
lrwxrwxrwx 1 sysadmin sysadmin 5 Apr 12 14:00 crossdir -> /proc
sysadmin@localhost:~$
```

تنفيذ `ln -s /proc crossdir` ، `ls -l crossdir`

## الملاحق

## الملحق رقم (١)

ملخص لأوامر المعمل الرابع:

Command -option string	المعنى
<b>whoami</b>	يعرض اسم المستخدم الحالي للنظام.
<b>uname</b>	عرض اسم نواة نظام التشغيل.
<b>uname -n</b>	<b>uname</b> عرض اسم نواة نظام التشغيل، أما مع الخيار <b>-n</b> مع فسيتم عرض اسم مضيف الشبكة.
<b>name --nodename</b>	<b>uname</b> عرض اسم نواة نظام التشغيل، أما مع الخيار <b>--nodename</b> فهو يشبه <b>-n</b> وذلك لعرض اسم مضيف الشبكة.
<b>pwd</b>	عرض المسار للموقع الحالي.
<b>echo Hello Student</b>	<b>echo</b> يقوم بالإخراج على الشاشة، لجملة Hello Student.
<b>echo \$PATH</b>	<b>echo</b> يقوم بالإخراج على الشاشة، وعلامة \$ تدل على أنه متغير، أخيراً اسم المتغير ( PATH ).
<b>which date</b>	<b>which</b> يحدد موقع ملف البرنامج في مسار المستخدم، وهنا مثال على استخدامه مع الأمر تاريخ <b>date</b> .
<b>echo *</b>	<b>echo</b> يقوم بالإخراج على الشاشة، وعلامة النجمة * لعرض كافة أسماء الملفات والدلائل في الدليل الحالي.
<b>echo D*</b>	<b>echo</b> يقوم بالإخراج على الشاشة، و D يعني لكافة الملفات والدلائل في الدليل الحالي التي تبدأ بحرف D، وتقوم علامة * بإكمال الاسم، حيث يمكن أن تكون قيمة النجمة * صفر أو أكثر.

<b>echo P*</b>	<b>echo</b> يقوم بالإخراج على الشاشة، و P يعني لكافة الملفات والدلائل في الدليل الحالي التي تبدأ بحرف P، وتقوم علامة * بإكمال الاسم، حيث يمكن أن تكون قيمة النجمة * صفر أو أكثر.
<b>echo *s</b>	<b>echo</b> يقوم بالإخراج على الشاشة، وعلامة * لإكمال الاسم، و s تعني عرض كافة الملفات والدلائل في الدليل الحالي التي تنتهي بحرف s، حيث يمكن أن تكون قيمة * صفر أو أكثر.
<b>echo D*n*s</b>	<b>echo</b> يقوم بالإخراج على الشاشة، و D يعني لكافة الملفات والدلائل في الدليل الحالي التي تبدأ بحرف D ويتوسطها حرف n وتنتهي بحرف s، وتقوم علامة النجمة * بإكمال اسم الملف.
<b>echo ???????</b>	<b>echo</b> يقوم بالإخراج على الشاشة، و ??????? تعني عرض الملفات والدلائل في الدليل الحالي التي تتألف من ستة أحرف، وبعبارة أخرى يكون طول الاسم بعدد الاستفهامات.
<b>echo D????????</b>	<b>echo</b> يقوم بالإخراج على الشاشة، و D???????? يعني لكافة الملفات والدلائل في الدليل الحالي التي تتألف من حرف D في البداية ثم ثمانية خانات، وبعبارة أخرى يكون طول الاسم بعدد الاستفهامات وحرف D وبالتالي تسعة أحرف.
<b>echo ?????*s</b>	<b>echo</b> يقوم بالإخراج على الشاشة، و ?????*s تعني عرض أسماء الملفات والدلائل في الدليل الحالي المكونة من ستة أحرف أو أكثر (خمسة استفهامات و حرف s و *) وتنتهي بحرف s. لاحظ أن النجمة * تعني صفر أو أكثر وبالتالي فإن طول الكلمة لا يمكن أن يقل عن ستة أحرف، فهو ستة أو أكثر.
<b>echo [DP]*</b>	<b>echo</b> يقوم بالإخراج على الشاشة، و الأقواس المربعة [ ] فهي لتحديد مجموعة؛ وبالتالي يتم عرض الملفات والدلائل في الدليل

	<p>الحالي التي تبدأ بأحد الحرفين D و P، وتقوم علامة النجمة * بإكمال الاسم.</p>
<b>echo [!DP]*</b>	<p><b>echo</b> يقوم بالإخراج على الشاشة، و الأقواس المربعة [ ] فهي لتحديد مجموعة؛ أما علامة التعجب! فهي للنفي؛ وبالتالي سيتم عرض الملفات والدلائل في الدليل الحالي التي لا تبدأ بأحد الحرفين D و P، حيث ستظهر جميع الملفات والدلائل ما عدا التي تبدأ بالحرفين السابقين، وتقوم علامة النجمة * بإكمال الاسم.</p>
<b>echo [D-P]*</b>	<p><b>echo</b> يقوم بالإخراج على الشاشة، و الأقواس المربعة [ ] فهي لتحديد مجموعة؛ أما علامة الشرطة - فهي لتحديد مدى معين؛ وبالتالي سيتم عرض الملفات والدلائل في الدليل الحالي بدايةً من الحرف D ثم E ثم ..... إلا أن نصل للحرف P، وتقوم علامة النجمة * بإكمال الاسم.</p>
<b>echo [!D-P]*</b>	<p><b>echo</b> يقوم بالإخراج على الشاشة، و الأقواس المربعة [ ] فهي لتحديد مجموعة؛ و علامة الشرطة - فهي لتحديد مدى معين؛ أما علامة التعجب! فهي للنفي؛ وبالتالي سيتم إظهار جميع الملفات والدلائل في الدليل الحالي التي لا تكون احرفها واقعة بين المدى D و P، وأخيراً تقوم علامة النجمة * بإكمال الاسم.</p>
<b>echo Today is `date`</b>	<p><b>echo</b> يقوم بالإخراج على الشاشة، للجملة Today is، أما علامة الاقتباس الخلفية فتدل على ان ما بداخلها هو امر وليس نص عادي، وبالتالي عند التنفيذ يتم اظهار الجملة ثم التاريخ.</p>
<b>echo Today is \$(date)</b>	<p><b>echo</b> يقوم بالإخراج على الشاشة، للجملة Today is، أما علامة الدولار \$ فتدل على ان ما يليها بين القوسين هو امر وليس نص عادي، وبالتالي سيتم اظهار الجملة ثم التاريخ.</p>

<p><b>echo</b> This is the command `date`</p>	<p><b>echo</b> يقوم بالإخراج على الشاشة، للجملة This is the command `date`، أما علامة الاقتباس المفردة فتقوم بمنع تنفيذ الأمر، وبالتالي يتم اخراج الجملة كما هي، فيظهر عند تنفيذ الأمر نفس الجملة وهي: `date` This is the command .</p>
<p><b>echo</b> This is the command \date\</p>	<p><b>echo</b> يقوم بالإخراج على الشاشة، للجملة This is the command \date\، أما الخط المائل مع علامات الاقتباس فيقومان بمنع تنفيذ الأمر، وبالتالي يتم اخراج الجملة كما هي، فيظهر عند التنفيذ نفس الجملة: \date\ This is the command .</p>
<p><b>echo</b> This is the command "date"</p>	<p><b>echo</b> يقوم بالإخراج على الشاشة، للجملة This is the command "date"، أما علامة الاقتباس المزدوجة فليس لها تأثير على علامة الاقتباس الخلفية. وبالتالي عند التنفيذ يتم اظهار الجملة ثم التاريخ.</p>
<p><b>echo "D*"</b></p>	<p><b>echo</b> يقوم بالإخراج على الشاشة، أما علامة الاقتباس المزدوجة فلها تأثير على رموز glob، حيث انها تعطل المعنى خاص بهم، وبالتالي يظهر عند التنفيذ التالي: D* .</p>
<p><b>echo</b> Hello; <b>echo</b> Linux; <b>echo</b> Student</p>	<p><b>echo</b> يقوم بالإخراج على الشاشة، للكلمة Hello، أما علامة الفاصلة المنقوطة فتقوم بالفصل بين الأوامر، حيث سيتم تنفيذ أمر الإخراج الثاني لكلمة Linux في سطر جديد، وبالمثل لأمر الإخراج الأخير حيث سيتم عرض كلمة Student في سطر ثالث، وبالتالي تنفيذ كل أمر في سطر منفصل.</p>
<p><b>false; echo</b> Not; <b>echo</b> Conditional</p>	<p>أمر <b>false</b> يُعطي نتيجة خطأ، ثم تأتي الفاصلة المنقوطة للفصل بين الأوامر، أما <b>echo</b> فيقوم بالإخراج على الشاشة للكلمة Not، ونفس الحال في الأمر <b>echo</b> الأخير فيقوم بالإخراج على الشاشة للكلمة Conditional في سطر جديد.</p>

<b>echo Start &amp;&amp; echo Going &amp;&amp; echo Gone</b>	<b>echo</b> يقوم بالإخراج على الشاشة، للكلمة <b>Start</b> أما علامة الشرط <b>او &amp;&amp;</b> فتحتاج ان تكون الجملتان صحيحة لكي يتم تنفيذها. وبما أنها صحيحة فسيتم إخراج الكلمة الثانية <b>Going</b> ، وأخيراً سيتم إخراج الكلمة الثالثة <b>Gone</b> ، ونتيجة المثال هي إخراج الكلمات <b>Start</b> و <b>Going</b> و <b>Gone</b> في سطور مستقلة.
<b>echo Success &amp;&amp; false &amp;&amp; echo Bye</b>	<b>echo</b> يقوم بالإخراج على الشاشة، للكلمة <b>Success</b> ، ثم علامة <b>او الشرطية &amp;&amp;</b> ثم امر <b>false</b> والذي يُعطي نتيجة خطأ، وبالتالي سيتم إخراج كلمة <b>Success</b> فقط.
<b>false    echo Fail Or</b>	<b>false</b> يُعطي نتيجة خطأ، ثم علامة جملة <b>أو   </b> والتي تحتاج أن يكون احد الجملتين صحيحاً لكي تتحقق، وبالتالي سيتم إخراج <b>Fail Or</b> عن طريق الامر <b>echo</b> لأنه صحيح.
<b>true    echo Nothing to see here</b>	<b>true</b> تُعطي نتيجة صحيحة، وتستخدم لعدم عمل أي شيء في الجملة ؛ فقط التوقف، وبالتالي فإنها تُسبب عدم التنفيذ للجملة.
<b>date</b>	إظهار الوقت والتاريخ.
<b>clear</b>	يزيل الأوامر من الشاشة الطرفية، لكنها تبقى في السجل.
<b>history</b>	لعرض سجل بالأوامر المدخلة مع الترقيم.
<b>history 5</b>	<b>history</b> لعرض سجل بالأوامر المدخلة مع الترقيم، والرقم 5 يعني عرض آخر خمسة أوامر قمت بكتابتها.
<b>!94</b>	<b>!</b> لتنفيذ أمر مرة أخرى بمعلومية رقم ذلك الأمر، ورقم 94 هو رقم الأمر المراد تنفيذه، ويتم معرفة أرقام الأوامر عن طريق الأمر <b>history</b> .

## الملحق رقم (٢)

ملخص لأوامر المعمل الخامس:

المعنى	Command -option string
لإظهار الوقت والتاريخ.	<b>date</b>
<b>man</b> لمعرفة المزيد عن أمر ما عن طريق صفحات دليل المستخدم. وفي المثال نريد عرض المزيد عن الأمر <b>date</b> .	<b>man date</b>
للخروج من صفحات دليل المستخدم.	<b>q</b>
علامة / هي للبحث داخل صفحة دليل المستخدم، عن الكلمة <b>file</b>	<b>/file</b>
<b>man</b> للبحث في صفحات دليل المستخدم، و <b>-k</b> يعني البحث بالكلمة الرئيسية <b>password</b> وبالتالي اظهر النتائج التي تحتوي على الكلمة السابقة في أي مكان بالمستند (العنوان و المحتوى) وبدون الدخول إلى الصفحات، فقط في سطر الأوامر. ملاحظة: الأوامر <b>man -k</b> و <b>apropos</b> يؤديان نفس الوظيفة.	<b>man -k password</b>
<b>apropos</b> أمر آخر للبحث في صفحات دليل المستخدم، بالكلمة الرئيسية <b>password</b> . وبالتالي اظهر النتائج التي تحتوي على الكلمة السابقة في أي مكان بالمستند (العنوان و المحتوى) وبدون الدخول إلى الصفحات، فقط في سطر الأوامر. ملاحظة: الأوامر <b>apropos</b> و <b>man -k</b> يؤديان نفس الوظيفة.	<b>apropos password</b>
<b>man</b> يقوم بالبحث في صفحات دليل المستخدم، و <b>-f</b> تعني البحث بالاسم <b>passwd</b> بدون الدخول إلى الصفحات، وبالتالي اظهر النتائج التي تحتوي على الاسم كامل في عنوان المستند فقط. بدون استعراض للمحتويات، فقط في سطر الأوامر.	<b>man -f passwd</b>

	ملاحظة: الأمران <b>man -f</b> و <b>whatis</b> يقومان بنفس العمل.
<b>man 5 passwd</b>	<b>man</b> للبحث في صفحات دليل المستخدم، ورقم 5 يعني البحث داخل القسم رقم 5 ، وأخيراً الكلمة التي نريد البحث عنها <b>passwd</b> ، واستعراض تلك المحتويات.
<b>whatis passwd</b>	<b>whatis</b> أمر آخر للبحث في صفحات دليل المستخدم بدون الدخول إلى الصفحات، وبالتالي يتم اظهار النتائج التي تحتوي على الاسم في عنوان المستند فقط وهو <b>passwd</b> بدون استعراض للمحتويات، فقط في سطر الأوامر. ملاحظة: الأمران <b>whatis</b> و <b>man -f</b> يقومان بنفس العمل.
<b>info date</b>	<b>info</b> صفحات المعلومات هي لإعطاء معلومات تفصيلية عن بعض الأوامر، مثل الأمر <b>date</b> .
<b>date --help</b>	<b>--help</b> عرض كيفية استخدام الأوامر الأساسية للأمر، <b>date</b> .
<b>ls /usr/share/doc</b>	<b>ls</b> عرض قائمة بالملفات والدلائل، و <b>-l</b> لإعطاء التفاصيل، للكائن <b>doc</b> ، والموجود بالمسار المطلق <b>/usr/share/doc</b> ، لاحظ أن الاستعراض من المكان الحالي ولا نحتاج الى الانتقال إلى ذلك المسار.
<b>locate crontab</b>	<b>locate</b> لتحديد موقع ملف، باسم <b>crontab</b> .
<b>locate -b "\crontab"</b>	<b>locate</b> لتحديد موقع ملف، و <b>-b</b> لإظهار الملفات التي بنفس الاسم وليس الامتداد، وأخيراً اسم الملف <b>crontab</b> .
<b>whereis passwd</b>	<b>whereis</b> لمعرفة موقع أمر ما. وهو هنا الأمر <b>passwd</b> .

## الملحق رقم (٣)

ملخص لأوامر المعمل السادس:

Command -option string	المعنى
<b>pwd</b>	عرض المسار للموقع الحالي.
<b>echo \$HOME</b>	<b>echo</b> يقوم بالإخراج على الشاشة، وعلامة \$ تدل على أنه متغير، أخيراً اسم المتغير (HOME).
<b>cd /</b>	<b>cd</b> أمر انتقال الى مسار ما، وهو الدليل الجذر / .
<b>cd</b>	<b>cd</b> أمر انتقال الى مسار ما، وافترضياً إذا لم يُكتب شيء بعد الأمر؛ يتم الانتقال إلى الدليل الرئيسي للمستخدم الحالي، ورمزه ~.
<b>cd ~</b>	<b>cd</b> أمر انتقال الى مسار ما، و علامة تيلدا ~ هي الدليل الرئيسي، وإذا لم يُكتب اسم المستخدم، فسيتم الانتقال الى الدليل الرئيسي للمستخدم الحالي.
<b>cd ~sysadmin</b>	<b>cd</b> أمر انتقال الى مسار ما، و علامة تيلدا ~ هي الدليل الرئيسي، وأخيراً اسم المستخدم <b>sysadmin</b> .
<b>cd /home/sysadmin</b>	<b>cd</b> أمر انتقال الى مسار ما، و <b>home</b> هو الدليل الرئيسي، وذلك للمستخدم <b>sysadmin</b> . وهو نفس الأمر <b>cd ~ sysadmin</b> .
<b>cd /home</b>	<b>cd</b> أمر انتقال الى مسار ما، وهو الدليل <b>/home</b> .
<b>echo ~ ~sysadmin ~root ~mail ~nobody</b>	<b>echo</b> يقوم بالإخراج على الشاشة، وذلك للمسارات التالية: ~ <b>~mail ~nobody ~root ~sysadmin</b> ، واحداً تلو الآخر.
<b>cd ~root</b>	<b>cd</b> أمر انتقال الى مسار ما، و علامة تيلدا ~ هي الدليل الرئيسي، ثم اسم المستخدم <b>root</b> . (سيتم رفض الوصول).

<b>cd /home/root</b>	<b>cd</b> أمر انتقال الى مسار ما، و <b>home</b> هو الدليل الرئيسي، وذلك للمستخدم <b>.root</b> .
<b>cd /usr/bin</b>	<b>cd</b> أمر انتقال الى مسار ما، وهو الكائن (ملف أو دليل) <b>bin</b> الموجود داخل الدليل <b>usr</b> بالمسار المطلق. <b>المسار المطلق:</b> المسار الكامل لموقع الكائن من البداية الى النهاية. <b>المسار النسبي:</b> مسار كائن داخلي، ولا يُكتب في بدايته الرمز <b>/</b> .
<b>cd ..</b>	<b>cd</b> أمر انتقال الى مسار ما، و <b>..</b> تعني العودة خطوة واحدة للخلف.
<b>cd ../..</b>	<b>cd</b> أمر انتقال الى مسار ما، و <b>../..</b> تعني العودة خطوتين للخلف.
<b>cd ../var</b>	<b>cd</b> أمر انتقال الى مسار ما، و <b>..</b> تعني العودة خطوة للخلف، و <b>/var</b> تعني الانتقال إلى الدليل <b>var</b> .
<b>ls</b>	<b>ls</b> عرض قائمة بالمحتويات الموجودة (ملفات ودلائل وملفات مضغوطة و...)، وإذا لم يتم تحديد أي موقع بعد الأمر فسيتم عرض المحتويات الموجودة في المسار الحالي.
<b>ls -a</b>	<b>ls</b> عرض قائمة بالملفات والدلائل، و <b>-a</b> لإظهار الملفات أو الدلائل المخفية، وحيث أنه لم يتم تحديد أي موقع فسيتم عرض المحتويات الموجودة بالمسار الحالي.
<b>ls -l /etc/hosts</b>	<b>ls</b> عرض قائمة بالملفات والدلائل، و <b>-l</b> لإعطاء التفاصيل، للكائن (ملف أو دليل) <b>hosts</b> ، لاحظ أن الاستعراض من المكان الحالي ولا نحتاج الى الانتقال إلى ذلك الكائن.
<b>ls -lh</b>	<b>ls</b> عرض قائمة بالملفات والدلائل، و <b>-l</b> لإعطاء تفاصيل كاملة، و <b>h</b> لعرضها بلغة مقروءة للإنسان. وحيث أنه لم يتم تحديد أي موقع بعد الأمر فسيتم عرض المحتويات الموجودة في المسار الحالي.

<b>ls -R /etc/udev</b>	<b>ls</b> عرض قائمة الملفات والدلائل، و <b>-R</b> لعرض أيضاً ما يتفرع من الكائن، وأخيراً اسم الكائن <b>udev</b> ، وذلك بدون الانتقال إليه.
<b>ls -d /etc/s*</b>	<b>ls</b> عرض قائمة بالملفات أو الدلائل، و <b>-d</b> تمنع عرض الدلائل الفرعية، وذلك للملفات التي تبدأ بحرف <b>s</b> ، والنجمة * لإكمال الاسم.
<b>ls -d /etc/????</b>	<b>ls</b> عرض قائمة بالملفات أو الدلائل، و <b>-d</b> تمنع عرض الدلائل الفرعية، للملفات التي تتكون من أربع خانات من الأحرف، لاحظ أن العمل من المكان الحالي.
<b>ls -d /etc/[abcd]*</b>	<b>ls</b> عرض قائمة بالملفات أو الدلائل، و <b>-d</b> تمنع عرض الدلائل الفرعية، للملفات التي تبدأ بأحد الأحرف <b>[abcd]</b> ، لاحظ أن العمل من المكان الحالي.
<b>cp /etc/hosts hosts</b>	<b>cp</b> أمر نسخ، للملف <b>hosts</b> من المسار المطلق ( <b>/etc/hosts</b> )، إلى الموقع الحالي، وذلك باسم <b>hosts</b> (مسار نسبي).
<b>cp -v /etc/hosts hosts</b>	<b>cp</b> أمر نسخ، <b>-v</b> لعرض الحدث، ويكون النسخ للملف <b>hosts</b> الواضح في المسار المطلق ( <b>/etc/hosts</b> )، باسم <b>hosts</b> إلى الموقع الحالي (مسار نسبي)، لاحظ أن العمل تم من المكان الحالي.
<b>cp -v /etc/hosts h</b>	<b>cp</b> أمر نسخ، و <b>-v</b> لعرض الحدث، للملف <b>hosts</b> من المسار المطلق ( <b>/etc/hosts</b> ) إلى الموقع الحالي الذي أعمل منه (استخدم <b>pwd</b> لمعرفة الموقع الحالي) والنسخة تكون باسم <b>h</b> ، لاحظ أن العمل تم من المكان الحالي.
<b>cp -v /etc/hosts /etc/f</b>	<b>cp</b> أمر نسخ، <b>-v</b> لعرض الحدث، ويكون النسخ للملف <b>hosts</b> الواضح في المسار المطلق ( <b>/etc/hosts</b> ) باسم <b>f</b> ، وسيتم نسخه إلى الموقع الجديد والذي يتضمن اسم الملف <b>/etc/f</b> . لاحظ أن العمل تم من المكان الحالي.
<b>rm hosts</b>	حذف الملف <b>hosts</b> بالمسار النسبي، لأننا لم نستخدم رمز /

<b>cp -v /etc/hosts .</b>	<b>cp</b> أمر نسخ، و <b>-v</b> لعرض الحدث، للملف <b>hosts</b> من المسار المطلق ( <b>/etc/hosts</b> ) والنقطة تعني النسخ إلى الموقع الحالي الملف بنفس اسم الملف الأصلي.
<b>cp -p hosts /home/sysadmin</b>	<b>cp</b> أمر نسخ و <b>-p</b> تعني النسخ مع الاحتفاظ بالتفاصيل الاصلية للملف، ثم اسم الملف المراد نسخه وهو <b>hosts</b> (بالمسار النسبي)، وأخيراً تحديد موقع النسخة الجديدة بالمسار المطلق ( <b>/home/sysadmin</b> ). وحيث انه لم يتم تحديد اسم جديد فسيتم النسخ بنفس الاسم الأصلي.
<b>rm hosts newname</b>	<b>rm</b> أمر حذف، للملفين <b>hosts</b> و <b>newname</b> بالمسار النسبي.
<b>mkdir Myetc</b>	<b>mkdir</b> لإنشاء دليل، باسم <b>Myetc</b> بالموقع الحالي (مسار نسبي).
<b>cp -R /etc/udev Myetc</b>	<b>cp</b> أمر نسخ، <b>-R</b> يعني نسخ جميع الدلائل الفرعية أيضاً، وذلك للدليل <b>udev</b> بالمسار المطلق <b>/etc/udev</b> وتسميته باسم <b>Myetc</b> (مسار نسبي)، وبما ان الدليل <b>Myetc</b> قد سبق انشاءه فسيقوم النظام بتحديث الدليل ونسخ البيانات اليه، النسخة ستكون في المكان الحالي.
<b>ls -lR Myetc</b>	<b>ls</b> عرض قائمة بالملفات أو الدلائل، و <b>-l</b> بالتفاصيل، <b>R</b> تعني عرض الدلائل الفرعية أيضاً للدليل <b>Myetc</b> (مسار نسبي).
<b>rm -r Myetc</b>	<b>rm</b> أمر حذف، <b>-r</b> لحذف الدلائل (بينما لحذف الملفات فقط <b>rm</b> ).
<b>rmdir</b>	<b>rmdir</b> لحذف الدلائل الفارغة فقط، ثم اسم الدليل.
<b>touch premove</b>	أمر <b>touch</b> له وظيفتين، الأولى تحديث الملف الموجود بهذا الاسم وهو في المثال <b>premove</b> ، اما اذا كان الملف غير موجود فسيقوم الأمر بإنشائه بالاسم السابق في المكان الحالي (مسار نسبي).
<b>mv premove postmove</b>	<b>mv</b> أمر قص ولصق، للملف <b>premove</b> ولصقه باسم <b>postmove</b> .

## الملحق رقم (٤)

### ملخص لأوامر المعمل السابع:

Command -option string	المعنى
<b>tar -cvf</b> mybackups/udev.tar /etc/udev	<b>tar</b> امر ارسفة، <b>-c</b> انشاء الملف، <b>v</b> لإظهار العملية، <b>f</b> لتحديد الاسم، والاسم هو <b>udev.tar</b> ويُوضع في الدليل <b>mybackups</b> (مسار نسبي)، وأخيراً اسم الدليل الذي نريد ارسفته وهو <b>udev</b> والواضح في المسار المطلق <b>/etc/udev</b> . لاحظ أن الاسم يجب أن يحتوي على نوع الامتداد وهو <b>.tar</b> .
<b>tar -tvf</b> mybackups/udev.tar	<b>tar</b> امر ارسفة، <b>-t</b> عرض محتويات الملف المؤرشف، <b>v</b> لإظهار العملية، <b>f</b> لتحديد الاسم، وهنا اسم الملف <b>udev.tar</b> الموجود في الدليل <b>mybackups</b> (مسار نسبي).
<b>tar -zcvf</b> mybackups/udev.tar.gz /etc/udev	<b>tar</b> امر ارسفة، <b>-z</b> لضغط الملف، <b>c</b> لإنشاء الملف، <b>v</b> لإظهار العملية، <b>f</b> لتحديد الاسم، ويكون اسم الملف <b>udev.tar.gz</b> في الدليل <b>mybackups</b> (مسار نسبي)، وأخيراً اسم الدليل الذي نريد ارسفته وهو <b>udev</b> والواضح في المسار المطلق <b>/etc/udev</b> . لاحظ أن الاسم يجب أن يحتوي على نوع الامتداد وبما أن الملف مؤرشف ومضغوط فيجب أن يكون <b>.tar.gz</b> .
<b>tar -xvf</b> udev.tar.gz	<b>tar</b> امر ارسفة، <b>-x</b> استخراج محتويات ملف مضغوط، <b>v</b> لإظهار العملية، <b>f</b> لتحديد الاسم. وهو <b>udev.tar.gz</b> (مسار نسبي).
<b>tar -rvf</b> udev.tar /etc/hosts	<b>tar</b> امر ارسفة، <b>-r</b> إضافة ملف جديد إلى ملف المؤرشف، <b>v</b> لإظهار العملية، <b>f</b> لتحديد الاسم. واسم الملف المؤرشف السابق

	فهو <b>udev.tar</b> (مسار نسبي)، أما الملف الجديد الذي نريد إضافته فهو <b>hosts</b> والموجود في الدليل <b>/etc</b> كما في المسار المطلق <b>/etc/hosts</b> .
<b>gzip words</b>	<b>gzip</b> امر ضغط، للملف <b>words</b> (مسار نسبي). لاحظ انه لا يجب عليك كتابة نوع الامتداد هنا كما مع الامر <b>tar</b>
<b>gunzip words.gz</b>	<b>gunzip</b> امر لفك الضغط، عن الملف <b>words.gz</b>
<b>bzip2 words</b>	<b>bzip2</b> أمر آخر للضغط، وذلك الملف <b>words</b> (مسار نسبي).
<b>bunzip2 words.bz2</b>	<b>bunzip2</b> امر لفك الضغط، عن الملف <b>words.bz2</b>
<b>zip words.zip words</b>	<b>zip</b> امر آخر للضغط، ثم اسم الملف المضغوط الجديد <b>words.zip</b> ، واخيراً الملف الذي تريد ضغطه <b>words</b> (مسار نسبي).
<b>zip w words</b>	<b>zip</b> امر آخر للضغط، ثم اسم الملف المضغوط الجديد <b>w</b> ، واخيراً الملف الذي تريد ضغطه <b>words</b> (مسار نسبي). لاحظ أنه لا يتوجب عليك وضع الامتداد هنا وهو <b>zip</b> لأن الأمر يقوم بذلك ولكن يتم وضعه عادة من المستخدم للتفريق.
<b>zip -r udev.zip /etc/udev</b>	<b>zip</b> امر ضغط، <b>-r</b> يستخدم لضغط الملفات الفرعية. ثم اسم الملف الجديد <b>udev.zip</b> (مسار نسبي)، وأخيراً الملف الذي نريد ضغطه وهو <b>udev</b> والواضح بالمسار المطلق <b>/etc/udev</b>
<b>unzip -l udev.zip</b>	<b>unzip</b> أمر لفك الضغط، الخيار <b>-l</b> لعرض محتويات الملف المضغوط <b>udev.zip</b> (مسار نسبي).
<b>unzip udev.zip</b>	<b>unzip</b> لفك الضغط، عن الملف المضغوط <b>udev.zip</b> (مسار نسبي).

## الملحق رقم (٥)

### ملخص لأوامر المعمل الثامن:

Command -option string	المعنى
<code>echo "Hello World" &gt; mymessage</code>	<b>echo</b> يقوم بإخراج على الشاشة، للجملة "Hello World"، اما علامة التوجيه > فتعني إعادة توجيه الجملة الى الملف mymessage.
<code>cat mymessage</code>	<b>cat</b> يعرض المحتويات، للملف mymessage.
<code>echo "Hello World" &gt;&gt; mymessage</code>	<b>echo</b> يقوم بإخراج الجملة "Hello World" على الشاشة، اما علامة التوجيه >> فتعني إعادة توجيه الجملة الى الملف mymessage دون مسح ما يوجد سابقاً بداخل الملف.
<code>find /etc -name hosts</code>	<b>find</b> أمر بحث، في الموقع /etc/ بمعيار الاسم عن hosts
<code>find /etc -name hosts 2&gt; err.txt</code>	<b>find</b> أمر بحث، في الموقع /etc/ بمعيار الاسم عن hosts مع توجيه رسائل الخطأ (القناة رقم ٢) إلى الملف err.txt .
<code>find /etc -name hosts &gt; std.out 2&gt; std.err</code>	<b>find</b> أمر بحث، في الموقع /etc/ بمعيار الاسم عن hosts مع توجيه الإخراج القياسي (قناة رقم ١) الى std.out وتوجيه رسائل الخطأ (القناة رقم ٢) إلى الملف std.err .
<code>find /etc -name hosts &gt;find.out 2&gt;&amp;1</code>	<b>find</b> أمر بحث، في الموقع /etc/ بمعيار الاسم عن hosts مع توجيه الإخراج القياسي (قناة رقم ١)، وأيضاً أعد توجيه رسائل الخطأ (القناة رقم ٢) مع الإخراج القياسي (قناة رقم ١) والذي سيذهب إلى الملف find.out . وبالتالي فإن الإخراج القياسي ورسائل الخطأ ستتم إعادة توجيههم الى نفس المكان وهو الملف find.out

<p><b>tr a-z A-Z</b></p>	<p><b>tr</b> تحويل النص من حروف صغيرة a-z إلى كبيرة A-Z، لاحظ أن المستخدم هو الذي يقوم بكتابة النص المراد تحويله. وللخروج بعد كتابة النص اضغط على <b>Ctrl + d</b> .</p>
<p><b>tr A-Z a-z &gt; myfile</b></p>	<p><b>tr</b> تحويل النص من حروف كبيرة A-Z إلى صغيرة a-z وتوجيه الإخراج القياسي (stdout) إلى الملف <b>myfile &gt;</b> ، وللخروج بعد كتابة النص اضغط على <b>Ctrl + d</b> .</p>
<p><b>tr a-z A-Z &lt; myfile</b></p>	<p><b>tr</b> تحويل النص من حروف صغيرة a-z إلى كبيرة A-Z وعلامة <b>&lt;</b> تعني اخذ البيانات الموجودة في الملف <b>myfile</b> وعمل التحويل عليها. إذن النص لا يتم إدخاله من قبل المستخدم.</p>
<p><b>ls -l /etc   more</b></p>	<p><b>ls</b> عرض قائمة بالملفات والدلائل، <b>-l</b> بالتفاصيل، للدليل <b>/etc</b>، ثم علامة <b> </b> لإعادة توجيه الناتج الى الامر <b>more</b> (أمر <b>more</b> يعرض النتائج على شكل صفحات).</p>
<p><b>cut -d: -f1 /etc/passwd</b></p>	<p><b>cut</b> أمر استخراج من قاعدة بيانات، <b>-d</b> للتحديد ، و <b>f1</b> للحقل رقم ١، وذلك للملف <b>passwd</b> والذي يحتوي على أسماء المستخدمين والموجود بالدليل <b>/etc/passwd</b> (مسار مطلق).</p>
<p><b>cut -d: -f1 /etc/passwd   sort</b></p>	<p><b>cut</b> أمر استخراج من قاعدة بيانات، <b>-d</b> للتحديد ، و <b>f1</b> للحقل رقم ١، وذلك للملف <b>passwd</b> والذي يحتوي على أسماء المستخدمين والموجود بالدليل <b>/etc/passwd</b> (مسار مطلق). ثم الرمز <b> </b> لإعادة توجيه النتائج الى الامر <b>sort</b> والذي يقوم بفرز تلك النتائج من الأقل الى الأعلى.</p>
<p><b>cut -d: -f1 /etc/passwd   sort   more</b></p>	<p><b>cut</b> أمر استخراج من قاعدة بيانات، <b>-d</b> للتحديد ، و <b>f1</b> للحقل رقم ١، وذلك للملف <b>passwd</b> والذي يحتوي على أسماء المستخدمين والموجود بالدليل <b>/etc/passwd</b> (مسار مطلق). ثم الرمز <b> </b> لإعادة توجيه النتائج الى الامر <b>sort</b> والذي يقوم</p>

	بفرز تلك النتائج. ثم اعد توجيه الناتج الى الامر <b>more</b> لعرض النتائج على شكل صفحات وذلك باستخدام رمز الانبوب   .
<b>find ~ -name "*"bash*"</b>	<b>find</b> أمر بحث، في الموقع ~ بمعيار الاسم عن "*"bash*".
<b>find ~/Music -mmin -5</b>	<b>find</b> أمر بحث، في الموقع ~/Music ~ بمعيار الوقت -mmin عن الملفات التي تم التعديل عليها خلال الدقائق 5- الماضية.
<b>find /usr -size +2M</b>	<b>find</b> أمر بحث، في الموقع /usr بمعيار الحجم -size عن الملفات التي حجمها أكبر من 2 ميغا.
<b>find /usr/share/bug -type d</b>	<b>find</b> أمر بحث، في الموقع /usr/share/bug بمعيار النوع -type d عن الكائنات التي من نوع دليل.
<b>find /usr/share/bug -type d -ls</b>	<b>find</b> أمر بحث، في الموقع /usr/share/bug بمعيار النوع -type d عن الكائنات التي من نوع دليل، و ls لعرض قائمة بها.
<b>cat /etc/passwd</b>	<b>cat</b> أمر عرض المحتويات، للملف passwd كما هو موضح بالمسار المطلق /etc/passwd .
<b>more /etc/passwd</b>	<b>more</b> لعرض الملف على شكل صفحات، للملف passwd كما هو موضح بالمسار المطلق /etc/passwd .
<b>h</b>	للبحث عن المساعدة داخل صفحات الامر <b>more</b> .
<b>&lt;SPACE&gt;</b>	للانتقال للصفحة التالية داخل صفحات الامر <b>more</b> .
<b>less /etc/passwd</b>	<b>less</b> أمر أحدث لعرض الملف على شكل صفحات، للملف passwd كما هو موضح بالمسار المطلق /etc/passwd .
<b>/bin</b>	للبحث داخل صفحات الامر <b>less</b> عن /bin .
<b>n</b>	لعرض اول نتيجة للبحث (bin)، وبالتالي عند الضغط على n سينتقل التحديد الى النتيجة التالية، وهكذا دواليك.
<b>N</b>	للعودة الى النتيجة السابقة في البحث.

<b>q</b>	للخروج من صفحات الامر <b>less</b> .
<b>head /etc/passwd</b>	<b>head</b> عرض اول عشرة أسطر، من الملف <b>passwd</b> .
<b>tail /etc/passwd</b>	<b>tail</b> عرض آخر عشرة أسطر، من الملف <b>passwd</b> .
<b>head -2 /etc/passwd</b>	<b>head</b> عرض أول عشرة أسطر، ولكن مع -2 سيتم عرض أول سطرين فقط، من الملف <b>passwd</b> .
<b>ls /etc   tail -5</b>	<b>ls</b> عرض قائمة بالملفات والدلائل، للدليل <b>/etc</b> ، ورمز الأنبوب   لإعادة التوجيه للأمر <b>tail</b> لعرض آخر خمسة أسطر.
<b>head -n -20 /etc/passwd</b>	<b>head</b> لعرض الأسطر من بداية الملف، -20 -n لإنقاص 20 سطر من العدد الكلي للأسطر للملف <b>/etc/passwd</b> . وبالتالي عرض الباقي فقط.
<b>fgrep</b>	امر للبحث عن طريق مطابقة الأحرف.
<b>grep sshd passwd</b>	<b>grep</b> امر للبحث عن سلسلة احرف، وهي <b>sshd</b> في الملف <b>passwd</b> بالمسار النسبي.
<b>grep '^root' passwd</b>	<b>grep</b> امر للبحث عن سلسلة احرف، وهي <b>root</b> ، وعلامة <sup>^</sup> تعني أن تكون الكلمة أو السلسلة في بداية السطر فقط، وذلك في الملف <b>passwd</b> .
<b>grep 'sync' passwd</b>	<b>grep</b> امر للبحث عن سلسلة احرف، وهي <b>sync</b> ، وذلك في الملف <b>passwd</b> أيًا كان موقعها بالسطر.
<b>grep 'sync\$' passwd</b>	<b>grep</b> للبحث عن سلسلة احرف، وهي <b>sync</b> ، وعلامة \$ تعني أن تكون الكلمة أو السلسلة في نهاية السطر فقط، في <b>passwd</b> .
<b>grep '.y' passwd</b>	<b>grep</b> امر للبحث عن سلسلة احرف، وهي حرف <b>y</b> في اخر الكلمة وإظهار الحرف السابق لها في الملف <b>passwd</b> .

<p><b>grep 'sshd root operator'</b> passwd</p>	<p><b>grep</b> امر للبحث عن سلسلة احرف، وذلك عن جميع الكلمات sshd root operator للملف passwd ، وبما ان الامر <b>grep</b> لا يميز الرمز (   )، فبالتالي سيعتبر الكلمات السابقة سلسلة واحدة، ويجب أن تكون جميعها في سطر واحد وإلا فإنه لن يقوم بإرجاع أي نتيجة.</p>
<p><b>grep -E 'sshd root operator'</b> passwd</p>	<p><b>grep</b> امر للبحث عن سلسلة من الاحرف، <b>-E</b> هو الوضع الموسع ليتم تمييز الرمز (   )، أي سيتم البحث عن الكلمات التالية sshd root operator او احدها على الأقل.</p>
<p><b>egrep 'no(b n)'</b> passwd</p>	<p><b>egrep</b> امر آخر للبحث بالوضع الموسع، عن سلسلة no والتي تنتهي إما بالحرف b او بالحرف n في الملف passwd. وهذا هو الإصدار الاحدث للأمر <b>grep</b> وفيه يتم استخدام التعابير مثل (-  - ? + - ).</p>
<p><b>head passwd   grep '[0-9]'</b></p>	<p><b>head</b> عرض اول عشرة أسطر، من الملف passwd (مسار نسبي)، ثم إعادة توجيه النتائج الى الامر <b>grep</b> للبحث عن الأرقام في المدى من 0 إلى 9 .</p>
<p><b>grep -E '[0-9]{3}'</b> passwd</p>	<p><b>grep</b> امر للبحث عن سلسلة من الاحرف، <b>-E</b> هو الوضع الموسع للبحث عن مدى الأرقام من 0 إلى 9 وعرض أول 3 خانات فقط وذلك للملف passwd.</p>

## الملحق رقم (٦)

### ملخص لأوامر المعمل التاسع:

Command	معنى الأوامر
Insert modes key	معنى مفاتيح وضع الإدراج
Cursor navigation Key	معنى مفاتيح التحكم بالمؤشر في وضع الأوامر
Save & Exit command	معنى أوامر الحفظ والخروج في وضع الأوامر
<b>vi myfile</b>	لإنشاء ملف برنامج نصي مع محرر <b>vi</b> .
<b>i</b>	وضع الإدراج: للدخول في وضع الإدراج.
<b>a</b>	وضع الإدراج: تحريك المؤشر لليمين ثم الدخول لوضع الإدراج.
<b>o</b>	وضع الإدراج: لإدراج سطر جديد تحت السطر الحالي.
<b>O</b>	وضع الإدراج: لإدراج سطر جديد فوق السطر الحالي.
<b>I</b>	وضع الإدراج: الدخول لوضع الإدراج من بداية السطر الحالي.
<b>A</b>	وضع الإدراج: ستدخل في وضع الإدراج في نهاية السطر
<ESC>	للخروج إلى وضع الأوامر.
<b>j</b>	وضع الأوامر: نقل المؤشر سطر للأسفل.
<b>k</b>	وضع الأوامر: نقل المؤشر سطر للأعلى.
<b>l</b>	وضع الأوامر: نقل المؤشر خانة واحدة يمين السطر.
<b>8l</b>	وضع الأوامر: نقل المؤشر من المكان الحالي الى يمين السطر 8 خانات.
<b>10l</b>	وضع الأوامر: نقل المؤشر من المكان الحالي الى يمين السطر 10 خانات.
<b>h</b>	وضع الأوامر: نقل المؤشر خانة واحدة يسار السطر.

w	وضع الأوامر: نقل المؤشر إلى بداية الكلمة التالية في السطر.
e	وضع الأوامر: نقل المؤشر إلى نهاية الكلمة الحالية.
b	وضع الأوامر: نقل المؤشر إلى بداية الكلمة السابقة.
\$	وضع الأوامر: نقل المؤشر إلى نهاية السطر الحالي (كمفتاح End).
0 (صفر)	وضع الأوامر: نقل المؤشر إلى بداية السطر الحالي (كمفتاح Home).
nG	وضع الأوامر: اكتب بدل n رقم السطر الذي تريد الذهاب إليه.
3G	وضع الأوامر: الذهاب إلى السطر الثالث.
1G	وضع الأوامر: الذهاب إلى السطر الأول.
G	وضع الأوامر: الذهاب إلى السطر الأخير.
dw	وضع الأوامر: حذف الكلمة الموجود عندها المؤشر.
u	وضع الأوامر: التراجع عن العملية الأخيرة.
2dw	وضع الأوامر: حذف كلمتين متتاليتين، بداية من الكلمة الموجود عندها المؤشر والتي تليها.
xxxx	وضع الأوامر: حذف أربعة خانوات، بداية من الخانة التي بها المؤشر.
4u	وضع الأوامر: التراجع عن العمليات الاربعة الأخيرة.
14x	وضع الأوامر: حذف 14 خانة، بداية من الخانة التي بها المؤشر.
5X	وضع الأوامر: حذف 5 خانوات قبل المؤشر (يسار المؤشر).
dd	وضع الأوامر: حذف سطر الحالي كاملاً.
p	وضع الأوامر: لصق ما تم حذفه، كالسطر المحذوف بالخطوة السابقة.
2u	وضع الأوامر: التراجع عن العمليتين الأخيرتين.
2dd	وضع الأوامر: حذف سطرين، الحالي والذي يليه.
4w	وضع الأوامر: نقل المؤشر إلى بداية الكلمة الرابعة في السطر الحالي.

<b>D</b>	وضع الأوامر: حذف الخانات من المكان وجود المؤشر إلى نهاية السطر.
<b>J</b>	وضع الأوامر: ضم سطرين، الحالي والذي يليه.
<b>yw</b>	وضع الأوامر: نسخ الكلمة الموجود بها المؤشر.
<b>P</b>	وضع الأوامر: لصق ما تم نسخه سابقاً.
<b>3J</b>	وضع الأوامر: ضم 3 اسطر معاً، بداية من الحالي ثم ما يليه.
<b>:%s/ text //g</b>	وضع الأوامر: البحث عن كلمة text ثم حذفها (اضف مسافة قبل وبعد كلمة text).
<b>~</b>	وضع الأوامر: تغيير حالة الأحرف من صغير الى كبير والعكس.
<b>:w</b>	وضع الأوامر: لحفظ الملف.
<b>:x</b>	وضع الأوامر: حفظ وإغلاق الملف.
<b>:wq</b>	وضع الأوامر: للكتابة على ملف والخروج.
<b>:wq!</b>	وضع الأوامر: للكتابة على ملف للقراءة فقط إن كان ممكناً والخروج.
<b>ZZ</b>	وضع الأوامر: حفظ وإغلاق (لاحظ عدم استخدام النقطتين : )
<b>:q!</b>	وضع الأوامر: الخروج من غير حفظ.
<b>:e!</b>	وضع الأوامر: تجاهل التغييرات وتحميل الملف مجدداً.
<b>:w!</b>	وضع الأوامر: للكتابة على ملف للقراءة فقط إن كان ممكناً
<b>/line</b>	وضع الأوامر: البحث عن كلمة line.
<b>n</b>	وضع الأوامر: الانتقال الى نتيجة البحث التالية.
<b>?line</b>	وضع الأوامر: البحث للخلف ( ما قبل الموضع الحالي) عن كلمة line.
<b>cw</b>	وضع الأوامر: استبدال الكلمة الحالية، فيتم حذفها والدخول لوضع الإدراج
<b>bash sample.sh</b>	أمر لتشغيل الملف sample.sh .

	ملاحظة: يجب ان يحتوي الملف على الجملة <code>#!/bin/bash</code>
<code>./sample.sh</code>	<code>./</code> أمر آخر للتشغيل، للملف <code>sample.sh</code> .
<code>cal</code>	لعرض التقويم.
<code>chmod a+x sample.sh</code>	<code>chmod</code> لتغيير الاذونات، <code>a</code> لجميع المستخدمين، <code>+</code> إعطاء إذن، <code>x</code> التنفيذ، على الملف <code>sample.sh</code> (مسار نسبي).
<code>read age</code>	قراءة المتغير المسمى باسم <code>age</code> .
<code>if test \$age -lt 16</code>	<code>if</code> جملة إذا الشرطية، <code>test</code> جملة اختبار، <code>\$age</code> هو متغير، <code>-lt</code> اصغر من الرقم 16. بمعنى أن المتغير <code>\$age</code> اقل من 16.
<code>then</code>	النتيجة إذا كانت جملة الاختبار صحيحة.
<code>else</code>	<code>else</code> غير ذلك، بمعنى إذا كانت النتيجة خاطئة.
<code>fi</code>	نهاية جملة إذا الشرطية.
<code>if [ \$age -lt 16 ]</code>	طريقة أخرى لاستدعاء جملة الاختبار <code>test</code> ، بوضع الشرط داخل الأقواس المربعة مع ترك مسافة بين القوسين. ملاحظة: سواء تحقق الشرط أو لم يتحقق فإن برنامج <code>if</code> ينتهي بعد الإدخال الأول.
<code>while [ \$num -le 100 ]</code>	<code>while</code> جملة للتكرار، والأقواس المربعة [ ] هي لاستدعاء أمر الاختبار <code>test</code> (طريقة أخرى لكتابة جملة الاختبار)، و <code>\$num</code> هو متغير يتم إدخاله من قبل المستخدم، <code>le 100</code> تعني أن الرقم مساوي أو أصغر من 100. لاحظ أن التكرار مرتبط بتحقق الشرط، فإذا لم يتحقق تستمر الحلقة، وإذا تحقق ينتهي البرنامج.
<code>do</code>	طالما الشرط متحقق، افعل التالي.
<code>done</code>	نهاية التكرار عند تحقق الشرط.
<code>for</code>	إجراء مجموعة من العمليات على سطر الأوامر.
<code>seq</code>	قائمة من قيم الاعداد الصحيحة.

## الملحق رقم (٧)

ملخص لأوامر المعمل العاشر:

Command -option string	المعنى
<b>lscpu</b>	عرض تفاصيل وحدة المعالجة.
<b>cat /proc/cpuinfo</b>	<b>cat</b> أمر عرض المحتويات، للملف <b>cpuinfo</b> والذي يحتوي على تفاصيل وحدة المعالجة.
<b>free -k</b>	<b>free</b> اكتشاف مقدار الذاكرة RAM، و <b>-k</b> بالكيلوبايت.
<b>free -m</b>	<b>free</b> اكتشاف مقدار الذاكرة RAM، و <b>-m</b> بالميجابايت.
<b>free -g</b>	<b>free</b> اكتشاف مقدار الذاكرة RAM، و <b>-g</b> بالجيجابايت.
<b>lspci</b>	عرض الأجهزة المتصلة بنواقل PCI
<b>lspci -k</b>	عرض الأجهزة المتصلة بنواقل PCI، <b>-k</b> لإظهار مشغل النواة والوحدات المستخدمة.
<b>lsusb</b>	عرض أجهزة usb المتصلة
<b>lshal</b>	يقوم بسرد قائمة بالأجهزة <b>devices</b> في النظام. ملاحظة: لا يعمل الأمر <b>lshal</b> عادة في البيئة الافتراضية VM.
<b>lsmod</b>	عرض وحدات يتم تحميلها حالياً.
<b>dmi decode</b>	يقوم بقراءة وطباعة المعلومات من SMBIOS.
<b>fdisk -cul</b>	<b>fdisk</b> لسرد محركات الأقراص، <b>-c</b> فيقوم بمنع طباعة التحذيرات التوافق DOS، <b>u</b> لعرض وحدات التخزين على شكل قطاعات بدلا من اسطوانات. أما <b>l</b> فهو لإظهار المحركات بشكل غير تفاعلي.

## الملحق رقم (٨)

ملخص لأوامر المعمل الحادي عشر:

Command -option string	المعنى
<b>ls /proc</b>	<b>ls</b> عرض قائمة، بالملفات الموجودة في الدليل /proc
<b>cat /proc/1/cmdline; echo</b>	<b>cat</b> عرض محتويات، لسطر الأوامر cmdline للعملية رقم (1) والموجودة في دليل العمليات /proc . معلومة: العملية الأولى خاصة ببدا التشغيل وهي /sbin/init
<b>ps -p 1</b>	<b>ps</b> عرض العمليات قيد التشغيل؛ والخيار <b>-p</b> لتحديد رقم العملية التي نريد عرضها وهي العملية رقم 1.
<b>cat /proc/cmdline</b>	<b>cat</b> عرض محتويات، لسطر الأوامر cmdline والموجودة في دليل العمليات /proc . وهي المعطيات التي تم تمريرها للنواة وقت التمهيد.
<b>ping localhost &gt; /dev/null</b>	<b>ping</b> يقوم باختبار الاتصال، للجهاز localhost، ثم توجيه الإخراج القياسي إلى الملف /dev/null. ويعتبر كسلة للمحذوفات. ملاحظة: العملية السابقة تعمل في المقدمة وبالتالي لا نستطيع كتابة أي أمر في موجه الأوامر إلا اذا قمنا بإيقاف أو إنهاء تلك العملية.
<b>Ctrl+c</b>	إنهاء العملية أو الوظيفة التي تعمل بالمقدمة بشكل نهائي وليس للإيقاف. واستعادة السيطرة على النافذة الطرفية.
<b>ping localhost &gt; /dev/null &amp;</b>	<b>ping</b> يقوم باختبار الاتصال، للجهاز localhost، ثم توجيه الإخراج القياسي إلى الملف /dev/null، أما الرمز & فهو لجعل الامر يعمل الخلفية. وبالتالي نستطيع الكتابة في موجه الأوامر.
<b>jobs</b>	معرفة أرقام الوظائف قيد التشغيل، وتأتي بين قوسين مربعة [ ] .

<b>fg %1</b>	<b>fg</b> إحضار للمقدمة، للوظيفة الأولى ذات الرقم 1. ملاحظة: يستخدم الرمز % مع الوظائف وليس مع العمليات.
<b>Ctrl+z</b>	يقاف الوظيفة التي بالمقدمة (وليس إنهاؤها) واستعادة السيطرة على النافذة الطرفية.
<b>fg %2</b>	<b>fg</b> إحضار للمقدمة، للوظيفة الثانية ذات الرقم 2.
<b>bg %1</b>	<b>bg</b> إرجاع للخلف، للوظيفة الأولى ذات الرقم 1.
<b>kill %3</b>	<b>kill</b> امر لإنهاء، وذلك للوظيفة الثالثة ذات الرقم 3. أما الرمز % فيستخدم للتفريق بين إنهاء الوظيفة أو إنهاء العملية.
<b>Killall ping</b>	<b>Killall</b> أمر لإنهاء الجميع (من نوع واحد) ، وهنا يستخدم لإنهاء الأوامر التي من نوع ping .
<b>top</b>	برنامج تفاعلي لإدارة العمليات، وبالتالي إصدار الأوامر داخله.
<b>k</b>	في الشاشة التفاعلية داخل top: لإنهاء عملية من العمليات.
<b>99</b>	في الشاشة التفاعلية داخل top: رقم العملية التي نريد إنهاؤها.
<b>Kill PID with signal [15]</b>	في الشاشة التفاعلية داخل top: لإنهاء العملية مع إرسال الإشارة الافتراضية رقم [15] اضغط على مفتاح إدخال Enter. وتستخدم تلك الإشارة لتحديد طريقة الإنهاء.
<b>Kill PID with signal [15]:9</b>	في الشاشة التفاعلية داخل top: لإنهاء العملية مع إرسال إشارة غير الافتراضية وهي رقم 9 تعتبر إشارة قوية لا يمكن تجاهلها.
<b>q</b>	للخروج من الأمر top.
<b>sleep 888888 &amp;</b>	<b>sleep</b> عمل نقطة توقف pause لأحد برنامج القشرة وذلك لفترة محددة من الزمن، و 888888 هو المكان الذي نريد إيقافه، و رمز & لجعل العملية تعمل في الخلفية.
<b>ps</b>	عرض بعض العمليات قيد التشغيل.

<b>kill 103</b>	<b>kill</b> امر إنهاء، للعملية ذات المعرف رقم PID 103.
<b>pkill -15 sleep</b>	<b>pkill</b> أمر إنهاء برنامج، وذلك بكتابة اسم البرنامج الذي نريد إنهائه وهو <b>sleep</b> ، أما رقم 15 فهو إشارة ترسل مع الأمر وذلك لتحديد طريقة الإنهاء.
<b>ps -e</b>	<b>ps</b> أمر عرض العمليات قيد التشغيل؛ أما <b>-e</b> فهو لعرض كافة العمليات.
<b>ps -o pid, tty, time, %cpu, cmd</b>	<b>ps</b> أمر عرض العمليات قيد التشغيل؛ و <b>-o</b> لتحديد الأعمدة التي نريد عرضها. حيث أن <b>%cpu</b> هو عامود جديد غير الأعمدة الافتراضية الأربعة الباقية.
<b>ps -o pid, tty, time, %mem, cmd -- sort %mem</b>	<b>ps</b> أمر عرض العمليات قيد التشغيل؛ و <b>-o</b> لتحديد الأعمدة التي نريد عرضها وهي <b>pid, tty, time, %mem, cmd</b> ، وأخيراً الامر <b>sort</b> لفرز الأعمدة تنازلياً حسب العامود <b>mem</b> .
<b>free</b>	<b>free</b> اكتشاف مقدار الذاكرة RAM.
<b>su - root</b>	<b>su</b> تبديل المستخدم، <b>root</b> - إلى المستخدم الجذر، وسيطلب النظام كلمة مرور للدخول إلى الحساب.
<b>ssh localhost</b>	الأمر <b>ssh</b> (يستخدم لتسجيل الدخول عن بعد)، وهنا نستخدمه لتوليد بيانات تسجيل الدخول (الناجحة والفاشلة) في ملف <b>var/log/auth.log/</b> .
<b>tail -5 /var/log/auth.log</b>	<b>tail</b> أمر عرض آخر 10 أسطر، و <b>-5</b> لعرض خمسة أسطر فقط من الملف على <b>auth.log</b> ، وسيتم عرض محاولات تسجيل الدخول والتي تمت بواسطة الأمر <b>ssh</b> السابق.
<b>crontab -e</b>	<b>crontab</b> برنامج للتنصيب أو للإزالة ويستخدم أيضاً في المهام المجدولة، <b>-e</b> للتعديل على السجل.

<p>0 2 * * 0 who &gt;&gt;  /tmp/whothere</p>	<p>هذا الامر يقوم بعملية جدولة كالتالي: الرقم الأول للدقائق، والثاني للساعات، والثالث لترتيب اليوم في الشهر، والرابع للشهر، والخامس لترتيب اليوم في الأسبوع، و الأمر <b>who</b> فهو لعرض المستخدمين الحاليين للنظام، و &gt;&gt; لإعادة توجيه البيانات السابقة مع الاحتفاظ بما يوجد في الملف <b>whothere</b> الموجود في المسار المطلق <b>/tmp/whothere</b> .</p>
<p><b>crontab -l   tail -2</b></p>	<p><b>crontab</b> برنامج للتنصيب او للإزالة ويستخدم ايضاً في المهام المجدولة، <b>-l</b> لعرض محتويات السجل ولكن في شاشة الإخراج القياسي (<b>stdout</b>)، ورمز   خاص بتحويل المخرجات، الى الأمر <b>tail -2</b> والذي سيعرض آخر سطرين من السجل.</p>
<p><b>tail /var/log/cron.log</b></p>	<p><b>tail</b> عرض آخر ١٠ أسطر، من السجل <b>cron.log</b> والموجود بالمسار المطلق <b>/var/log/cron.log</b> .</p>
<p><b>sudo tail -5</b>  /var/log/cron.log</p>	<p><b>sudo</b> أمر للحصول على امتيازات الجذر، <b>tail -5</b> عرض آخر خمسة أسطر، من الملف <b>cron.log</b> والموجود بالمسار المطلق <b>/var/log/cron.log</b> .</p>
<p><b>ldconfig -p   less</b></p>	<p><b>ldconfig</b> أمر لتحديث ذاكرة التخزين <b>cache</b> والروابط الرمزية، <b>-p</b> يعرض قائمة المكتبات المشتركة، ورمز   فهو لإعادة التوجيه، <b>less</b> أمر للعرض على شكل صفحات.</p>
<p><b>ldd /bin/bash</b></p>	<p><b>ldd</b> طباعة المكتبة المشتركة وما يتبعها، للملف <b>bash</b> القابل للتنفيذ والموجود بالمسار المطلق <b>/bin/bash</b> .</p>

## الملحق رقم (٩)

ملخص لأوامر المعمل الثاني عشر:

Command -option string	المعنى
<b>ifconfig</b>	عرض عنوان الجهاز (بروتوكول الانترنت IP)
<b>route</b>	عرض معلومات جدول التوجيه
<b>route -n</b>	<b>route</b> عرض جدول التوجيه، <b>-n</b> للعرض على شكل أرقام.
<b>grep 127.0.0.1 /etc/hosts</b>	<b>grep</b> أمر بحث عن سلسلة، وهي عنوان 127.0.0.1 في الملف <b>hosts</b> والموجود في المسار المطلق <b>/etc/hosts</b> .
<b>ping -c4 localhost</b>	<b>ping</b> أمر اختبار اتصال، <b>-c</b> قم بوقف الاختبار بعد ارسال 4 حزم <b>packets</b> ، للجهاز <b>localhost</b> .
<b>cat /etc/resolv.conf</b>	<b>cat</b> عرض محتويات، الملف <b>resolv.conf</b> والموجود بالمسار المطلق <b>/etc/resolv.conf</b> لمعرفة <b>nameserver</b>
<b>dig localhost.localdomain</b>	<b>dig</b> أمر خاص بالبحث لإعادة تحويل الاسم إلى عنوان IP، للاسم <b>localhost.localdomain</b> .
<b>dig cserver.example.com</b>	<b>dig</b> أمر خاص بالبحث لإعادة تحويل الاسم إلى عنوان IP، للاسم <b>cserver.example.com</b> .
<b>dig -x 192.168.1.2</b>	<b>dig</b> أمر خاص بالبحث لإعادة تحويل الاسم إلى عنوان IP، <b>-x</b> لعكس العملية وتحويل عنوان IP إلى اسم للعنوان <b>192.168.1.2</b> .
<b>cat /etc/services</b>	<b>cat</b> لعرض المحتويات، للملف <b>services</b> والذي يحتوي على قائمة بأسماء وارقام المنافذ.
<b>netstat --help</b>	<b>netstat</b> لعرض معلومات عن الشبكة، <b>--help</b> للحصول على استخدامات الامر <b>netstat</b> .

<b>netstat -tl</b>	<b>netstat</b> لعرض معلومات عن الشبكة، <b>t</b> خاص بـمنافذ TCP، <b>l</b> لمعرفة حالة الاستماع، وبعبارة أخرى معرفة ما اذا كانت منافذ TCP تقوم بالاستماع إلى اتصال وارد أو لا.
<b>netstat -tln</b>	<b>netstat</b> لعرض معلومات عن الشبكة، <b>t</b> خاص بـمنافذ TCP، <b>l</b> لمعرفة حالة الاستماع، و <b>n</b> لإحضار النتائج على شكل أرقام وليس رموز. وبعبارة أخرى معرفة ما اذا كانت منافذ TCP تقوم بالاستماع إلى اتصال وارد أو لا.
<b>netstat -t</b>	<b>netstat</b> لعرض معلومات عن الشبكة، <b>-t</b> لعرض قنوات الاتصال sockets لطبقة tcp.
<b>netstat -u</b>	<b>netstat</b> لعرض معلومات عن الشبكة، <b>-u</b> لعرض قنوات الاتصال sockets لطبقة udp.
<b>netstat -w</b>	<b>netstat</b> لعرض معلومات عن الشبكة، <b>-w</b> لعرض قنوات الاتصال sockets لطبقة raw.
<b>netstat -x</b>	<b>netstat</b> لعرض معلومات عن الشبكة، <b>-x</b> لعرض قنوات الاتصال sockets لطبقة unix.
<b>netstat -a</b>	<b>netstat</b> لعرض معلومات عن الشبكة، <b>-a</b> لعرض جميع قنوات الاتصال sockets سواء في حالة استماع أو لا.
<b>netstat -l</b>	<b>netstat</b> لعرض معلومات عن الشبكة، <b>-l</b> للاستماع الى احد المنافذ.
<b>netstat -n</b>	<b>netstat</b> لعرض معلومات عن الشبكة، <b>-n</b> للعرض على شكل ارقام.
<b>netstat --numeric-hosts</b>	<b>netstat</b> لعرض معلومات عن الشبكة، <b>--numeric-hosts</b> عرض عناوين المضيف على شكل ارقام، دون التأثير على اسم المنفذ أو المستخدم.

<b>netstat --numeric-ports</b>	<b>netstat</b> لعرض معلومات عن الشبكة، <b>--numeric-ports</b> عرض المنافذ على شكل ارقام، دون التأثير على اسم المضيف أو أسماء المستخدمين.
<b>netstat --numeric-users</b>	<b>netstat</b> لعرض معلومات عن الشبكة، <b>--numeric-users</b> عرض معرفات المستخدم على شكل ارقام، دون التأثير على اسم المضيف أو أسماء المنفذ.
<b>netstat -N</b>	<b>netstat</b> لعرض معلومات عن الشبكة، <b>-N</b> عرض أسماء العنود.
<b>netstat -e</b>	<b>netstat</b> لعرض معلومات عن الشبكة، <b>-e</b> عرض معلومات إضافية.
<b>netstat -o</b>	<b>netstat</b> لعرض معلومات عن الشبكة، <b>-o</b> عرض المعلومات والمتعلقة بالمؤقت الشبكي <b>networking timers</b> .
<b>netstat -p</b>	<b>netstat</b> لعرض معلومات عن الشبكة، <b>-p</b> لعرض رقم العملية PID وأيضا اسم البرنامج الذي ينتمي إليه كل مأخذ توصيل <b>socket</b> .
<b>netstat -v</b>	<b>netstat</b> لعرض معلومات عن الشبكة، <b>-v</b> اعرض الحدث.
<b>netstat -c</b>	<b>netstat</b> لعرض معلومات عن الشبكة، <b>-c</b> طباعة المعلومات المحددة كل ثانية بشكل متواصل.
<b>netstat -r</b>	<b>netstat</b> لعرض معلومات عن الشبكة، <b>-r</b> لعرض جدول التوجيه.
<b>netstat -i</b>	<b>netstat</b> لعرض معلومات عن الشبكة، <b>-i</b> لعرض الواجهات (interfaces).
<b>netstat -g</b>	<b>netstat</b> لعرض معلومات عن الشبكة، <b>-g</b> عرض معلومات عضوية مجموعة البث المتعدد <b>multicast</b> لـ IPv4 و IPv6
<b>netstat -M</b>	<b>netstat</b> لعرض معلومات عن الشبكة، <b>-M</b> عرض قائمة باتصالات <b>masqueraded</b> (برنامج اتصال بلينكس مشابه لنظام 1:many)
<b>netstat -s</b>	<b>netstat</b> لعرض معلومات عن الشبكة، <b>-s</b> عرض تلخيص بإحصائيات كل بروتوكول.

<b>netstat -W</b>	<b>netstat</b> لعرض معلومات عن الشبكة، <b>-W</b> عدم اقتطاع عناوين IP باستخدام تقنية " إخراج واسع حسب الحاجة " .
<b>netstat -F</b>	<b>netstat</b> لعرض معلومات عن الشبكة، <b>-F</b> طباعة جدول معلومات التوجيه من (Forwarding Information Base : FIB) .
<b>netstat -C</b>	<b>netstat</b> لعرض معلومات عن الشبكة، <b>-C</b> طباعة معلومات التوجيه من ذاكرة الطرق الموجودة في cache بدلاً من FIB .
<b>netstat -V</b>	<b>netstat</b> لعرض معلومات عن الشبكة، <b>-V</b> للإصدار .

## الملحق رقم (١٠)

ملخص لأوامر المعمل الثالث عشر:

Command -option string	المعنى
<b>head /etc/passwd</b>	<b>head</b> عرض أول 10 أسطر من ملف <b>passwd</b> والموجود بالمسار المطلق <b>/etc/passwd</b> ، والخاص بحسابات المستخدمين والنظام.
<b>grep sysadmin /etc/passwd</b>	<b>grep</b> امر للبحث عن سلسلة من الاحرف، وهي المستخدم <b>sysadmin</b> في الملف <b>passwd</b> والموجود بالمسار المطلق <b>/etc/passwd</b>
<b>getent passwd sysadmin</b>	<b>getent</b> استرجاع معلومات من قاعدة بيانات او الملف، وهو <b>passwd</b> والخاص بحسابات المستخدمين، حول المستخدم <b>.sysadmin</b> .
<b>man 5 passwd</b>	<b>man</b> للبحث في صفحات دليل المستخدم، ورقم 5 يعني البحث داخل القسم رقم 5 ، وأخيرا الكلمة التي نريد البحث عنها <b>.passwd</b> .
<b>id</b>	<b>id</b> عرض معلومات الهوية لحساب مستخدم، وإذا لم يكتب أي شيء بعد الأمر فستظهر معلومات الهوية الخاصة بالمستخدم الحالي للنظام.
<b>id root</b>	<b>id</b> عرض معلومات الهوية لحساب مستخدم، وهو <b>.root</b> .
<b>getent group sysadmin</b>	<b>getent</b> استرجاع معلومات من قاعدة بيانات او الملف، وهو <b>group</b> والخاص بالمجموعات، حول المجموعة <b>.sysadmin</b> .

<b>getent group adm</b>	<b>getent</b> استرجاع معلومات من قاعدة بيانات او الملف، وهو <b>group</b> والخاص بالمجموعات، حول المجموعة <b>adm</b> .
<b>whoami</b>	عرض اسم المستخدم الحالي للنظام.
<b>groups</b>	<b>groups</b> لعرض المجموعات التي ينتمي إليها مستخدم، وإذا لم يكتب أي شيء بعد الأمر فستظهر المجموعات الخاصة بالمستخدم الحالي للنظام.
<b>groups root</b>	<b>groups</b> لعرض المجموعات التي ينتمي إليها مستخدم، وهو <b>root</b> .
<b>who</b>	<b>who</b> أمر للحصول على القائمة الحالية من المستخدمين المتواجدين على النظام.
<b>w</b>	<b>w</b> الأمر للحصول على عرض أكثر تفصيلاً للمستخدمين الذين هم حالياً على النظام الخاص بك
<b>visudo</b>	<b>visudo</b> أمر لتعديل تكوينات الأمر <b>sudo</b> وذلك بغرض السماح للمستخدمين بالوصول إليه واستخدامه. الدخول للأمر <b>visudo</b> يكون عن طريق حساب الجذر. لاحظ أن الأمر سيفشل إذا لم يكن المستخدم هو <b>root</b> .
<b>su -</b>	<b>su</b> أمر بتبديل المستخدمين وذلك بكتابة اسم المستخدم بعد - ، إذا لم يتم كتابة أي شيء بعد - فهذا يعني الانتقال الى حساب <b>root</b> .
<b>visudo</b>	<b>visudo</b> أمر لتعديل تكوينات الأمر <b>sudo</b> وذلك بغرض السماح للمستخدمين بالوصول إليه واستخدامه. الدخول للأمر <b>visudo</b> يكون عن طريق حساب الجذر. لاحظ أن الأمر سينجح إذا كان بحساب المستخدم <b>root</b> .

<p><b>exit</b></p>	<p><b>exit</b> أمر للخروج من الحساب الحالي، وهنا سيتم الخروج من حساب الجذر الى حساب المستخدم <code>sysadmin</code>.</p>
<p><b>head -3 /etc/shadow</b></p>	<p><b>head</b> عرض أول 10 أسطر من ملف <code>shadow</code> والموجود بالمسار المطلق <code>/etc/shadow</code> ، و <code>-3</code> يعني عرض أول ثلاثة اسطر فقط. من الملف <code>shadow</code> والذي يحتوي على كلمات السر المشفرة. الأمر السابق سيفشل لأن <code>sysadmin</code> ليس له صلاحية عرض الملف <code>shadow</code> .</p>
<p><b>ls -l /etc/shadow</b></p>	<p><b>ls</b> عرض قائمة بالملفات والدلائل، و <code>-l</code> لإعطاء التفاصيل، للكائن <code>shadow</code>، لاحظ أن الاستعراض من المكان الحالي ولا نحتاج الى الانتقال إلى ذلك الكائن.</p>
<p><b>sudo head -3 /etc/shadow</b></p>	<p><b>sudo</b> أمر للحصول على امتيازات الجذر، <b>head</b> لعرض أول 10 أسطر من ملف <code>shadow</code> والموجود بالمسار المطلق <code>/etc/shadow</code> ، و <code>-3</code> يعني عرض أول ثلاثة اسطر فقط. من الملف <code>shadow</code> والذي يحتوي على كلمات السر المشفرة. الأمر السابق سينجح لأن <code>sysadmin</code> سيستخدم صلاحيات الجذر بواسطة الأمر <b>sudo</b>.</p>

## الملحق رقم ( ١١ )

ملخص لأوامر المعمل الرابع عشر:

Command -option string	المعنى
<b>su -</b>	<b>su</b> أمر تبديل المستخدمين وذلك بكتابة اسم المستخدم بعد - ، إذا لم يتم كتابة أي شيء بعد - فهذا يعني الانتقال الى حساب .root.
<b>groupadd -r research</b>	<b>groupadd</b> أمر انشاء مجموعة، <b>-r</b> يجعل معرف المجموعة GID في النطاق المحجوز بين ١-٩٩٩، وأخيراً اسم المجموعة وهو . research .
<b>groupmod -n run research</b>	<b>groupmod</b> للتعديل على مجموعة، وخيار <b>-n</b> لتغيير اسم هذه المجموعة الى الاسم run وذلك للمجموعة .research.
<b>groupmod -g 11 research</b>	<b>groupmod</b> للتعديل على مجموعة، وخيار <b>-g</b> من أجل تغيير معرف المجموعة GID، ثم بعد ذلك المعرف الجديد الذي تريده وهو على سبيل المثال 11 ثم اسم المجموعة.
<b>groupdel</b>	<b>groupdel</b> لحذف مجموعة طالما أنها ليست مجموعة أساسية لمستخدم، ثم بعد ذلك اسم المجموعة التي تريد حذفها.
<b>getent group research</b>	<b>getent</b> استرجاع معلومات من قاعدة بيانات او الملف، وهو group والخاص بالمجموعات، حول المجموعة .research.
<b>usermod -aG research sysadmin</b>	<b>usermod</b> أمر تعديل على مستخدم، <b>-a</b> لإضافة المستخدم الى مجموعة، <b>G</b> لجعلها مجموعة ثانوية له، ثم اسم المجموعة .research، وأخيراً المستخدم sysadmin.
<b>usermod -aG A,B sysadmin</b>	<b>usermod</b> أمر تعديل على مستخدم، <b>-a</b> لإضافة المستخدم الى مجموعة، <b>G</b> لجعلها مجموعة ثانوية له، ثم اسم المجموعة A

	ونستخدم علامة الفاصلة " , " لإضافة مجموعة ثانوية أخرى وهي B وذلك للمستخدم sysadmin.
<b>usermod -G X sysadmin</b>	<b>usermod</b> أمر تعديل على مستخدم، <b>G</b> لإضافة المستخدم الى مجموعة ثانوية وحذف جميع المجموعات الثانوية السابقة، ثم اسم المجموعة X وذلك للمستخدم sysadmin. لاحظ ان عدم استخدام الخيار <b>a</b> سيحذف جميع المجموعات الثانوية السابقة.
<b>groups sysadmin</b>	<b>groups</b> لعرض المجموعات التي ينتمي إليها مستخدم، وهو sysadmin.
<b>id sysadmin</b>	<b>id</b> عرض معلومات الهوية لحساب مستخدم، وهو sysadmin.
<b>getent group research</b>	<b>getent</b> استرجاع معلومات من قاعدة بيانات او الملف، وهو group والخاص بالمجموعات، حول المجموعة research.
<b>useradd -G research -c 'Linux Student' -m student</b>	<b>useradd</b> أمر لإنشاء مستخدم جديد، <b>-G</b> لجعله عضو ثانوي في المجموعة research، <b>-c</b> لوضع تعليق وهو اسم المستخدم الكامل والذي يظهر في الواجهة الرسومية وهو هنا 'Linux Student'، <b>-m</b> لإنشاء دليل رئيسي لهذا المستخدم، وأخيراً اسم هذا المستخدم الجديد student.
<b>getent group research</b>	<b>getent</b> استرجاع معلومات من قاعدة بيانات او الملف، وهو group والخاص بالمجموعات، حول المجموعة research.
<b>getent group student</b>	<b>getent</b> استرجاع معلومات من قاعدة بيانات او الملف، وهو group والخاص بالمجموعات، حول المجموعة student.
<b>getent passwd student</b>	<b>getent</b> استرجاع معلومات من قاعدة بيانات او الملف، وهو passwd والخاص بحسابات المستخدمين، حول المستخدم student.

<b>getent shadow student</b>	<b>getent</b> استرجاع معلومات من قاعدة بيانات او الملف، وهو <b>shadow</b> والخاص بكلمات المرور، حول المستخدم <b>student</b> . لاحظ ظهور الرمز! في الحقل الثاني وذلك لعدم تعيين كلمة مرور للمستخدم <b>student</b> .
<b>passwd student</b>	تعيين باسورد للمستخدم <b>student</b>
<b>getent shadow student</b>	<b>getent</b> استرجاع معلومات من قاعدة بيانات او الملف، وهو <b>shadow</b> والخاص بكلمات المرور، حول المستخدم <b>student</b> .
<b>last</b>	<b>last</b> لمعرفة ما إذا كان المستخدم قد قام بتسجيل دخول في النظام، وإذا لم يكتب اسم المستخدم فسيُنفذ الأمر على المستخدم الحالي للنظام.
<b>last student</b>	<b>last</b> لمعرفة ما إذا كان المستخدم قد قام بتسجيل دخول في النظام، ثم اسم المستخدم وهو <b>student</b> .
<b>lastb</b>	<b>lastb</b> يُظهر محاولات تسجيل الدخول الى النظام حتى الفاشلة.
<b>usermod -L student</b>	<b>usermod</b> لتعديل على مستخدم، <b>-L</b> يستعمل لحجب أو قفل الحساب وبالتالي لا يستطيع المستخدم الدخول على النظام؛ وأخيراً اسم المستخدم <b>student</b> . ويمكن فك الحجب بنفس الأمر السابق ولكن مع الخيار <b>-U</b> كما يلي.
<b>usermod -U student</b>	<b>usermod</b> أمر تعديل على مستخدم، <b>-U</b> لفك الحجب وبالتالي يستطيع المستخدم الدخول إلى النظام، وأخيراً المستخدم <b>student</b> .
<b>userdel student</b>	<b>userdel</b> لحذف حساب مستخدم، ثم اسم المستخدم الذي تريد حذفه وهو في هذا المثال <b>student</b> .
<b>userdel -r student</b>	<b>userdel</b> لحذف حساب مستخدم، <b>-r</b> لحذف الدليل الرئيسي والبريد للمستخدم، ثم اسم المستخدم الذي تريد حذفه <b>student</b> .

## الملحق رقم (١٢)

ملخص لأوامر المعمل الخامس عشر:

Command -option string	المعنى
<b>cd /tmp</b>	<b>cd</b> أمر انتقال الى مسار ما، وهو الدليل /tmp .
<b>mkdir priv-dir pub-dir</b>	<b>mkdir</b> لإنشاء دليل، و <b>priv-dir</b> هو اسم الدليل الأول، و <b>pub-dir</b> هو اسم دليل ثاني. ملاحظة: يستطيع الأمر <b>mkdir</b> انشاء عدد من الدلائل بخطوة واحدة بشرط وجود مسافة بين أسمائهم.
<b>touch pub-dir/pub-file</b>	أمر <b>touch</b> له وظيفتين، الأولى تحديث ملف موجود بهذا الاسم، أو إنشائه إذا لم يجد الملف، فيتم هنا إنشاء ملف باسم <b>pub-file</b> داخل الدليل <b>pub-dir</b> .
<b>ls -l pub-dir</b>	<b>ls</b> عرض قائمة بالملفات والدلائل، و <b>-l</b> لإعطاء التفاصيل، للدليل <b>pub-dir</b> ، لاحظ أن الاستعراض من المكان الحالي ولا نحتاج الى الانتقال إلى ذلك الكائن.
<b>chmod o-rx priv-dir/</b>	<b>chmod</b> لتغيير الاذونات، <b>o</b> للآخرين، - إزالة إذن، <b>r</b> القراءة و <b>x</b> التنفيذ، على الملف <b>priv-dir</b> ، أما علامة / والتي بعد اسم الدليل فهي اختيارية للدلالة على أنه دليل.
<b>ls -ld pub-dir/</b>	<b>ls</b> عرض قائمة بالملفات والدلائل، و <b>-l</b> لإعطاء التفاصيل، و <b>d</b> لعرض تفاصيل الدليل نفسه وليس ما بداخله، للدليل <b>pub-dir</b> ، أما / والتي بعد اسم الدليل فهي اختيارية للدلالة على أنه دليل.

<b>chmod o+w pub-dir/</b>	<b>chmod</b> لتغيير الاذونات، <b>o</b> للآخرين، <b>+</b> إعطاء إذن، <b>w</b> الكتابة، على الملف <b>pub-dir</b> ، أما علامة <b>/</b> والتي بعد اسم الدليل فهي اختيارية للدلالة على أنه دليل.
<b>chmod g-rw,o-r priv-dir/priv-file</b>	<b>chmod</b> لتغيير الاذونات، <b>g</b> للمجموعة، - إزالة إذن، <b>r</b> القراءة، و <b>w</b> الكتابة، و علامة <b>"</b> , <b>"</b> لإضافة اذونات جديدة، و <b>o</b> للآخرين، - إزالة إذن، <b>r</b> القراءة، على الملف <b>priv-dir</b> .
<b>ls -l pub-dir/pub-file</b>	<b>ls</b> عرض قائمة بالملفات والدلائل، و <b>-l</b> لإعطاء التفاصيل، للملف <b>pub-file</b> ، والموجود في المسار <b>pub-dir/pub-file</b> .
<b>chmod a=rw pub-dir/pub-file</b>	<b>chmod</b> لتغيير الاذونات، <b>a</b> لكافة المستخدمين ( <b>o,g,u</b> )، = إعطاء إذن، <b>r</b> القراءة، و <b>w</b> الكتابة، على الملف <b>pub-dir</b> ، والموجود في المسار <b>pub-dir/pub-file</b> . لاحظ سيتم حذف الإذن الذي لم يُذكر (التنفيذ <b>x</b> ) من الجميع.
<b>chmod u=x pub-dir/pub-file</b>	<b>chmod</b> لتغيير الاذونات، <b>u</b> للمالك، = إعطاء إذن، <b>x</b> التنفيذ، على الملف <b>pub-dir</b> ، والموجود في المسار <b>pub-dir/pub-file</b> . لاحظ سيتم حذف الأذونات الذي لم تُذكر ( <b>w</b> ، <b>r</b> ) من المالك فقط، وبدون أي تغيير على البقية (المجموعة، الآخرين).
<b>echo "date" &gt; test.sh</b>	<b>echo</b> يقوم بالإخراج على الشاشة، للأمر <b>"date"</b> ، وعلامة التوجيه <b>&gt;</b> تعني إعادة توجيه الأمر إلى الملف <b>test.sh</b> .
<b>./test.sh</b>	<b>./</b> أمر تشغيل، للملف <b>test.sh</b> .
<b>chmod u+x test.sh</b>	<b>chmod</b> لتغيير الاذونات، <b>u</b> للمالك، <b>+</b> إعطاء إذن، <b>x</b> التنفيذ، على الملف <b>test.sh</b> .
<b>stat test.sh</b>	<b>stat</b> أمر عرض القيمة الثمانية للأذونات، للملف <b>test.sh</b> .

<p><b>chmod 775 test.sh</b></p>	<p><b>chmod</b> لتغيير الاذونات، والأرقام هي القيمة الثمانية للأذونات، حيث أن أول ٧ خاصة بالمالك وتعني قراءة وكتابة وتنفيذ (٤+٢+١) وثاني ٧ خاصة بالمجموعة وتعني قراءة وكتابة وتنفيذ (٤+٢+١) و ٥ خاصة بالآخرين وتعني قراءة وتنفيذ (٤+١) للملف <b>test.sh</b></p>
<p><b>su -</b></p>	<p><b>su</b> أمر تبديل المستخدمين وذلك بكتابة اسم المستخدم بعد - ، وإذا لم يتم كتابة اسم بعد - فهذا يعني الانتقال الى حساب <b>root</b>.</p>
<p><b>chown root:root pub-dir</b></p>	<p><b>chown</b> لتغيير المالك والمجموعة للكائن، إلى المستخدم <b>root</b> و مجموعة <b>root</b> للملف <b>pub-dir</b></p>
<p><b>chown bin pub-dir/pub-file</b> <b>ls -l pub-dir/pub-file</b></p>	<p><b>chown</b> تغيير المالك والمجموعة ، إلى المستخدم <b>bin</b>، (وحيث أنه لم يتم ذكر المجموعة فلن يتم تغييرها) للملف <b>pub-file</b>.</p>
<p><b>chgrp -R users priv-dir</b></p>	<p><b>chgrp</b> لتغيير المجموعة فقط ، <b>-R</b> تعني بشكل متكرر وبالتالي سيتم تغيير المجموعة للدليل ولكل الملفات والأدلة الفرعية التي بداخله، إلى المجموعة <b>users</b>، للدليل <b>priv-dir</b></p>

## الملحق رقم (١٣)

ملخص لأوامر المعمل السادس عشر:

Command -option string	المعنى
<b>ls -ld /tmp</b>	<b>ls</b> عرض قائمة بالملفات والدلائل، و <b>-l</b> لإعطاء التفاصيل، و <b>d</b> لعرض تفاصيل الدليل نفسه وليس ما بداخله، للدليل <b>/tmp</b> . لاحظ وجود <b>t</b> في الأذونات وهي إذن البت اللاصق <b>sticky bit</b> ، والذي يتيح الإضافة والتعديل ولكن الحذف خاص بمُنشئ الملف.
<b>ls -ld /var/tmp</b>	<b>ls</b> عرض قائمة بالملفات والدلائل، و <b>-l</b> لإعطاء التفاصيل، و <b>d</b> لعرض تفاصيل الدليل نفسه وليس ما بداخله، للدليل <b>/var/tmp</b> . لاحظ وجود <b>t</b> في الأذونات وهي إذن البت اللاصق <b>sticky bit</b> ، والذي يتيح الإضافة والتعديل ولكن الحذف خاص بمُنشئ الملف.
<b>ls -l /etc/shadow</b>	<b>ls</b> عرض قائمة بالملفات والدلائل، و <b>-l</b> لإعطاء التفاصيل، للملف <b>shadow</b> (ويوجد به كلمات السر المشفرة للمستخدمين).
<b>ls -l /usr/bin/passwd</b>	<b>ls</b> عرض قائمة بالملفات والدلائل، و <b>-l</b> لإعطاء التفاصيل، للأمر <b>passwd</b> . (لاحظ وجود حرف <b>t</b> في الأذونات وهو إذن <b>setuid</b> ، وبالتالي سيتم تنفيذ الأمر <b>passwd</b> من قبل المستخدم الفعلي المُشغل للأمر وكأنه المالك الحقيقي (المستخدم الجذر).
<b>ls -l /usr/bin/wall</b>	<b>ls</b> عرض قائمة بالملفات والدلائل، و <b>-l</b> لإعطاء التفاصيل، للأمر <b>wall</b> . (لاحظ وجود حرف <b>x</b> في الأذونات وهو إذن <b>setgid</b> ، لذلك فإنه سَيُنفذ من قبل أعضاء المجموعة المُشغلين للأمر وكأنهم مجموعة المالكين الحقيقيين (tty).

<b>echo "data" &gt; source</b>	<b>echo</b> يقوم بالإخراج على الشاشة، للكلمة "data"، وعلامة التوجيه > تعني إعادة توجيه الأمر إلى الملف <b>source</b>
<b>ls -li source</b>	<b>ls</b> عرض قائمة بالملفات والدلائل، و <b>-l</b> لإعطاء التفاصيل، و <b>i</b> لعرض رقم <b>inode</b> ، للملف <b>source</b> .
<b>ln source hardlink</b>	<b>ln</b> لإنشاء ارتباط صلب، بين الملف <b>source</b> و <b>hardlink</b> . لاحظ: الارتباط الصلب فقط بين الملفات.
<b>ls -li source hardlink</b>	<b>ls</b> عرض قائمة بالملفات والدلائل، و <b>-l</b> لإعطاء التفاصيل، و <b>i</b> لعرض رقم <b>inode</b> ، للملفين <b>source</b> و <b>hardlink</b> .
<b>ln hardlink hardtoo</b>	<b>ln</b> لإنشاء ارتباط صلب، بين الملف <b>hardlink</b> و <b>hardtoo</b> .
<b>ls -li hardlink hardtoo source</b>	<b>ls</b> عرض قائمة بالملفات والدلائل، و <b>-l</b> لإعطاء التفاصيل، و <b>i</b> لعرض رقم <b>inode</b> ، للملفات <b>hardlink</b> ، <b>hardtoo</b> ، <b>source</b> .
<b>rm hardtoo</b>	<b>rm</b> أمر حذف، للارتباط <b>hardtoo</b> .
<b>ln -s source softlink</b>	<b>ln</b> لإنشاء ارتباط صلب، <b>-s</b> لجعله ارتباط رمزي بين الملف <b>source</b> والملف <b>softlink</b> .
<b>ln -s /proc crossdir</b> <b>ls -l crossdir</b>	<b>ln</b> لإنشاء ارتباط صلب، <b>-s</b> لجعله ارتباط رمزي بين الدليل <b>/proc</b> والملف <b>crossdir</b> . لاحظ: الارتباط الرمزي يكون مع الملفات وأيضاً الدلائل.

## المراجع

n.d. Web. 25 Mar. 2016. "NDG Linux Essentials." *Cisco Networking Academy*. N.p.

<<https://1299123.netacad.com/courses/362135>>.



المؤسسة العامة للتدريب التقني والمهني  
Technical and Vocational Training Corporation

الكلية التقنية بمكة المكرمة  
College of Technology at Makkah