

Written Sp The-o-o-Ne

بسم الله الرحمن الرحيم

نقاط الدرس:

- استخدام الأمر (cat) لإستعراض الملفات النصية.
- استخدام الأمر (cat) كمحرر نصوص.
- تكلمة لإستخدام بعض الحروف الوصفية:
 ١. إعادة توجيه الخرج بإستخدام (>).
 - إستخدامها مع الأمر (cat).
 - إستخدامها مع الأمر (ls).
 ٢. إعادة توجيه الدخل بإستخدام (<).
 ٣. إعادة توجيه الخطأ إلى ملف ما بإستخدام (>2).
 ٤. إعادة توجيه الخرج إلى آخر الملف بإستخدام (>>).

• استخدام الأمر (cat) لإستعراض الملفات النصية:

يستخدم الأمر (cat) عادة في أنظمة (*.nix) على أنه يستعرض محتويات الملفات النصية أو الملفات (text) بمعنى عام ، كما ويستخدم هذا الأمر أيضاً في بعض الأحيان على أنه محرر نصوص كما سنتعلم في هذا الدرس كيفية إستخدام هذا الأمر المفيد حقاً ، كما أن هذا الأمر غالباً ما يستخدم لإستعراض الملفات الصغيرة الحجم ذات الأسطر القليلة والسبب في ذلك أن الأمر (cat) يقوم بعرض الملف كاملاً على الشاشة لذلك إن كان الملف عدد أسطره كثيرة جداً ولا تستوعب الشاشة جميع محتوياته فإنه فقط سوف يعرض آخر أسطر في الملف على مدى إستيعاب شاشتك.

والصورة العامة لهذا الأمر هو :

<code>bash-2.05\$ cat -[option(s)] [argument(s)]</code>

أي أن لهذا الأمر عند إستخدام يمكنك أن تضيف بعض الخيارات له التي تتناسب مع حاجتك (لن نناقش هنا أي خيار لهذا الأمر وذلك لقلّة خياراته والتي لم أرى أحداً يستخدمها من قبل ولكن للإستزادة يمكنك إستخدام ملفات المساعدة في النظام بإستخدام الأمر ((man cat)). في الصورة السابقة عند إستخدامنا للأمر (cat) فإنه لا بد لنا أن نقوم بذكر اسم الملف (argument) الذي نريد أن نستعرض محتوياته ولكن يمكنك أيضاً أن لا تذكر اسمه وذلك لأن الكلمة (argument) في الصورة السابقة تكون إختيارية كما تعرفنا سابقاً عن كيفية معرفة وفهم سطر الأمر. ولكن انتبه أنه وعند إستخدامك لهذا الأمر من دون اسم الملف سوف يقوم

هذا الأمر بتنشغيل نفسه على أنه محرر نصوص كما ستلاحظ في هذا الدرس سيتم مناقشة ذلك.

ولكي يتم فهم هذا الأمر جيداً لو أردنا أن نقوم مثلاً بإستعراض محتويات الملف (file1.txt) والذي أنشأناه سابقاً إلى حسابك في الإنترنت. فإن الأمر سيكون على الصورة التالية:

```
bash-2.05$ cat file1.txt
```

ولاحظ أنه لتنفيذ الأمر السابق لابد أن تكون في نفس الدليل الموجود فيه الملف أو كتابة مسار الملف كاملاً. في المثال السابق سيظهر لك محتويات الملف الذي قمنا بإستعراضه وتلاحظ أنه عرض جميع النص وذلك لقلة عدد الأسطر. ويمكنك أن تعمل مع جميع الملفات ذات الأسطر القليلة بنفس الطريقة المذكورة في المثال.

• إستخدام الأمر (cat) كمحرر نصوص:

كما قلنا سابقاً أن هذا الأمر مفيد جداً بصراحة (من وجهة نظري J) لذلك فإن من خدمات هذا الأمر أيضاً هو إستخدامه كمحرر للنصوص ويمكنك ذلك من خلال كتابة الأمر بمفرده فقط على سطر الأوامر على الصورة التالية:

```
bash-2.05$ cat
```

عن كتابة السابق فإنه سوف يتم فقط الإنتقال إلى السطر التالي وستجد أن المؤشر يومض ومعنى ذلك أنه يريد منك أن تدخل له ما شئت من الكلام بعد كتابته على الشاشة ثم بعد الإنتهاء قم بالضغط على حرف التحكم (^D) (سبق وأن تم شرحه سابقاً). وتعني (^D) للملفات أنها نهاية الملف (EOF). أي أنه بعد إنتهائك من كتابة ما تريد وضغطت على الحرف (^D) فإن ذلك إيعاز أنك قد انتهيت من الكتابة إلى الملف ولاحظ أنه بعد ضغطك على (^D) سوف يعود لك سطر الأوامر مرة أخرى.

في المثال السابق عند إستخدامنا للأمر (cat) لتحرير نص وقمنا بالتحرير ولكن النتيجة أنه لم يتكون لدينا أي ملف في دليلنا الحالي والسبب ببساطة هو أننا لم نقم بتحديد اسم للملف الذي نريد أن نكتب إليه وهذا ما يدعى ب (إعادة توجيه الخرج) أي أننا نقوم بتوجيه النص الذي نكتبه إلى الملف الذي نحدده. وتمكننا عملية إعادة توجيه الخرج إذا ما تم إستخدامها مع الأمر (cat) إلى إنشاء ملف جديد بالنص الذي نكتب J.

١. إعادة توجيه الخرج بإستخدام (>):

كما ذكرنا سابقاً أنه في أنظمة (*.nix) بعض الحروف التي تعني إلى عملية معينة بالنسبة للغلاف. ومن هذه الأحرف هي الحرف (>) وهذا الحرف يمكننا من إعادة توجيه الخرج بدلاً عن وحدة الخرج الرئيسية وهي الشاشة لدى المستخدم

إلى مكان آخر كملف مثلاً. ويمكن أن تستخدم هذه العلامات مع العديد مع الأوامر سنناقش في هذا الدرس الأمر (cat) والأمر (ls).

١. استخدام عمليات إعادة توجيه الخرج (>) مع الأمر (cat):

عندما نقوم باستخدام هذا الحرف مع الأمر (cat) فإنه في هذه الحالة يعمل هذا الأمر كمحرر نصوص كما ذكرنا سابقاً. حيث أنه وكما قلنا عند استخدام الأمر (cat) فقط من دون أي اسم سوف يكون محرر ولكن لن يكون هناك أي ملف ناتج عن العملية وذلك لأننا لم نحدد دليل نوجه إليه الخرج. لذلك فإن الحرف (>) يستخدم مع الأمر (cat) على الصورة العامة التالية:

```
bash-2.05$ cat > filename
```

حيث أننا نعني ب (filename) اسم الملف الذي تريد أن يخرج إليه النص الذي قمت بكتابته. ولكي يتم فهم العملية بشكل أكبر لنفترض أننا نريد أن نكتب النص الموجود بالأسفل أدناه في ملف بالاسم (theone):

```
Hello guys, I am learning Unix now
And I am so happy with it.
```

لذلك عندما نريد كتابة النص السابق في ملف باستخدام الأمر (cat) فإننا سنطبقه على الصورة التالية:

```
bash-2.05$ cat > theone
```

```
— hello guys, I am learning Unix now
and I am so happy with it
^D
```

لاحظ أنه بعد تطبيقنا للأمر (cat >theone) اختفى سطر الأمر وظهرت علامة (—) الظاهرة باللون الأحمر وستجده يومض، ومعنى ذلك أن الغلاف في هذه الحالة ينتظر ويتوقع منك مدخلات لذلك قمنا بكتابة النص الذي أردناه ثم بعد أن انتهينا لاحظ حرف التحكم (^D) في الحقيقة هو لا يظهر ولكني كتبت فقط لكي يتضح لك أنه لابد عند الإنتهاء من كتابة النص الذي تريده لابد من كتابة هذا الحرف. الآن وبعد أن انتهيت من كتابة النص الذي تريده حاول أن تستعرض محتويات دليلك الحالي وستجد ملف بالاسم (theone) استعرض محتوياته بالأمر (cat) على الصورة :

```
bash-2.05$ cat theone
hello guys, I am learning Unix now
and I am so happy with it
bash-2.05$
```

لاحظ في المثال السابق قد تم إستعراض محتويات الملف ثم عاد لك مؤشر الغلاف مرة أخرى.

٢. إستخدام علامات إعادة توجيه الخرج (>) مع الأمر (ls):

يمكننا إستخدام هذه الطريقة في أحيان كثيرة منها لو أردنا أن نحتفظ بخرج محتويات دليل ما لرؤيته لاحقاً أو حتى لحفظه، أو يمكن أيضاً أن نوجه خرج الأمر (ls) إلى ملف ومن خلال الناتج الموجود في الملف نبحث عن كلمة معينة في هذا الخرج إستخدامات كثيرة يمكنك أن تستعمل مخيلتك لتطويع الأوامر تحت أمرتك **JJ**.

والصورة العامة للأمر (ls) مع علامات التوجيه هو:

```
bash-2.05$ ls >filename
```

حيث أن (filename) هو عبارة عن الملف الذي تريد توجيه الخرج إليه فلو مثلاً أردنا توجيه خرج إستعراض محتوى الدليل (etc) فإننا سنكتب الأمر التالي:

```
bash-2.05$ ls /etc >result
bash-2.05$
```

في المثال السابق تلاحظ أنه لم يظهر أي ناتج على الشاشة ولكن عاد مؤشر الغلاف لك . وفي هذه الحالة يمكنك إستخدام الأمر (cat) لرؤية المحتوى للملف (result) ، أعتقد أنك لم ترى جميع محتويات الملف وذلك لأنه وكما قلنا أن الأمر (cat) يفضل إستخدامه مع الملفات ذات الأسطر القليلة ولكن سنعرف الحل بإذن الله تعالى في الدرس القادم.

ملاحظة: لاحظ أنك إذا قمت بتوجيه الخرج إلى ملف موجود في الدليل الحالي فإنه سوف يسمح لجميع محتوى الملف وستكتب النتائج الجديدة. وبعد قليل سنعرف كيف نتغلب على هذه المشكلة.

١. إعادة توجيه الدخل بإستخدام (<):

وهذه العلامة قليل ما أجد من يستخدمها ولكن مع إستخدام مخيلتك اعتقد أنك ستجد لها أكثر من إستخدام في بالك. فمثلاً لو قمنا بإنشاء ملف في دليلنا الحالي بالاسم (one) ووضعنا في هذا الملف اسم الدليل (/etc) بنفس الصورة (لاحظ أنك لا بد من أن تستخدم الأمر (cat) لكي تقوم بكتابة اسم الدليل في الملف كما شرحنا سابقاً). بعد ذلك إستخدم الأمر (ls) على الصورة :

```
bash-2.05$ ls <one
```

في هذه الحالة ستجد أنه قد تم إستعراض محتويات الدليل (/etc) دون كتابته وذلك لأننا تخلينا عن لوحة المفاتيح كوحدة دخل أساسية للغلاف بالملف المكتوب فيه اسم الملف **JJ**.

١. إعادة توجيه الخطأ إلى ملف ما بإستخدام (>2):

هل سبق وأن واجهتك الجملة التالية عند إستخدامك لأمر ما بصورة خاطئة:

```
bash-2.05$ LS
```

```
bash: LS: command not found
```

أو حتى واجهتك أي رسالة خطأ أخرى من الغلاف . تفيد العلامة (>2) في أنها تقوم بتوجيه رسائل الخطأ بدلاً من أن تظهر على الشاشة إلى ملف ما تقوم أنت بتحديدته وذلك لأي غرض يخصك مثلاً لكي ترى بعد فترة من عملك على الغلاف ماهي الأخطاء الذي قمت بإرتكابها إنشاء إستخدامك للنظام وهكذا. والصورة العامة لهذا العلامة هي:

```
bash-2.05$ command 2> errorfile
```

حيث أن (command) هو الأمر الذي ستقوم بتنفيذه و (errorfile) هو الملف الذي ستوجه إليه رسائل الخطأ ولاحظ أنه يمكنك تغييره كيفما تشاء. لكي تتضح الفكرة بشكل أكبر. نعرف أن أمر إستعراض محتويات دليل ما هو (ls) ونعرف أيضاً أن نظام اليونكس يعتبر حساس بالنسبة لحالة الحروف لذلك لو قمنا بإستخدام الأمر (LS) بدلاً من إستخدام (ls) فإنه من المؤكد أن الغلاف سيرد عليك برسالة خطأ أن هذا الأمر غير صحيح. لذلك لو أردنا أن نوجه رسالة الخطأ هذه إلى الملف (error) في دليلنا الحالي فإننا سنقوم بالتالي:

```
bash-2.05$ LS 2> error
bash-2.05$
```

لاحظ أنه لم يظهر لك أي رسالة خطأ بالرغم من أنه من المفترض أن تظهر لك رسالة خطأ تعلمك بعدم وجود هذا الأمر. ولكننا قمنا بتوجيه رسالة الخطأ هذه إلى الملف (error) . حاول أن تلقى نظرة على محتوى الملف (error).

١. إعادة توجيه الخرج إلى آخر الملف بإستخدام (>>):

عندما تكلمنا عن إعادة توجيه الخرج لملف فإننا قلنا أنه في حال وجود الملف في نفس الدليل فإنه سوف يتم فقد محتويات هذا الملف ولذلك فهذه تعتبر مشكلة لذلك وجدت العلامة (>>) وهي تقوم أيضاً بعمل إعادة توجيه الخرج إلى ملف ولكن الفرق بينها وبين (>) أنها تقوم بإلحاق النص الذي قمت بإعادة توجيهه إلى آخر الملف وبالتالي لن يتم فقدان محتويات أي ملف عندك. ويوجد طريقة أستخدامها أحياناً وهي أن لو كنت أمتلك ملفين وأريد أن أدمجها معاً فإنني أقوم بدمجها بإستخدام هذه العلامة. مثلاً لدينا في دليلنا الحالي الملف (file1.txt) صحيح؟! وأيضاً لدينا الملف (theone) ونريد أن ندمج الملف (theone) بنهاية الملف (file1.txt) ونضع الناتج في الملف (file1.txt) ستكون الطريقة في هذه الحالة هي :

```
bash-2.05$ cat theone >>file1.txt
```

الآن حاول ان تستعرض محتويات الملف (file1.txt) وستجد أنه قد تم دمج الملفين سوياً.

وتقبلوا خالص تحيات أخوكم المحب (The-oNe)
الرجاء إرسال مقترحاتكم وآرائكم على العناوين التالية:

The-one@pharaonics.net

OR

The_o0ne@hotmail.com

OR

The_o0one@yahoo.com