

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ
الْحَمْدُ لِلَّهِ الَّذِي هَدَانَا لِهَذَا وَمَا كُنَّا لِنَشْكُرَهُ
إِلَّا بِرَحْمَتِهِ الْعَظِيمِ

نظام الملفات

وَقُلْ رَبِّي زَيْنِي عِلْمًا



أنواع الملفات الشائعة

الملفات يمكن أن تحتوي على عدة أنواع من المعلومات. مثل:

بيانات المستخدم:

هي الملفات التي تحتوي على البيانات التي قمت أنت بإنشائها وقمت بتطويرها. فيمكن مثلا أن تقوم بإنشاء ملف يحتوي على نصوص أو أرقام.

فيما بعد:

سوف تعرف في هذا الفصل كيف تنشئ ملفا وتضع فيها ما تريد من النصوص والأرقام إن شاء الله.

ويمكن أن يكون الملف معقد ولا تستطيع قراءته هكذا وقد ترى فيه رموز وحروف غريبة الشكل، مثل هذه الملفات يتم تكوينها من خلال برامج خاصة وتحتاج لهذه البرامج أيضا حتى تستطيع قراءة هذا الملف. مثلا: نفرض أنك قمت ومن خلال برنامج رسم برسم شكل معين فلو قمت باستخدام الأمر الذي يعرض ما بداخل الملف وهو الأمر more أو less ستجد أن الناتج عبارة عن رموز وأشكال غريبة ليست كما أنت رسمتها من قبل أما لو قمت باستعراض هذا الملف من خلال هذا البرنامج الخاص بالرسم فإنه سيظهر لك الرسم كما رسمته أنت تماما.

بيانات النظام:

حيث أن هذه الملفات تحتوي على رموز وأرقام لا يفهمها إلا الكمبيوتر، بحيث أنها تتكون من لغة Assembly أو لغة الآلة.

أسماء الملفات

أسماء الملفات يمكن أن يصل طولها إلى 256 حرف، حيث يمكن أن تحتوي على حروف كبيرة أ، صغيرة وأيضا يمكن أن تحتوي على تلك الرموز () () () () ويمكن أيضا أن يبدأ به الاسم، ويوجد هناك رموز لا يمكن أن يبدأ بهم الاسم مثل () () () () لأنهم محجوزين من النظام حيث أن هذه الرموز يستخدمها النظام.

انتبه:

تذكر أننا استخدمنا تلك العلامات () () () () عندما تعاملنا مع النظام، وبالتالي لا يمكن أن نستخدمهم مع أسماء الملفات.

مقدمة عن الفهارس

الدليل الأصلي والدليل الفرعي

الفهرس Root

في نظام لينوكس يعتبر الفهرس root هو المالك الرئيسي للفهارس الأخرى، وهذا يعتبر الفهرس الأعلى وكل الفهارس تكون تحت الفهرس root أي أنها متفرعة منها. ولو تريد تشبيه لنظام الملفات في اللينوكس فيمكن أن تشبهه بالشجرة حيث أن الفهرس root هو الجذر وباقي الفهارس هي الفروع والأغصان.

أسماء الفهارس

كيفية تسمية الفهرس مثل تسمية الملف تماما،

الفهرس Home

يقدم لينوكس لكل مستخدم من المستخدمين فهرس خاص به يدعى هذا الفهرس Home. وداخل هذا الفهرس يستطيع المستخدم أن يخزن الملفات والفهارس الخاصة به ويستطيع أيضا أن ينشئ فهارس فرعية. وللمستخدم له الأحقية الكاملة في التعامل مع ملفاته الخاصة به في فهرس Home الخاص به، لأنه لا يوجد به ملفات خاص بالنظام أو خاصة بالمستخدمين الآخرين.

ملاحظة:

أنت لا تستطيع أن تغير مكان الفهرس Home الذي حدده لينوكس حتى يكون كل شيء مرتب ولكي يحافظ على أمن النظام.

الأمر pwd

هذا الأمر يحدد لك الفهرس الذي تعمل من خلاله أي أنه يحدد موقعك تماما.

فلنفرض مثلا أن اسمك الذي دخلت به النظام هو aly دخلت داخل الفهرس aly المتفرع من الفهرس home وأنت داخل الفهرس aly قمت بكتابة هذا الأمر pwd فسترى تلك النتيجة:

```
darkstar:~$ pwd
```

```
/home/aly
```

```
darkstar:~$
```

ملاحظة:

عندما تدخل إلى نظام لينوكس فإنه يضعك على الفهرس الخاص بك

ملاحظة:

الأمر pwd اختصار لجملة print working directory ، ومعنى كلمة working directory معناها الفهرس العامل ومعنى هذا أن أي أمر ستكتبه سيتم تنفيذه على الفهرس العامل فمثلا لو كتبت الأمر ls فسيعرض الملفات والفهارس التي توجد في الفهرس الذي تعمل من خلاله وهكذا مع باقي الأوامر الأخرى.

الاسم الحقيقي والنسبي للملف

عندما تحدد اسم ملف ما لكي تعامل معه من خلال أي أمر مثل more مثلا فإن لينوكس سيبحث عن هذا الأمر في الفهرس الحالي الذي توجد عليه فمثلا لو كنت في الفهرس /home/aly ثم قمت بكتابة الأمر التالي more test فإن الأمر سيبحث عن هذا الملف في الفهرس الحالي وهو aly وإذا لم يوجد هذا الملف في هذا الفهرس فلم ينفذ الأمر more شيئا، ولنفرض أيضا أنه يوجد لديك فهرس آخر متفرع من الفهرس aly واسمه مثلا batman وهذا الفهرس يحتوي على الملف test ثم قمت بكتابة الأمر التالي وأنت ما زلت موجودا في الفهرس aly more batman/test فإن الأمر more سيعرض لك محتويات الملف test الذي يوجد في الفهرس batman. أما في حالة أنك غيرت مكان تواجدك وذهبت إلى الفهرس الأصلي root وأردت أن تستعرض محتويات الملف test الموجود في الفهرس batman السابق ذكره فلا بد من كتابة الأمر التالي

```
more /home/batman/test
```

ولكي نفهم معنى الاسم الحقيقي والنسبي للملف إليك هذا المثال:

نفرض أننا لدينا ملف اسمه test في الفهرس batman تحت الفهرس aly تحت الفهرس Home ونفرض أيضا أنه لدينا ملف اسمه أيضا test وأيضا تحت الفهرس batman ولكنه تحت الفهرس omr تحت الفهرس Home .

فإن هذا الاسم batman/test يعتبر اسم نسبي لأنه مشترك بين فهرسين أما الاسمين

/home/aly/batman/test

/home/omr/batman/test

فهما اسمين حقيقيين.

الأمر cd

نستخدم هذا الأمر لتغيير الفهرس الذي نعمل عليه (working directory) ولنفرض مثلا أنك تريد أن تتحول إلى فهرس آخر فما عليك إلا أن تستخدم النموذج التالي

cd <directory specification>

حيث أنك ستستبدل <directory specification> باسم الفهرس الذي تريد أن تعمل عليه .

ملاحظة :

يجب أن يكون هناك مسافة بين الأمر cd واسم الفهرس .

لنفرض الآن في الفهرس الخاص بك وقمت بكتابة الأمر التالي cd .. ف'ن لينوكس سوف ينقلك إلى الفهرس Home وإذا قمت بكتابة الأمر مرة أخرى فسينقلك إلى الفهرس الأصلي وإذا كتبت الأمر مرة أخرى فلن يحدث شيئا لأنه لا يوجد فهرس أعلى من الفهرس الأصلي root

darkstar:~\$ cd ..

darkstar:/home\$ pwd

/home

```
darkstar:/home$ cd ..
```

```
darkstar:/$ pwd
```

```
/
```

```
darkstar:/$ cd ..
```

```
darkstar:/$ pwd
```

```
/
```

وتستطيع أن تستخدم الاسم الحقيقي للفهرس مع الأمر cd كما يلي

```
darkstar:/$ cd /usr/bin
```

```
darkstar:/usr/bin$ pwd
```

```
/usr/bin
```

ملاحظة:

لاحظ أنك عندما قمت بتغيير الفهرس الذي تعمل عليه فإن لينوكس يغير المحث إلى اسم الفهرس الحقيقي الذي تعمل عليه.

كما في المثال السابق

```
darkstar:/$ cd /usr/bin
```

```
darkstar:/usr/bin$
```

لو كنت تعمل تحت أي فهرس مهما كان مكانه وكتبت الأمر cd فقط فإن لينوكس سيعيدك إلى فهرسك الخاص بك الذي يوجد تحت الفهرس Home. أما لو كتبت / cd فسيعيدك إلى الفهرس الأصلي.

ملاحظة:

لاحظ المحث التالي

darkstar:~\$

العلامة ~ معناها أنك في الفهرس الخاص بك ولا تحتاج عندما تريد إلى الانتقال إليه أن تكتب

~ cd ولكن اكتب فقط cd كما علمت منذ قليل.

ملاحظة:

لو كان لديك في النظام أكثر من مستخدم وأردت أن تذهب إلى الفهرس الخاص بمستخدم معين قم بكتابة الأمر التالي cd <username>~ ، لنفرض أنه يوجد هناك مستخدم اسمه aly وأردت أن تدخل إلى الفهرس الخاص به قم بكتابة الأمر التالي cd ~aly . وهكذا.

انتبه:

عندما تريد الذهاب إلى فهرس معين من خلال اسمه الحقيقي فخذ حذرك عندما تكتب أسماء الفهارس لأنه لو وجد أي خطأ في أي اسم فهرس يمكن أن لا يؤدي العمل الذي أنت تقصده، فمثلاً:

/usr/doc/faq/unix

وأيضاً لديك الفهارس التالية:

/usr/docs/faq/unix

لاحظ الفرق بينهما، فهما يختلفان في الحرف s الموجود في كلمة doc .

فيمكن تريد أن تحذف الملفات الموجودة في الفهرس UNIX التابع للفهرس docs ثم قمت بكتابة الفهرس doc بدلا من docs فإنه سيحذف ملفات أخرى غير تلك التي أنت تقصدها.

إنشاء ملفات وحذفها

سنتكلم الآن عن كيفية إنشاء الملفات وحذفها ويمكن أن تتم هذه العملية بأكثر من طريقة.

انتبه:
يجب أن تكون حذرا عند التعامل مع أوامر إنشاء وحذف الملف لأنك
ممكن أن تكتب على ملف موجود أو ممكن أن تحذف هذا الملف وقد
يكون هذا الملف ضروريا لك أو للنظام، ويستحسن أن تدخل
كمستخدم عادي وليس ك root إلا إذا كنت متأكد أن لن ترتكب خطأ
خلال قراءتك لهذا الفصل.

الآن انتقل إلى الفهرس الخاص ك في الفهرس Home من خلال الأمر cd وتأكد أنك على الفهرس الخاص بك من خلال الأمر pwd .

تذكر:
قمنا سابقا بإنشاء ملف من خلال > .

الأمر cat

يقوم الأمر cat بأخذ المدخلات من أي مصدر ثم يقوم بإخراجها إلى أي مصدر. والحالة القياسية لأمر cat أن يأخذ مدخلاته من لوحة المفاتيح ويخرجها على الشاشة.

قم بكتابة الأمر cat ثم اضغط enter ، فعند هذه الحالة فإن الأمر cat ينتظر منك أن تدخل أي شيء من خلال لوحة المفاتيح وبمجرد أنك ضغطت enter فإنه سيظهر لك ما كتبت على الشاشة.

```
darkstar:~$ cat
```

```
ASSLAMO ALYCOM
```

```
ASSLAMO ALYCOM
```

```
HAY
```

```
HAY
```

والآن إذا أردنا أن ندخل نصوص من خلال لوحة المفاتيح إلى ملف ما فما عليك إلا أن تتبع النموذج التالي:

```
darkstar:~$ cat > newfile
```

Hello world

Here's some text

حيث newfile هو اسم الملف الذي تريد إنشائه وتضع فيه النص الذي كتبتة. وعندما تنتهي قم بضغط (Ctrl-D) وسوف ترجع إلى محث لينوكس.

والآن وبعد أن قمت بإنشاء الملف طبعا تريد أن تقرأ محتوياته.

تذكر:
يمكنك أن تقرأ محتويات ملف ما من خلال الأمر more أو less .

وتستطيع أيضا أن تقرأ محتويات الملف من خلال الأمر cat أيضا كما يلي:

```
darkstar:~$ cat newfile
```

Hello world

Here's some text

```
darkstar:~$
```

حيث أن newfile هو اسم الملف الذي تريد أن تقرأ محتوياته.

تذكر:
قلت عندما بدأت بشرح الأمر cat أنه cat يأخذ المدخلات من أي مصدر ثم يقوم بإخراجها إلى أي مصدر وفي حالة قراءة الملف من خلال الأمر cat فإن الأمر cat يأخذ مدخلاته من الملف ويخرجها على الشاشة.

من الممكن أن يحدث بعد أن أنشأت الملف newfile أنك تريد أن تضيف إليه أشياء أخرى في هذه الحالة ستستخدم الرمز >> ، كما هو موضح:

```
darkstar:~$ cat >> newfile
```

Some more lines

هذا السطر تريد أن تضيفه في الملف

ثم بعد ذلك قم بضغط (Control-D) ، ثم قم بإظهار محتويات الملف باستخدام الأمر cat كما يلي:

```
darkstar:~$ cat newfile
```

```
Hello world
```

```
Here's some text
```

```
Some more lines
```

```
darkstar:~$
```

لاحظ أنه تم إضافة السطر Some more lines وحتى تعرف المقصود من الأمر cat راقب التالي:

سنقوم الآن بإنشاء ملف يدعى anotherfile ثم سنكتب فيه كما هو مبين وبعد ذلك نضغط (Control-D)
darkstar:~\$ cat > anotherfile

```
Different text
```

```
darkstar:~$
```

وبعد ذلك قمنا بوضع محتويات الملفين newfile anotherfile في الملف thirdfile وتأكد من ذلك بإظهار محتويات thirdfile ولاحظ النتيجة الرائعة!!!!!!

```
darkstar:~$ cat newfile anotherfile> thirdfile
```

```
darkstar:~$ cat thirdfile
```

Hello world

Here's some text

Some more lines

Different text

darkstar:~\$

ملاحظة:
يقوم الأمر cat بأخذ مدخل واحد أو عدة مدخلات ثم يخرجها جميعا
خلال مخرج واحد فقط.

إنشاء الفهارس

حتى تقوم بإنشاء فهرس استخدم الأمر التالي mkdir والشكل العام لهذا الأمر هو كالتالي:

mkdir<name>

حيث أن name هو اسم الفهرس الذي تريد إنشاؤه. وعندما تقوم بهذه العملية فإن لينوكس يقوم بإنشاء فهرس فرعي للفهرس الذي توجد أنت عليه عند كتابتك للأمر، كما يلي:

darkstar:~\$ ls

anotherfile newfile thirdfile

darkstar:~\$ mkdir newdir

darkstar:~\$ ls

```
anotherfile newdir/ newfile thirdfile
```

نسخ ونقل الملفات

عندما تريد أن تنسخ ملف ما قم بكتابة الأمر cp ، أما عندما تريد نقل ملف ما قم باستخدام الأمر mv. والشكل العام لاستخدام هذين الأمر هو الآتي:

```
mv <source> <destination>
```

```
cp <source> <destination>
```

حيث أن source هو الملف المصدر و destination هو الملف الذي تريد أن تنسخ أو تنقل إليه.

وهذا مثال عن استخدام الأمرين cp و mv .

```
darkstar:~$ ls
```

```
anotherfile newdir/ newfile thirdfile
```

```
darkstar:~$ mv anotherfile movedfile
```

```
darkstar:~$ ls
```

```
movedfile newdir/ newfile thirdfile
```

```
darkstar:~$ cp thirdfile xyz
```

```
darkstar:~$ ls
```

anotherfile newdir/ newfile thirdfile xyz

ملاحظة:

الفرق بين الأمر cp و mv هو أن الأمر cp يقوم بنسخ الملف إلى ملف آخر ولكن بنفس الاسم لو كانا الملفين في نفس الفهرس أما لو كانا في فهرسين مختلفين فيمكن أن تنسخ الملف باسمه، أما الأمر mv فأنت تقوم بإنشاء ملف جديد باسم جديد يحتوي على محتويات الملف الأصلي مع حذف الملف الأصلي لو كانا الاثنين في نفس الفهرس أما لو كانا في فهرسين مختلفين فيمكن أن تنقل الملف باسمه الأصلي.

يمكن أن تتم عملية نسخ ونقل الملفات من فهرس إلى فهرس آخر بعدة طرق منها:

```
darkstar:~$ mv newfile newdir/newfile
```

حيث أنك نا قمت بنقل الملف newfile من الفهرس الخاص بك إلى الفهرس newdir الموجود والمتفرع من الفهرس الخاص بك.

ويمكن أيضا أن تتم هكذا

```
darkstar:~$ mv newfile newdir
```

ولكن يجب أن يكون الفهرس newdir موجود مسبقا. أما لو لم يكن هذا الفهرس موجودا فسوف يقوم لينوكس بتغيير اسم الملف newfile إلى الاسم newdir . ويمكن أيضا أن تتم بالطريقة التالية:

```
darkstar:~$ cd newdir
```

```
darkstar:~newdir$ cp ../newfile .
```

حيث أنه في السطر الأول قمنا بالدخول إلى الفهرس newdir وفي السطر الثاني قلنا للنظام قم بنسخ الملف newfile من الفهرس الأب (أي الذي يعتبر أساس الفهرس newdir حيث أن الفهرس newdir متفرع من الفهرس الخاص بك) إلى الفهرس newdir .

يمكنك أن تغير اسم الملف إلى اسم آخر عند عملية نقله أو نسخه إلى فهرس آخر كالتالي

```
darkstar:~$ cp newfile newdir/anothername
```

حيث أننا هنا قمنا بنسخ الملف newfile من الفهرس الحالي إلى الفهرس newdir وغيرنا اسم الملف إلى anothername .

ملاحظة:
عملية النقل أسرع من عملية النسخ....واترك لك السؤال.. لماذا؟

النسخ والنقل باستخدام علامتي *,?

إذا أردت أن تنقل أو تنسخ عدد كبير من الملفات إلى مكان آخر فسوف يصبح الأمر مملا إذا فعلت عملية النقل أو النسخ مع كل ملف لذلك يوجد هناك حلا لهذه المشكلة وهو استخدام العلامتين *,?.

إذا أردت أن تنقل كل الملفات من الفهرس الحالي إلى فهرس آخر فقم باستخدام النموذج التالي:

```
darkstar:~$ cp * /foldername
```

حيث أن foldername هو اسم الفهرس الذي تريد أن تنقل الملفات إليه. ويمكن أيضا أن تستخدم الرمزين *,? كما تريد فمثلا لو أردت أن تنسخ كل الملفات التي تبدأ بالحرف a فقم بتعديل الأمر السابق إلى

```
darkstar:~$ cp a* /foldername
```

وهكذا.

أما لو أردت أن تنسخ الملفات التي يتكون اسمها من حرفين وتبدأ بالحرف a فقم بكتابة الأمر كما يلي:

```
darkstar:~$ cp a? /foldername
```

نقل الفهارس

تتم نقل الفهارس مثلما تتم نقل الملفات حيث أن الشكل العام للأمر هو كالاتي:

```
mv <directory> <destination>
```

قم بتنفيذ الأمر التالي:

```
darkstar:~$ mv newdir /tmp
```

```
darkstar:~$ cd /tmp
```

```
darkstar:/tmp$ ls
```

```
/newdir
```

وبعد تنفيذ هذا الأمر أصبح الفهرس newdir فهرس فرعي للفهرس /tmp .

ملاحظة:

عندما تقوم بنقل فهرس ما فإن كل الملفات التابعة له والفهارس الفرعية التي توجد معه تنقل معه إلى المكان الجديد.

حذف الملفات والفهارس

بعد أن علمت كيف تقوم بعملية إنشاء الملفات والفهارس ، آن الوقت الآن لتعرف عملية حذف الملفات والفهارس. لكي تقوم بعملية حذف الملف قم باستخدام الأمر rm (حيث أن rm اختصار لكلمة remove) حيث أن الشكل العام لهذا الأمر كالتالي:

rm <filename>

فمثلا لنفرض أنه يوجد ملف يدعي test وهذا الملف موجود في الفهرس الخاص بك، وفي حالة وجودك في الفهرس الخاص بك، فما عليك إلا أن تكتب الأمر التالي:

darkstar:~\$ rm test

أما لو كان هذا الملف في الفهرس /tmp قم بكتابة الأمر التالي:

darkstar:~\$ rm /tmp/test

ويمكنك أيضا أن تستخدم العلامتين *,? في عملية الحذف ولكن يجب أن تكون منتبها عند استخدام هاتين العلامتين.

فلو مثلا قمت كتابة الأمر التالي:

darkstar:~\$ rm *

فهذا معناها أنك قد قمت بحذف جميع الملفات التي توجد في الفهرس الخاص بك.

ماذا سيحدث لو كتبت الأمر التالي:

darkstar:~\$ rm *test

هذا يعني أنك سوف تقوم بحذف جميع الملفات التي تنتهي ب test.

حذف الفهارس

عادة ما يستخدم الأمر rmdir لحذف الفهارس ويكون الأمر بالشكل التالي:

```
rmdir <directory>
```

ولكن قبل أن تحذف أي فهرس يجب أن يكون خاليا ، وإلا سوف تظهر لك هذه الرسالة عندما تقوم بتنفيذ الأمر السابق:

```
rmdir: <directory>: Directory not empty
```

وهذا من باب التأمين لملفاتك.

ولكن ماذا لو أنك تريد حذف فهرس يحتوي على كثير من الفهارس الفرعية وكل فهرس فرعي لديه أيضا فهارس فرعية، طبعاً حتى تتمكن من حذف هذا الفهرس بالطريقة السابق ذكرها سيصبح الأمر صعب جداً، لذلك يوج حل أفضل وهو الأمر:

```
rm -r <directory>
```

فهذا الأمر سيحذف لك الفهرس بكل فهارسه الفرعية.