

# LINUX

سطر الأوامر

## نظرة عن قرب أكثر

أبو عبد الرحمن  
أبو عبد الرحمن



مجتمع لينوكس العربي

منتديات ترياق العرب

مكتبة الطارق الالكترونية

```
shuffl@P0w3R:~/Desktop$ mkdir test
shuffl@P0w3R:~/Desktop$ ls -l
drwxr-xr-x  8 shuffl shuffl  4096 2007-03-11 07:56 terminolgies
drwxr-xr-x  2 shuffl shuffl  4096 2007-04-17 09:32 test
-rw-r--r--  1 shuffl shuffl    0 2007-04-17 09:00 test.txt
```

```
shuffl@P0w3R:~$ ls -l~
```

```
shuffl shuffl  4096 2007-02-21 19:03 aaaa
ffl shuffl 1865301 2007-02-21 18:45 arp.html
ffl shuffl 31295 2006-11-27 10:07 arp.html.zip
```

```
shuffl@P0w3R:~/Desktop$ ls -l wireless.log
-rw-r--r--  1 shuffl shuffl 0 2007-04-16 18:17 wireless.log
shuffl@P0w3R:~/Desktop$ mkdir test
```

بسم الله الرحمن الرحيم

جميع الحقوق محفوظة للكاتب

أبو عبدالرحمن

في مجتمع لينوكس العربي

<http://www.linuxac.org>



تم إعداد وتصميم هذا الكتاب الإلكتروني

لـ مكتبة الطارق الإلكترونية

## كل ما يبحث عنه الهكرز المسلم ##

لتحميل المزيد من الكتب والملفات المفيدة في جميع مجالات الكمبيوتر والانترنت  
اذهب إلى موقع المكتبة على الرابط التالي

<http://www.t0010.com/books/index.php>

---

تصميم الكتاب الإلكتروني

---

aLTar3Q

---





الموضوع كامل تم نسخة من مجتمع لينوكس العربي  
وقد قمت بجمعه وتنسيقه على صيغة كتاب الكتروني  
ولم استأذن من كاتب الموضوع ولكن أعتقد انه لم يكتبه  
إلا لتعليم شباب المسلمين وعلى هذا الأساس جرى نشره  
بالنسبة لموضوع الكتاب فهو مهم جداً والمفروض أي هكر  
يعرف هذه الأوامر لأننا نتعامل مع سيرفرات  
وهذه السيرفرات على انظمة لينوكس  
اتمنى أن يكون موضوع مفيد للجميع  
مع العلم بأن الموضوع لم يكتمل من المصدر

www.geotipo.org



## :: مقدمة ::

**بعون** من الله وفضله سوف أقوم بتقديم نظرة تكاد تقترب من الشمولية لموضوع مهم وهو سطر الأوامر و الذي يعتبر خاصية من أهم خصائص لينوكس على الإطلاق . ولكي تلمس ذلك على سبيل المثال لو قمت بإدارة أحد السيرفرات حينئذ الخيار الأمثل أمامك هو سطر الأوامر لقوته ومرونته وسرعته في تنفيذ الأوامر بغض النظر عن المميزات الأخرى وتتمثل في الأمان وفى قلة عدد الثغرات المتاحة أمام المخربين.

وإن شاء الله الموضوع مقسم إلى قسمين الأول للموضوع ذاته وسيكون هنا والآخر لوضع الردود والاستفسارات على هذا الرابط :

<http://www.linuxac.org/forum/showthread.php?t=1603>

حتى يكون الموضوع مسلسل ويخرج إن شاء الله بصورة مشرفة .

في البداية حينما يتبادر إلى الأذهان سطر الأوامر تجد أفكار غريبة لدى الأشخاص بخصوص هذا الموضوع منها عمليات معقدة وأوامر صعبة لا يعرف كيفية استخدامها إلا المحترفين وإلخ من هذه الأفكار . قد يكون الموضوع من جهة صحيح ولكن بالفعل وعن تجربتي الشخصية لاستخدام اللينوكس لا يمكن الفرار من سطر الأوامر سواء كنت مبتدئ أو متوسط المستوى أو محترف.

فمثلا سطر الأوامر يُمكنكَ بمجرد كتابة أمر واحد أن تنفذ ما لا يمكن تنفيذه من خلال الواجهات الرسومية مهما كانت قوة الواجهة الرسومية المستخدمة سواء كانت **Gnome** أو **KDE** أو أين كانت الواجهة.

كانت هذه مقدمة بسيطة عن ماهية سطر الأوامر وما يمكن أن يقوم به من مهام ولكن ما ذكرناه كانت نبذة بسيطة .

سطر الأوامر بحر كبير من المعرفة يزداد فيه زادك بالعمل اليومي عليه ومعرفة مهمة ووظيفة كل أمر .

يوجد بعض الأنواع من الطرفية أو الشل أو سطر الأوامر مسميات لمعنى واحد الواجهة التى تقوم أنت بكتابة ما يحلو لك من أوامر فيها مثل طرفية **csch , tcsh , zsh**

ولكن الطرفية التى سوف نعمل عليها من خلال الشرح بإذن الله هى طرفية **bash** والتى ترمز إلى **Bourne again shell** وسوف نتناول شرح مفصل وبالتدرج لأوامر كثيرة ومفيدة بإذن الله .

## الجزء الأول : أساسيات لابد من الإلمام بها

في هذا الجزء إن شاء الله سوف نقوم باستعراض بعض الأساسيات المهمة والتي ينبغي لمستخدم لينوكس اليومى معرفتها والإلمام بها ونقسمها فى العناصر التالية :

- 1- كل شىء فى عالم لينوكس يعتبر ملف ( لا تقلق سوف أقوم بشرح كل عنصر على حده )
- 2- أقصى طول لأسماء الملفات على لينوكس .
- 3- الأسماء فى لينوكس حساسة تجاه الحروف الكبيرة والصغيرة.
- 4- بعض العلامات الخاصة تتجنب كتابتها فى أسماء الملفات .
- 5- ال WildCards وما طيفتها ؟

طبعا من القراءة الأولية للعناصر لا يستطيع الفرد العادى من معرفة المراد منها ولذلك نبدأ أولا مع أول عنصر:

### -كل شىء فى عالم لينوكس ملف-

في هذا البند غموض ما بمعنى البعض يتعامل بشكل يومي مع أنظمة التشغيل المختلفة سواء كانت أنظمة لينوكس أو ماك أو حتى ويندوز ولا يعلم غير شىء وحيد وهو أن ما يسمى ملف هى الملفات التى يمكن تحريرها سواء كانت بالكتابة ، آخرون قد يعلمون أن الصوتيات والفيديوهات تعتبر ملفات ، آخرون يعلمون أن الصور ملفات وهكذا .... وهذا صحيح ولكن ما أود أن أصل إليه أنهم يعتقدون أن هذه هى الملفات فقط ولا شىء سواها على سبيل المثال القرص الصلب من معناه الظاهرى كتلة من تركيبة معادن لا يمكن التعامل معها إلا من خلال أسلاك ال Data وال power ولا يمكن التغيير فى بيانات الهارد من خلال استخدام نظام التشغيل !

جاء لينوكس ليغير كل هذه المفاهيم ويعتبر أى شىء يخص الحاسب ملف سواء كان قرص صلب ، كروت الشبكات ، كروت الشاشة أى عتاد بجانب البرامج يعتبر ملف أيضا .

البعض قد يستغرب ذلك ولكن كيف للينوكس أن يقوم بذلك ؟

الإجابة على هذا السؤال بسيطة جدا وهى أن أى شىء يقوم لينوكس بالتعامل معه يعتبره مجرد سيل من ال bits وال bytes ولا شىء سوى ذلك ولكن على النقيض البرامج هى التى تتولى معرفة الأشياء مثلا محررات النصوص هى عبارة عن ملفات بالنسبة للينوكس ولكن محرر النصوص هو من صمم لكى يتعامل مع الملفات الكتابية وكيفية حفظها والتعديل عليها إلخ من هذه الأمور ، كذلك وصلات الشبكة المختلفة يتعامل لينوكس معها على أساس أنها ملفات مكونة من bits و bytes ولكن تطبيقات الانترنت هى من تعلم كيفية التعامل مع الوصلات الشبكية وهكذا .

## -العنصر الثانى وهو الطول الأقصى لأسماء الملفات على أنظمة اللينوكس -

لنرجع قليلا بذاكرتنا إلى الخلف حين كنا نستخدم نظام **Ms-Dos** ولنتذكر كم كان أقصى طول للملفات ؟ الإجابة بسيطة جدا وهو ثمانية أحرف بالإضافة إلى ثلاثة أحرف هى امتداد ونوع الملف إذا كان نصي يأخذ الشكل **txt**. وإذا كان فيديو مثلا يأخذ الشكل **wmv**. وهكذا .

ولكى تلاحظ ذلك مثلا أنت تستخدم ويندوز قم بتسمية ملف نصي جديد بأى اسم ولكن راعى أن يكون الاسم طويلا مثلا **communication theory** وتأكد من عدد أحرف الكلمة ستجدها عشرون حرفا ( البعض سيندهش يقول هى تسعة عشر حرفا )

صحيح عدد الحرف تسعة عشر حرفا ولكن لو تمعنا قليلا سوف نجد أننا قمنا بتحرير مسافة بين كلمة **communication** وكلمة **theory** ونظام التشغيل يتعامل معها على أساس أنها حرف من حروف الكلمة .

والآن قم بعمل إعادة تشغيل الجهاز والدخول إلى نظام التشغيل **Ms-Dos** عن طريق قرص اقلع ذاتى أو قرص مرن .

لنفرض الآن أنك على محث الأوامر الخاص بال Dos ويأخذ الشكل التالى مثلا **<A:\>**

بعد ذلك قم بالدخول إلى القسم الذى قمت بإنشاء الملف النصي عليه أولا قم بكتابة اسم القسم وليكن **E:** ثم اضغط زر التأكيد **Enter** ثم قم باستعراض الملفات الموجودة على القسم باستخدام الأمر **dir** .

الآن مثلا ظهرت الملفات الفرعية على القسم ماذا تلاحظ ؟ .....

سوف تلاحظ أن ال **Dos** قام باختزال اسم الملف من **communication theory** إلى الشكل التالى **commun~1.txt** وظهرت علامة جديدة هى ~ وتفيد بأن هناك تكملة للإسم .

أما أنظمة **Pre-OS X Macs** فالطول الأقصى لأسماء الملفات فيها كبير نسبيا بالمقارنة بنظام **Ms-Dos** وهو واحد وثلاثين حرفا. (31)

أما عند الحديث عن أنظمة اللينوكس واليونكس بصفة عامة فحدث ولا حرج تستطيع الطرفية أن تتعامل مع أسماء ملفات طولها الحرفى يصل إلى 255 حرف ويعتبر طول مناسب جدا لوصف ما يحلو لك لأسماء ملفاتك عند التعامل معها من خلال الطرفية !

ولكن من المفضل عند كتابة أسماء الملفات ألا يزيد طول الإسم على ثمانون حرفا كى يسهل عليك الأمر عند التعامل معه من خلال الطرفية حتى لا يحدث انكسار للاسم فى سطر آخر هذه نصيحة فضلا لا أمراً أمامك كل الخيارات مثنان فما أعلى !!!

## -العنصر الثالث وهو حساسية الاسماء تجاه الحروف الكبيرة والصغيرة او- (Capitall and Small)

اختلافا مع الويندوز وأنظمة الماك لينوكس عالم آخر !! نعم ففي هذا العنصر تجد دقة لينوكس المتناهية عند تحديد الأسماء والمسميات للملفات على النظام بمعنى أن لينوكس يتعامل بحساسية مفردة مع أسماء الملفات التي تحتوى حروفا كبيرة وصغيرة ولفهم ذلك مثال:

يوجد لدينا ثلاثة ملفات بنفس الاسم ( لغويا ) وهم على الترتيب:

```
*mysql.txt  
*Mysql.txt  
*MySQL.txt
```

في المثال أعلاه نجد ثلاثة ملفات نصية تحمل نفس الاسم من جهة اللغة ولكن من جهة لينوكس فالأمر مختلف جدا ، يتعامل لينوكس مع هذه الأسماء الثلاثة على أنها ثلاثة ملفات مختلفة كل منها لا يمت للآخر بصلة .

أما لو كنت على أحد أنظمة التشغيل الأخرى مثل ويندوز وقمت مثلا بإنشاء الملف الأول **mysql.txt** وهممت بإنشاء الملف الثاني **Mysql.txt** سوف يقوم النظام بإعطاء رسالة تظهر أمامك مفادها " هذا الملف موجود مسبقا هل تريد وضع الملف الجديد مع استبداله بالملف القديم ؟ "

وأما الحساسية فأیضا تعنى أن كل شيء في لينوكس مختلف إذا تضمن حروفا كبيرة وصغيرة ، بمعنى الأمر **ls** يختلف تماما عن **LS** يختلف عن **LS** وهكذا قيس الأمور كلها على نفس هذا المنوال ولذلك يجب مراعاة هذه النقطة عند كتابتك لأمر ما أو ملف ما أن تقوم بكتابته بالشكل الصحيح حتى لا يحدث مالا يحمد عقباه .

مثلا قد تنوى حذف ملف معين كما ذكرنا بالأعلى مثل ملف **mysql.txt** ولكن عن طريق الخطأ قمت بإدخال اسم ملف آخر وهو **Mysql.txt** حينها سيستجيب لك النظام ويفعل ما تأمره به !! لأنك حينئذٍ السيد الأمر مملوكه بفعل ما يريد تنبه !!!

ولذلك من إحدى مميزات لينوكس أنه يعلمك الحذر في كل ما تفعله ، وأن تكون دقيقا في كل أمورك صغيرة كانت أو كبيرة والمثل المضروب بذلك " الوقاية خير من العلاج " ! وذاك شيء جيد مقارنة بأي نظام طبيخ آخر ( = < : )



## - العنصر الرابع وهو علامات خاصة تتجنبها عند كتابتك لأسماء ملفاتك -

فنقول بحمد الله كل نظام تشغيل يحتوى على مجموعة من العلامات الخاصة تكون محجوزة للنظام فقط ولا يحق لأي مستخدم أن يقوم بوضعها فى أسماء الملفات على سبيل المثال عند تسمية ملف على نظام التشغيل ماك فلا يحق للمستخدم وضع (:) فى الأسم أو **colon**

بالمثل مستخدمى ويندوز لا يحق لهم وضع ال (\) أو ال **backslash**، وبالتالى ينطبق هذا الأمر على لينوكس كونه لا يعدو نظام تشغيل له خصوصياته ومن أهم العلامات الخاصة المحجوزة للينوكس هى ال (/) أو ال **forward slash**.

ولكى نستطيع فهم ذلك نضرب مثالا :

نفترض أنك تريد استعراض محتويات ملف يحمل الإسم التالى **books/to\_buy.txt** ستقوم بتنفيذ الأمر التالى :

```
less /home/oem/books/to_buy.txt
```

على الفور ستظهر هذه الرسالة فى الطرفية مفادها :

```
/home/oem/books/to_buy: No such file or directory
```

ولكن لماذا حدث ذلك مع العلم أننا قمنا بكتابة اسم الملف بطريقة صحيحة ؟

الإجابة حينما قمت أنت بكتابة اسم الملف المحتوى على ال / قام الشل بالبحث عن اسم الملف بالترتيب من خلال المسار المعطى أولا فى المجلد **home/** ثم المجلد **oem/** وجاء ليقوم بفتح الملف وجد مجلد آخر اسمه **books/** غير موجود بالمسار لأن علامة / بينت للطرفية أن ال **to\_buy** ملف يندرج تحت مجلد آخر اسمه **books/**

من ضمن العلامات الخاصة التى لا يحق لمستخدم لينوكس استخدامها هى ال (-) أو ال **dash** فى بداية أسماء الملفات حيث أن استخدام هذه العلامة يحول الملف من ملف عادى ليكون فى صيغة جديدة وهى استخدام الملف كا **option** لأمر تقوم بتنفيذه من خلال الطرفية ولاستيعاب ذلك نفترض التالى :

تحاول إنشاء ملف جديد باسم **books\_to\_buy.txt** وبالفعل قمت بإنشاءه ولكن لسبب ما أردت حذفه لأنك لست فى حاجة إليه فمن الطبيعى أن تقوم باستخدام الأمر **rm** وهو الأمر الخاص بحذف الملفات وبالتالى ستكون صيغة الأمر التالى كالتالى :

```
rm -books_to_buy.txt
```

سيظهر لك التالى فى الشل :

```
rm: invalid option -- b
```

ومن ضمن المحظورات أيضا عند تسمية الملفات تجنب وضع مسافات بين كل كلمة فى اسم الملف مثلا يوجد الملف **books to buy.txt** أى يوجد مسافة بين كلمة **books** وكلمة **to** كذلك يوجد مسافة بين كلمة **to** وكلمة **buy**



فلو أردت مثلاً حذف هذا الملف فسيحدث خطأ لا تتوقعه حيث أن الطرفية تتعامل مع المسافات وكأنها **arguments** مختلفة فعند حذف الملف ستقوم بتنفيذ الأمر التالي :

```
rm books to buy.txt
```

ستتعامل الطرفية بالترتيب مع الأمر لتنفيذه كالتالي :

أولاً ستحاول الطرفية حذف الملف **books** ثم الملف **to** ثم الملف **buy.txt** وبالتالي قد يحدث ما لا تتوقعه وتقوم بحذف ملفات أخرى لا تريد حذفها ووقتها لا يفيد الندم على ما فات تنبه لذلك

```
rm: cannot remove `books': No such file or directory
rm: cannot remove `to': No such file or directory
rm: cannot remove `buy': No such file or directory
```

كما رأيت حاول الأمر **rm** حذف الملفات بالترتيب ولكن لا جدوى !!

ولكن سؤال يتبادر للأذهان وهو كيفية حذف ملف يحتوي اسمه على كلمات بينها مسافات ؟؟

يأتى هنا دور ال \ أو ال **backslash** حيث تقوم هذه العلامة بإهمال المسافة الموجودة بين كل كلمة من كلمات اسم الملف .

ولتوضيح ذلك ، من المثال السابق تريد حذف الملف **books to buy.tx** ستقوم بعمل التالي فى الشل :

```
rm books to buy.txt
```

وسيكون كل شئ على ما يرام ولكنها طريقة مرهقة فلذلك من الأفضل عدم وضع مسافات بين كلمات أسماء الملفات .

ولكن ألا توجد طريقة أفضل من تلك ال \ ؟

بالفعل توجد طريقة أخرى نوعاً ما أسهل لحذف ملفات تحتوى أسماؤها على مسافات بين كل كلمة وكلمة وهو وضع اسم الملف المراد حذفه بين علامتى " أى **quotation marks** مثال :

```
rm "books to buy.txt"
```

ولكن مع مرور الوقت ستشعر بالإرهاق الشديد لكثرة استخدامك لل **quotation marks** ولذلك من الأفضل كما أسلفنا عدم وضع مسافات سيكون خيار جيد أليس كذلك ؟ !

وبعد أن استعرضنا بعض العلامات الخاصة بنظام لينوكس والتي لا يمكن استخدامها عند تسمية الملفات ، يكون كل شئ متاح لك فى التسمية بدأ من استخدام حروف فقط سواء كانت حروف صغيرة أو كبيرة ، أو حروف مع أرقام ، أو حتى أرقام فقط.

## - العنصر الخامس فهو ما هى ال WildCards وما وظيفتها ؟ -

تعال لتتخيل سويا أنك تمتلك ملف نصي و مجموعة من الصور لمناظر طبيعية خلابة من صنع الملك سبحانه وتعالى أشجار وجبال وسماء وأنهار وما فى الطبيعة من جمال ربانى وهذه الصور مثلا تأخذ ترتيبا رقميا تصاعدي وتريد مثلا حذف هذه الصور نظرا لوجود نسخة أخرى منها على القرص الصلب فكيف يمكنك عمل ذلك من خلال الطرفية ؟

الإجابة يقينا ستقوم باستخدام أمر الحذف **rm** ، ولكن هل ستقوم بحذف الصور صورة صورة أو دفعة واحدة ؟

طبعا من الممكن أن تقوم بحذف الصور عن طريق الواجهة الرسومية باستخدام الفأرة وذلك بعمل ضغطة يمنى على المجلد الموجود فيه الصور وتختار كلمة **Delete** ، ولكن كيف لك أن تقوم بذلك من خلال الطرفية ؟

هنا يأتى دور ال **Wildcards** فى تنفيذ أمر ما على مجموعة ملفات دفعة واحدة وحينها لن تحتاج لتنفيذ الأمر أكثر من مرة ، وال **Wildcards** عبارة عن علامات خاصة أيضا مكونة من :

- 1- \* أو ال **asterisk**
- 2- ? أو ال **question mark**
- 3- [ ] أو ال **square brackets**

والآن نستعرض كيفية الإستفادة من ال **Wildcards**

على سبيل المثال الصور تاخذ الترتيب التالى :

```
mido1.jpg  
mido2.jpg  
mido3.jpg  
mido4.jpg  
mido5.jpg  
mido6.jpg  
mido7.jpg  
mido8.jpg  
mido9.jpg  
mido10.jpg  
mido11.jpg  
mido12.jpg  
mido1.txt
```

السؤال الأول : حذف الصور بالكامل ؟

نقوم باستخدام ال **wildcard (\*)** لإهمال تأثير ما يأتى بعدها كالتالى :

```
rm mido*.jpg
```

هنا قام الأمر بحذف الصور كاملة ولكن مع الأخذ فى الاعتبار أننا قمنا بكتابة الإمتداد حتى لا نقوم بحذف الملف **mido1.txt**

السؤال الثانى : حذف الصور من رقم 10-12 ؟

```
rm mido1*.jpg
```

هنا قام الأمر بحذف الصور المرقمة من 10-12 مع الأخذ فى الاعتبار أننا قمنا بكتابة الإمتداد حتى لا نقوم بحذف الملف [mido1.txt](#)

السؤال الثالث : حذف الصور بما فيها الملف النصى ؟

```
rm mido*
```

السؤال الرابع : حذف الملف النصى فقط ؟

```
rm *.txt
```

السؤال الخامس : حذف المجلد بالكامل ؟

```
rm *
```

كما رأينا كانت هذه بعض استخدامات ال **\*** فى تنفيذ أمر عدة مرات بدون تكرار الأمر ونلاحظ السرعة فى تنفيذ الأوامر ولكن يجب الحذر حتى لا تقع فى أخطاء تسبب مشاكل !!

أما عن ال **wildcard** الثانية وهى ال **?** فلتوضيح كيفية استخدامها نقوم بعمل التالى :

السؤال الأول : حذف الصورة الرقمة من 10-12 ؟

```
rm mido1?.jpg
```

السؤال الثانى : حذف الصور المرقمة من 1-9 ؟

```
rm mido?.jpg
```

لاحظ هنا الفرق ما بين ال **?** ، **\*** أن ال **?** تقوم بحذف خانة واحدة بعدها وليس كل ما يليها ولذلك الصور من 10-12 لن تكون فى نتائج الحذف ( **جرب ذلك بنفسك** )

السؤال الثالث : حذف الصور كلها بما فيها الملف النصى ؟

```
rm mido?.*
```

هذه كانت أمثلة على استخدام ال **?** فى تنفيذ أمر عدة مرات بدون تكرار أيضا مع ملاحظة الفرق بينها وبين ال **\*** فى حالة السؤال الثانى.

والآن ننتقل إلى العلامة الثالثة والأخيرة وهى ال [] أو ال square brackets والتي تستطيع من خلالها تنفيذ أمر عدة مرات بدون تكرار ولنرى التالى :

السؤال الأول : حذف الصور 11 ، 12 فقط ؟

```
rm mido1[12]
```

السؤال الثانى : حذف الصور من 10 - 12 ؟

```
rm mido1[0-2]
```

وهكذا لتحديد خيارات الحذف أو الصور المراد حذفها يتم وضعها داخل ال []

وبعد أن استعرضنا بعض الأساسيات التى من الممكن احتياجها فى وقت ما سوف نستعرض أشياء أخرى تعتبر أساسية فى التعامل مع الطرفية وقد نقلتها من كتاب linux admin للكاتب أحمد نجيب الشهاوى لدعم الموضوع .

## أهم المفاتيح للتعامل مع الشل

space : به يتم الفصل بين الأمر والخيار الذى يتبعه.

enter : هذا المفتاح هو الذى يخبر الشل أنك انتهيت من كتابة الأمر لكى تبدأ فى التعامل مع الأمر.

backspace : عند كتابة أمر أو خيار خطأ يتم به مسح الحروف الخاطئة .

tab : يعتبر هذا المفتاح من أهم المفاتيح، فهو يظهر خاصية من أهم خصائص الطرفية ألا وهى إكمال الأمر.

مثلا إذا أردت الأمر mkdir فما عليك إلا كتابة الأحرف mkd ومن ثم ستكمل لك الطرفية بقية الأمر إذا ضغطت tab مرتين .

أما إن كان هناك أكثر من أمر يبدأ ب mkd ( مثلا mkdir و mkdev ) فسيتم عرضهم أمامك لكى تختار الأمر المناسب.

ويلاحظ أنه إذا تم الضغط على ال tab بدون كتابة أى أحرف فسيعرض عليك النظام عرض كل الأوامر الموجودة به (يمكنك تجربة هذه الملاحظة )

shift+page up : عند امتلاء الشاشة أمامك وأردت إسترجاع الجزء الذى اختفى بالأعلى فيمكننا بهذه المفاتيح رؤيتها.

shift+page down : عكس الذى قبله.



**up or down** : الأسهم أعلى وأسفل ، هذه أيضا تُظهر خاصية من الخصائص الهامة للـ **shell** فال **up arrow** يأتي لك ثانية بالأمر السابق دون الحاجة لإعادة كتابته ثانية.

**ctrl+alt+F1** : إذا كنت تعمل على الواجهة الرسومية وأردت الانتقال الى الواجهة النصية أى الشاشة السوداء، فهذه المفاتيح الثلاثة يمكنك الانتقال إليها .

((هناك 6 شاشات **console** تبدأ من **F1** وحتى **F6**، أما **F7** فترجع بك ثانية إلى الواجهة الرسومية ))

طبعا هذه كانت مقدمة بسيطة عن بعض العناصر والأساسيات اللى لازم نعرفها خلال فترة معرفتنا بسطر الأوامر وإن شاء الله فى الفصل القادم سوف نستعرض الأوامر الأساسية التى تقريبا يحتاجها مستخدم لينوكس فى شغله العادى .

## الجزء الثانى : الأوامر الأساسية فى لينوكس

استعرضنا فى الفصل السابق بعض الأساسيات التى سيتم تناولها على مدار الموضوع وفى هذا الفصل سنتعرض لبعض الأوامر الأساسية والمهمة لمستخدم لينوكس بشكل يومى وسيتم شرحها بالتفصيل إن شاء الله.

### الأمر : ls

يعتبر الامر **ls** من أكثر الأوامر شيوعا وانتشارا واستخداما لما له من أهمية شديدة لمستخدم لينوكس بشكل شبه يومى . ولمعرفة أهمية الأمر نفترض أنك تمتلك ملفات تخصك على إحدى المجلدات وفى يوم ما أردت استعراض هذه الملفات على هذا المجلد فما الذى يمكنك من ذلك ؟

الإجابة هو الأمر **ls** حيث يقوم الأمر باستعراض الملفات والمجلدات الفرعية تحت المجلد الرئيسى التى توجد فيه هذه الأشياء ، والآن لنبدأ سويا رحلة البحث عن المتعة باستخدام سطر الأوامر ، لمستخدمى واجهة جنوم عليهم الذهاب للمسار التالى :

Applications => Accessories => Terminal

والآن على الفور أول شئ نقوم بكتابة الأمر **ls** فى صورته البسيطة دون دمج مع أى خيارات اضافية ثم بعد ذلك نضغط على الزر **enter** كما يلى :

```
oem@ubuntu:~$ ls
1.rmvb
38AR_Crucifixion_Fact_or_Fiction_vs_Douglas_512KB.rmvb
Desktop
documentation and presentation of parallel processing for SDR using OMAP.rar
Examples
w32codecs_20061022-0.0_i386.deb
```

كما يمكنك استعراض الملفات الموجودة على مجلد فى المسار الحالى دون الحاجة إلى كتابة مسار المجلد ولنفترض أننا نريد استعراض محتويات المجلد **Desktop** نقوم بعمل التالى :

```
oem@ubuntu:~$ ls Desktop
33391_hi.ram
4.rm          index_files
index.php     spraah2.pdf
```

كما يمكنك أيضا استعراض نفس المجلد عن طريق المسار المرتبط الخاص به أو ما يسمى ال **relative path** وذلك كالتالى :

```
oem@ubuntu:~$ ls /home/oem/Desktop/
33391_hi.ram      spraah2.pdf
4.rm              index_files
index.php
```

ولكن ألا توجد طريقة أسهل فى التعامل لعرض ملفات ال **home directory** تكون أفضل من ذلك ؟

نعم ، توجد العلامة **~** والتى تقوم نفس مقام المسار **home/oem/** دون أدنى اختلاف ولنتأكد من ذلك من خلال الطرفية بعمل التالى:

```
oem@ubuntu:~$ ls ~/Desktop
33391_hi.ram      spraah2.pdf
4.rm              index_files
index.php
```

ونلاحظ التالى أن الأمر **ls** قام باستعراض الملفات الموجود على المجلد الحالى أى المسار الفعلى الذى نكون عليه لحظة كتابة الأمر أو أين من المجلدات الفرعية على المسار ، ولنتنقل إلى نقطة أخرى ألا وهى : هل يتطلب أن نكون على المسار كى نعرض محتوياته؟

الإجابة بالطبع لا فقد تكون على مسار وتقوم باستعراض ملفات مسار آخر ولاستيعاب ذلك نفترض أننا على مسار **home/oem/** ونريد استعراض ملفات مسار آخر فلنرى ذلك فى الطرفية :

```
oem@ubuntu:~$ pwd
/home/oem
```

لاحظ أننا أولا قمنا بمعرفة المسار الحالى عن طريق استخدام الأمر **pwd** لمجرد التأكد ليس إلا.

```
oem@ubuntu:~$ ls /var/cache/
apt  debconf      gnome-system-tools  man      setup-tool-backends
cups  dictionaries-common  locate      pppconfig
```

كما رأيت فعلنا ذلك بكل سهولة ويسر ، ولنتنقل إلى نقطة أخرى وهى كيفية إيجاد ملف معين على مجلد معين على مسار معين كيف لك أن تقوم بذلك ؟

أوضح قليلا الآن حان دور استخدام الـ **wildcards** نفترض أننا نود البحث عن ملف اسمه **mysql\_under\_ground.pdf** وكل ما نتذكره عن الملف اسمه والامتداد الخاص به والمجلد الذي يوجد به ولكن المجلد يحتوي على ملفات ذات الامتداد **pdf** وأخرى ذات امتداد **txt** وما نريده هو عرض الملفات التي تنتهي بامتداد **pdf** نقوم بعمل التالي :

```
oem@ubuntu:~$ ls ~/Desktop/*.pdf
/home/oem/Desktop/mysql_under_ground.pdf /home/oem/Desktop/spraak2.pdf
/home/oem/Desktop/new.pdf
```

الآن ظهرت أهمية استخدام الـ **wildcards** كما رأينا ، ولكن نحدد عنصر البحث في استعراض الملف بواسطة جزء من اسمه نقوم بعمل التالي :

```
oem@ubuntu:~$ ls ~/Desktop/*under*.pdf
/home/oem/Desktop/mysql_under_ground.pdf
```

بعد أن ظهرت أهمية الـ **wildcards** وكيفية استخدامها ودمجها مع الأمر **ls** نبدأ الآن في استعراض الخيارات التي يمكن اضافتها مع الامر .

## 1- استعراض الملفات الموجودة على المجلدات الفرعية:

View a List of Files in Subfolders (ls -R or ls --recursive)

يعتبر الخيار **-R** من ضمن الخيارات المهمة والتي يتم إضافتها إلى الأمر **ls** وذلك لاستعراض الملفات الموجودة على المجلدات الفرعية داخل مجلد رئيسي بسهولة ويسر دون الاحتياج إلى فتح كل مجلد على حده كما يلي :

```
oem@ubuntu:~$ ls -R ~/Desktop
/home/oem/Desktop:
33391_hi.ram          index.php
4.rm                 mysql_under_ground.pdf
new.pdf              spraah2.pdf
index_files

/home/oem/Desktop/index_files:
362922962_1887f6723a_o.png ips_menu_html.js
365159135_b0fcd2be76_o.jpg ips_menu.js
7951.png              ips_xmlhttprequest.js
Alfred.gif            js.js
arab.                  jss.php
arab_data              Market.jpg

/home/oem/Desktop/index_files/arab_data:
arab_01.gif arab_04.gif arab_10.gif arab_13.gif
arab_02.gif arab_08.gif arab_11.gif arab_14.gif
arab_03.gif arab_09.gif arab_12.gif arab_15.gif
```

كما رأينا أعلى قام الأمر أولا باستعراض الملفات بالمجلد الرئيسي **Desktop** ثم بعد



ذلك تم استعراض الملفات الموجودة داخل المجلد الفرعى `index_file` ثم بعد ذلك تم استعراض الملفات الموجودة داخل المجلد الفرعى `arab_data` ، ولاحظ أيضا أن كل مجلد فرعى يظهر بالمسار الارتباطى الخاص به أو `relative path`.

ولكن هل الخيار `R-` دائما هو الأفضل ؟

الإجابة لا ، ولتعرف لماذا جرب التالى فى الطرفية وستعرف ماذا أقصد :

```
ls -R /var
```

## 2 - استعراض الملفات الموجودة على أى مجلد فى عمود رأسى واحد فقط:

```
View a List of Contents in a Single Column (ls -lor ls --format=single-column)
```

تعودنا قبل ذلك عند كتابة الأمر `ls` وحده دون اضافة أى خيارات أن تظهر الملفات على شكل أعمده وصف وبين كل عمود وعمود مسافتين لسهولة قراءة خرج الأمر ، ولكن الجديد مع الخيار `-1` أنه يمكنك استعراض ملفات أى مجلد فى عمود رأسى واحد فقط كما يلى :

```
oem@ubuntu:~$ ls -1 ~/Desktop
33391_hi.ram
4.rm
index_files
index.php
mmmm
spraah2.pdf
```

ملحوظة : يمكن استخدام الخيار `R-` مع الخيار `-1` لعمل الوظيفتين اللتان يقوم بهما الخياران فى آن واحد أى استعراض الملفات على المجلدات الفرعية داخل المجلد الرئيسى مع عرض نتيجة كل مجلد فرعى فى عمود رأسى واحد فقط ليكون الأمر بالشكل التالى :

```
ls -1R ~/Desktop
```

أما إذا استخدمت الخياران `1R-` مع الأمر `ls` فى مجلد رئيسى يحتوى على مئات المجلدات الفرعية مثل `var/` ، فقد تحتاج إلى ... `ctrl+c` حينها ستعرف فائدة الضغط على هذه المفاتيح.....

## 3 - استعراض الملفات الموجودة على أى مجلد مع وضع Comma بين كل ملف وآخر :

view Contents As a Comma-Separated List ( ls -m )

خيار آخر لتنظيم شكل الملفات عند استعراضها باستخدام الأمر **ls** وهو الخيار **-m** حيث يقوم بوضع **comma** بين كل ملف وآخر كما يلي :

```
oem@ubuntu:~$ ls -m /var
backups, cache, games, lib, local, lock, log, mail, opt, run, spool, tmp
```

#### 4 - استعراض الملفات والمجلدات المخفية على مسار معين:

View Hidden Files and Folders (ls -a or ls --all)

حتى هذه اللحظة كانت نتائج استعراض الملفات والمجلدات سواء كانت رئيسية أو فرعية باستخدام الأمر **ls** الملفات والمجلدات المرئية فقط ، ولكن ماذا لو كانت هناك ملفات أو مجلدات مخفية ؟ هنا الخيار **-a** ليقوم بهذه المهمة ، فعلى سبيل المثال مجلد ال **home directory** الخاص بك يحتوى ملفات مخفية والبعض بل يمكن الكثير لا يعلم ذلك ولكي تعرف ذلك تابع التالي :

```
oem@ubuntu:~$ ls -a ~/
.
..
.bash_history
.bash_logout
.bash_profile
.bashrc
books~
.DCOPserver_ubuntu__0
.DCOPserver_ubuntu_:0
Desktop
.dmrc
using OMAP.rar
.esd_auth
.evolution
Examples
.fonts.cache-1
.mozilla
.mplayer
```

لاحظ : المجلدات المخفية يكون في بدايتها نقطة أو ال **dot** ، وقد رأيتَ في المجلد الرئيسي والخاص بالمستخدم الذي أنشأته أثناء مرحلة التنصيب يحتوى على الكثير من المجلدات الفرعية مع مجلدات أخرى فرعية ولكنها مخفية .

بعض من هذه المجلدات المخفية تكون لبرامج أنت تقوم بتنصيبها وتحتوى على ملفات الـ **configuration** الخاصة بها ، والبعض الآخر لبرامج أساسية يتم تنصيبها أثناء تثبيت التوزيعة بدون تدخل من المستخدم.

أيضا من ضمن الأشياء التى أودُ لفت انتباهك إليها هو أول سطرين فى نتيجة الأمر **ال .. (dot and double dots)** .. حيث تشير ال . إلى المجلد الحالى الذى توجد فيه الآن أما ال .. فتشيران إلى المجلد الأب الذى يحوى المجلد الرئيسى وطبيعى فى هذه الحالة هو المجلد **home/** بحيث يكون المجلد **home/** هو المجلد الأب أو فيما يسمى بال **parent directory** للمجلد . **oem/** هذان المجلدان متواجدان بشكل عام فى كل المسارات التى توجد بالتوزيعة ولا تستطيع التخلص منهم . وبالتالي توقع أن تجدّهما فى كل مرة تستخدم فيها الخيار **-a** . ولك أن تشكر الخيار **-a** أن نبهك إلى وجود ملفات ومجلدات لم تكن تعلم عنها شيئا !!

## 5 - استعراض الملفات مع إظهار أنواعها:

Visually Display a File's Type (ls -F or ls --classify)

بالأعلى استعرضنا بعض الخيارات التى يتم إضافتها للأمر **ls** لتكسيه رونقا رائعا وتُضفى عليه لمسات جذابه لتعرض لك ما تريده ولكن هل تظن الأمر ينتهى عند هذا الحد ؟ البعض قد يظن أن الأمر **ls** قد أتى بكل مافى جعبته ، ولكن ليس الآن هناك المزيد والمزيد .

من ضمن الخيارات الرائعة التى تأتى مع الأمر **ls** هو الخيار **-F** ، حيث يقوم هذا الخيار بعرض نوع الملف الموجود على المجلد سواء كان الملف تنفيذى أو **Executable** ، مجلد أو **Directoty** ، رابط رمزى أو **Symbolic link** ، أو **FIFO** وهى اختصار لعبارة **First in First out** وليس هناك مجال لشرح ال **FIFO** لأنه يحتاج درس آخر خاص به ، كل ما يهمنى الآن أن تعرفه أن الخيار **-F** يقوم بعرض رموز أو علامات كل منها تدل على نوع من الأنواع التى قمت بذكرها ..... والآن لنستعرض كيفية كتابة الأمر فى الطرفية كما يلى :

```
oem@ubuntu:~$ ls -F /bin
bzgrep*      gunzip*      nano*        umount*
bzip2*       gzexe*       nc*          uname*
bzip2recover* gzip*        netcat@      uncompress*
bzless*      hostname*    netstat*     vdir*
bzmoro*      ip*         pidof@       which*
mido/        mysql/
```

```
oem@ubuntu:~$ ls -F /dev
log=      ptyc9 ptyrd ptyx1 tty18      ttybe ttyr2 ttyt6 ttyya
loop/     ptyca ptyre ptyx2 tty19      ttybf ttyr3 ttyt7 ttyyb
```

لاحظ : قمت فى المثال السابق باستعراض أنواع بعض الملفات للمجلد **bin/** ، والبعض الآخر للمجلد **dev/** باستخدام الخيار **-F** وظهرت لدى نتيجة الأمر كما رأيت ، تدل العلامة **\*** على أن الملف الذى توجد أمامه أنه ملف تنفيذى أو كما قلنا **Executable** ، أما العلامة **@** فتدل على أن الملف رابط رمزى أو **Symbolic link** ، أما العلامة **/** فتدل على أن الملف مجلد ، والعلامة **=** تدل على ملف من نوع **Socket** ولذلك فهى علامات خاصة لمعرفة نوع الملفات من بين الأنواع السابق ذكرها .

**ملاحظة :** المجلد **bin/** الشكل الافتراضى له لا يحتوى على مجلدات نهائيا ولذلك قمت بوضع مجلدين فرعيين بداخل المجلد للتوضيح فقط...

هذا الجدول تلخيص للكلام السابق :

Character	Meaning
*	Executable
/	Directory
@	Symbolic Link
	FIFO
=	Socket



## 6 - استعراض الملفات ، التصاريح عليها ، المالك ، وأشياء أخرى ...

List Permissions, Ownership, and More (ls -l or ls --format=long)

من هنا نبدأ العمل الحقيقي ..... أجل فمن هذه اللحظة يبدأ مستخدم الطرفية لي تشعر وكأنه محترف ، تعلمنا سابقا أن الأمر **ls** يقوم بإظهار المجلدات الفرعية والملفات التي توجد داخل المجلدات الرئيسية بأشكال مختلفة نظرا لاستخدام خيارات مختلفة أيضا ، منها ما يظهر نوع الملف ، ومنها ما يظهر الملفات المخفية ، ومنها ما يقوم باستعراض الملفات في هيئات مختلفة سواء كانت مفصولة ب **Comma** أو تظهر في شكل عمود رأسى واحد وهكذا ....

ولكن لم نتعرض إلى الملفات بذاتها ، هنا الخيار **-l** ليقوم بدور مهم جدا على الملفات ألا وهو استعراض التصاريح عليها ، استعراض المالك لها ، تاريخ الإنشاء أو التعديل ، حجم الملف وغيره من المعلومات المفيدة جدا والتي قد يستخدمها بعض المخبرين أثناء العبث بأغراض الآخرين !!

ومثال على ذلك لتوضيح صيغة الأمر **ls** وإضافة الخيار **-l** معه كما يلي :

```
oem@ubuntu:~$ ls -l
total 972012
drwxr-xr-x 3 oem oem 364911700 2006-10-22 20:01 mysql
-rwxr-xr-x 3 oem oem 364911700 2006-10-22 20:01 addext
drwxr-xr-x 4 oem oem 4096 2007-01-27 06:26 Desktop
drwxr-xr-x 2 oem oem 4096 2007-01-25 18:24 mm
```

والخيار **-l** يرمز إلى الكلمة **long** ، وبالتالي نستنبط من معنى الكلمة **long** ألا وهو الطول ، أى أن الخيار يمدنا بالكثير من المعلومات المهمة جدا عن الملفات والمجلدات والآن لنستعرض كيفية فهم السطر من اليمين إلى اليسار ونأخذ مثال على ذلك السطر الأول :

**أولا :** الإسم **mysql** يدل على اسم الملف الذى تم استعراضه من قبل الأمر **ls** نتيجة كتابة الخيار **-l** معه.

**ثانيا :** الأرقام **20:01** ، **2006-10-22** تدل على الساعة والتاريخ اللذين تم فيهما التعديل الأخير على الملف .

**ثالثا :** الرقم **364911700** يوضح حجم الملف **mysql** ولكن بال **bytes**

**رابعا :** الإسم **oem** من جهة اليمين يوضح المجموعة المالكة للملف.

**خامسا :** الإسم **oem** من جهة اليسار يوضح المالك الأصلي للملف ( **المستخدم الذى قام بإنشاء الملف** ) .

**سادسا :** الرقم بجوار مالك الملف ، مثلا فى حالة الملف **Desktop** ( كما ذكرنا قبل **ذلك المجلد يعتبر نوع من الملفات الخاصة** ) فى السطر الخاص به رقم 4 وهذا الرقم له حالتان :

1 - إذا كان الرقم فى سطر خاص بملف عادى فهو يوضح عدد ال **hard links** الموجودة لذلك الملف.

2 - إذا كان الرقم فى سطر خاص بمجلد فهو يوضح عدد المجلدات الفرعية الموجودة بداخل المجلد الرئيسى وفى مثالنا المجلد **Desktop** يحتوى على أربع مجلدات فرعية .

**ملاحظة :** قمت بإنشاء مجلد فارغ وأسميته **mm** كما تلاحظون فى السطر الأخير ولكن يوجد فى السطر الخاص بهذا المجلد رقم 2 فكيف ذلك وقد ذكرت أن المجلد فارغ ؟

لنرجع قليلا إلى الوراء وبالتحديد فى الخيار **-a** فقد ذكرت أنه يوجد مجلدان فرعيان بشكل دائم مع أى مجلد رئيسى وهما **ال .** و **ال ..** أى المجلد الحالى والمجلد الأب (**parent directory**) ولذلك وجدنا عدد المجلدات الفرعية بالمجلد **mm** وقد وُضِعَ فى السطر الخاص به ولنرى ذلك:

```
oem@ubuntu:~$ ls -a mm/  
.  
..
```

**سابعا :** الرموز **drwxr-xr-x** وتشير إلى التصريح الخاص بالملف **mysql** ، ولكن من الوهلة الأولى يبدو وكأنه شفرة سرية !!

ولكنه فى الحقيقة من الأمور السهل جداً فهمها واستيعابها بقليل من المعرفة ، لاحظ معى هناك عشرة أحرف مقسمين إلى أربع مجموعات ( **قد لا يبدو الأمر كذلك** )

المجموعة الأولى : تتكون من الحرف الأول، (1)

المجموعة الثانية : تتكون من الحرف الثانى إلى الحرف الرابع (2-4)

المجموعة الثالثة : تتكون من الحرف الخامس إلى الحرف السابع (5-7)

المجموعة الرابعة : تتكون من الحرف الثامن إلى الحرف العاشر (8-10)

وعلى سبيل المثال نأخذ التصريح الخاص بالملف **mysql** ويبدو مقسما إلى المجموعات السابقة ذكرها بالشكل التالي :

```
d|rw|rx|r-x|r-x
```

والآن تعالوا معي لنعرف بالضبط كيفية فهم هذا التصريح من اليسار إلى اليمين :

المجموعة الأولى والتي تشمل الحرف الأول ( وفي حالتنا هذه هو حرف ال **d** ) تمثل نوع الملف سواء كان مجلد (**directory**) أو ملف عادي (**regular file**) وطبعا من البديهي أن يرمز الحرف **d** إلى الكلمة (**directory**) أو مجلد .

وبالتالي يعتبر الملف **mysql** من نوع مجلد أو ملف خاص كما ذكرنا سابقا ، بينما العلامة - أو ال **dash** توضح أنه ملف.

وطبعا توجد بعض الخيارات التي تؤدي هذه المهمة ولكن عن طريق الرموز الخاصة بها مثل الخيار **-F** والذي من ضمن علاماته ال **/** والتي ترمز إلى مجلد كما يمكن معرفة ذلك أيضا من خلال الخيار **--color** حيث عند دمج هذا الخيار مع الأمر **ls** تظهر ألوان للملفات بشكل مختلف .

ويرمز كل لون إلى نوع معين من الملفات وأكتفى فقط بسرد الجدول الخاص بهذه الألوان كالتالي :

Color	Meaning
Default shell text color	Regular File
Green	Executable
Blue	Directory
Magenta	Symbolic Link
Yellow	FIFO
Magenta	Socket
Red	Archive (.tar ,.zip ,.deb ,.rpm )
Magenta	Images (.jpg ,.gif ,.png ,.tiff)
Magenta	Audio (.mp3 ,.wav ,.ogg)

وبالطبع يوجد الكثير من هذه العلامات التي توضح الأنواع المختلفة من الملفات والجدول التالي يوضح كلامي :

Character	Meaning
-	Regular File
-	Executable
d	directory
l	Symbolic Link
s	Socket
b	Block Device
c	Character Device
p	Named Pipe

أما الأحرف التسعة الباقية فطبيعي تمثل المجموعة الثانية ، والثالثة ، والرابعة على الترتيب وبالتالي تكون التصاريح الخاصة بالملف على الترتيب تخص كلاً من المالك الأصلي أو **file's owner** ، المجموعة التابع لها الملف أو **file's group** .

والتصريح الأخير لمجموعة المستخدمين الباقين على النظام أو **other users** ، ففي المثال السابق نجد التصريح الخاص لمالك الملف **oem** هو **rwX** والتصريح الخاص بالمجموعة **oem** هو **r-X** والتصريح الخاص بباقي المستخدمين على النظام هو **r-X** ولكن ماذا تعني هذه الحروف ؟

في كل حالة يعنى الحرف **r** العبارة التالية "**yes , read is allowed**" أو يُسمح بقراءة الملف .

أما الحرف **w** فيعنى "**yes , write is allowed**" أو يسمح بالكتابة على الملف ( وكلمة "**write**" تعنى كلاً من التعديل والحذف أو ال **changing and deleting** )

أما الحرف **x** فيعنى "**yes , execute is allowed**" أو يسمح بتنفيذ الملف إذا كان **Executable** ،

أما عن العلامة - فتعنى "**no , dont allow this action**" فمثلا لو أتت ال - مكان الحرف **r** فى تصريح أى مجموعة فوقتها تعنى "**no , read is not allowed**" أو تصريح القراءة غير مسموح به كذلك لو أتت هذه مكان الحرف **w** أو الحرف **x**

وبالنظر إلى الملف **addext** مثلا نجد أن المالك **oem** له تصريح **rwX** أى يستطيع القراءة والكتابة والتنفيذ على الملف .

أما المجموعة **oem** فلها التصريح **r-X** أى تستطيع القراءة والتنفيذ فقط ، وباقي المستخدمين لهم التصريح **r-X** أى يستطيعوا أيضا القراءة والتنفيذ فقط .



هذا الكلام ينطبق على الملفات حيث معنى القراءة والكتابة مفهوم ولكن ألا يوجد شيء غريب بالموضوع كيف لنا مثلاً أننا ننفذ مجلد كما في حالة المجلد `mysql` ؟

قلنا أن كلا من الحرفين `r` ، `w` سواء بالنسبة للمجلد أو الملف العادي يحتملان نفس المعنى ولكن الوقفة هنا مع الحرف `x` حيث يختلف تنفيذ الملف عن تنفيذ المجلد كيف ذلك ؟

الحرف `r` إذا جاء في تصريح خاص بمجلد فهو يعنى أنه يمكنك تصفح محتويات المجلد دون عوائق ، أما إذا جاء الحرف في تصريح ملف فكما قلنا أننا أنه يمكنك قراءة الملف .

أما بالنسبة للحرف `w` إذا جاء في تصريح مجلد فهو يعنى أنه يمكنك إضافة محتويات أخرى للمجلد ، تسمية محتويات موجودة بأسماء أخرى ، حذف أى ملف من ملفات المجلد .

أما إذا جاء الحرف `w` في تصريح ملف فهو يعنى أنه يمكنك الكتابة على هذا الملف والتعديل عليه بإحدى محررات النصوص المعروفة مثل `nano` أو `vi` وغيرها .

والآن حان دور الحرف `x` فإذا جاء الحرف `x` في تصريح مجلد فهو يعنى أنه يمكنك الولوج إلى ذلك المجلد وتنفيذ البرامج الموجودة بداخله . أو الولوج إلى المجلدات الفرعية التي توجد بداخله .

أما إذا جاء الحرف في تصريح ملف فهو يعنى أنه يمكنك تنفيذ الملف ليقوم بعمل مهمة أو أداء وظيفة معينة أو بمعنى "Execute the file to become a program".

والآن وكما تعلمنا مدى قوة الخيار `-l` والتي ظهرت بشكل جلي في الأمثلة السابقة ، ولكن تزداد قوة الخيار `-l` أكثر وأكثر عند دمجه مع خيارات أخرى مثل الخيار `-a` والذي تعلمنا أيضاً قوته في إظهار الملفات المخفية والتي قد لا نلقى لها بالاً ليكون ناتج الامر `ls` فى النهاية بشكل التالى بعد دمج الخيارين :

```
oem@ubuntu:~$ ls -al ~
drwxr-xr-x  3 root root      4096 2007-01-22 02:12 .
drwxr-xr-x 31 oem  oem      4096 2007-01-28 21:57 ..
-rw-r--r--  1 oem  oem     18164916 2007-01-28 06:05 A3.41.zip
-rw-----  1 oem  oem      6260 2007-01-29 08:33 .bash_history
-rw-r--r--  1 oem  oem      2227 2007-01-22 02:12 .bashrc
-rw-----  1 oem  oem         0 2007-01-23 11:50 books~
drwxr-xr-x  9 oem  oem      4096 2007-01-29 10:23 Desktop
```

كما ترون قمنا باستعراض الملفات الموجودة على ال `home directory` أو `~` باستخدام خياران يقومان بوظيفتين مختلفتين وهما على الترتيب :

\*استعراض الملفات المخفية والتي تخص الخيار `-a`

\*استعراض الملفات ، التصاريح عليها ، المالك ، وأشياء أخرى... والتي تخص الخيار `-l`

## 7 - استعراض الملفات مع عكس الترتيب الهجائي لها :

Reverse The Order Contents Are Listed (ls -r or ls --reverse)

لو تمعنا قليلا فى الخيار السابق ألا وهو **-l** ودققنا النظر لوجدنا أنه يقوم بسرد الملفات ويقوم بترتيبها ترتيباً هجائياً بمعنى حسب ترتيب الحروف باللغة الإنجليزية يبدأ أولاً بالملفات التى تحتوى فى بداية أسمائها على الحرف **A** .

ثم الملفات التى تحتوى على الحرف **B** ، ثم الملفات التى تحتوى على الحرف **C** ، وهكذا ولكن ماذا لو أردنا أن نقوم بعمل العملية العكسية ، بمعنى أوضح نستعرض الملفات التى تحتوى فى بداية أسمائها على الحرف **z** ، ثم الحرف **y** ، ثم الحرف **x** .....انتهاءً بالحرف **A** ؟

هنا الخيار **-r** ليقوم بتلك المهمة حيث يقوم باستعراض الملفات ترتيباً هجائياً معكوساً وبالتالي يتم دمج الخيار **-r** مع الأمر **ls** كما يلى :

```
oem@ubuntu:~$ ls -alr ~
```

```
drwxr-xr-x 9 oem oem      4096 2007-01-29 10:23 Desktop
-rw-r--r-- 1 oem oem        0 2007-01-23 11:50 books~
-rw-r--r-- 1 oem oem     2227 2007-01-22 02:12 .bashrc
-rw-r--r-- 1 oem oem     6260 2007-01-29 08:33 .bash_history
-rw-r--r-- 1 oem oem    18164916 2007-01-28 06:05 A3.41.zip
drwxr-xr-x 31 oem oem     4096 2007-01-28 21:57 ..
drwxr-xr-x 3 root root     4096 2007-01-22 02:12 .
```

لاحظ : كما أسلفنا سابقا وذكرنا أن لينوكس حساس تجاه الحروف ال **capital** والحروف ال **small** ولذلك الخيار **-r** مختلف تمام عن الخيار **-R**

## 8 - استعراض الملفات وتصنيفها عن طريق الإمتداد الخاص بها :

Sort Contents By File Extension (ls -X or ls --sort=extension)

يُمكنك الخيار **-X** من استعراض الملفات وترتيبها ترتيبا هجائيا ولكن عن طريق الامتداد الخاص بها ، بمعنى نفترض أنه يوجد لديك أربعة ملفات بامتدادات مختلفة على سبيل المثال :

```
keymouse.zip  
mido.rar  
mysql.txt  
image.jpg
```

فحينما تقوم بتنفيذ الأمر **ls -X** على المجلد الذي توجد فيه هذه الملفات ، ولنفترض أنه المجلد **~** أو **home directory** فسيقوم الأمر بترتيب الملفات كالتالى :

```
oem@ubuntu:~$ ls -X  
image.jpg  
mido.rar  
mysql.txt  
keymouse.zip
```

**لاحظ :** ترتيب الحروف الهجائية للغة الإنجليزية للملفات الأربعة أو الحرف **j** ، ثم الحرف **r** ، ثم الحرف **t** ثم أخيرا الحرف **z** .

وبالطبع تستطيع دمج الخيار **-X** مع الخيار **-l** ليعرض لك الكثير من المعلومات عن الملفات وفى نفس الوقت يقوم بترتيبهم حسب الإمتدادات الخاصة بهم كما يلى :

```
drwxr-xr-x 9 oem oem 4096 2007-01-29 10:23 Desktop  
drwxr-xr-x 2 oem oem 4096 2007-01-27 07:50 mm  
-rw-r--r-- 1 oem oem 18164916 2007-01-28 06:05 A3.41.tar.gz  
-rw-r--r-- 1 oem oem 183457 2007-01-23 15:37 gg.mht  
-rw-r--r-- 1 oem oem 647142 2007-01-25 11:05 Firefox_wallpaper.png  
-rw-r--r-- 1 oem oem 364911700 2006-10-22 20:01 1.rmvb  
-rw-r--r-- 1 oem oem 5654152 2004-01-01 05:46 IAA_2.2.0.2150.zip
```

ملحوظة : عندما توجد مجلدات فرعية داخل المجلد الرئيسى فتكون فى مقدمة الترتيب لأنها طبيعى لا تحتوى على امتدادات ، ولاحظ أيضا إنه عندما توجد ملفات تحتوى أكثر من امتداد كما فى حالة الملف **A3.41.tar.gz** سيكون الترتيب خاص بالإمتداد النهائى ألا وهو **gz** ويكون الترتيب على أساسه .

## 9 - استعراض الملفات وتصنيفها عن طريق الأحجام الخاصة بها

Sort Contents By Size (ls -S or ls --sort=size)

تستطيع أيضا استعراض الملفات عن طريق الأحجام الخاصة بها على القرص الصلب لديك باستخدام الخيار **-S** وإضافته للأمر **ls** كما يلي :

```
oem@ubuntu:~$ ls -laS ~
```

```
-rw-r--r-- 1 oem oem 364911700 2006-10-22 20:01 1.rmvb
-rw-r--r-- 1 oem oem 18164916 2007-01-28 06:05 A3.41.zip
-rw-r--r-- 1 oem oem 5654152 2004-01-01 05:46 IAA_2.2.0.2150.zip
-rw----- 1 oem oem 1962871 2007-01-24 12:23 ghhh
-rw-r--r-- 1 oem oem 647142 2007-01-25 11:05 Firefox_wallpaper.png
```

لاحظ : قمت بإضافة الخيار **-S** إلى الخيارين **-l** و **-a** لبيان وظيفته بشكل واضح ، كما أود لفت انتباهك إن أحجام الملفات التي يقوم الخيار **-S** بإظهارها تكون بالـ **bytes**

## 10 - استعراض الملفات وتصنيفها عن طريق الأحجام الخاصة بها في صورة GB،MB،KB

Express File Sizes in Terms of KB , MB , GB (ls -h or ls --human-readable)

**ملاحظة :**

KB = kilo byte

MB = mega byte

GB = giga byte

استعرضنا في الخيار السابق كيفية عرض الملفات باستخدام الأمر **ls** وتصنيفها مستخدمين الخيار **-S** ولكن كما ذكرت كانت الأحجام المعروضة بالـ **byte** فقط .

ولذلك أتى الخيار **-h** ليكون مُكملا للخيار **-S** حيث يقوم بعرض أحجام الملفات بصورة أفضل من سابقتها وهي عرضها في صورة **giga byte** ، **mega byte** ، **giga byte** ولنرى ذلك بالمثال التالي :

```
oem@ubuntu:~$ ls -laSh ~
```

```
-rw-r--r-- 1 oem oem 349M 2006-10-22 20:01 1.rmvb  
-rw-r--r-- 1 oem oem 18M 2007-01-28 06:05 A3.41.zip  
-rw-r--r-- 1 oem oem 5.4M 2004-01-01 05:46 IAA_2.2.0.2150.zip  
-rw----- 1 oem oem 1.9M 2007-01-24 12:23 ghvh  
-rw-r--r-- 1 oem oem 632K 2007-01-25 11:05 Firefox_wallpaper.png  
-rw-r--r-- 1 oem oem 437K 2007-01-24 09:10 parallel processing for SDR using OMAP.rar  
-rw-r--r-- 1 oem oem 196K 2007-01-29 08:01 .xsession-errors
```

لاحظ أنني قمت بإضافة الخيار **-h** إلى خيارات أخرى لكي نحصل على نتائج أكثر فعالية

وبذلك نكون قد انتهينا من شرح معظم الخيارات المهمة والتي  
يمكن إضافتها للأمر **ls** حتى نحصل على نتائج أكثر  
فعالية ومفيدة في نفس الوقت في تحقيق مطلوبة .

## الأمر pwd :

(Display The Path of Your Current Directory)

وبعد أن انتهينا من الأمر `ls` ، سنبدأ بمشيئة الله في الانتقال إلى مرحلة جديدة من الأوامر ، ونستهل بدايتها مع الأمر `pwd`.

قد تطرقت لذكر الأمر `pwd` سابقا خلال شرحى للأمر `ls` وذكرت الوظيفة التى يقوم بها ، ألا وهى عرض المسار الحالى الذى توجد فيه .

ولنقوم بتوضيح الأمور أكثر نفترض أنك قمت مثلا بتنفيذ عدة أوامر من خلال الطرفية وقمت بالولوج عدة مرات إلى مسارات مختلفة وأثناء أداء هذه المهام لم تتذكر المسار الحالى الذى توجد فيه فما العمل ؟

الحل فى الأمر `pwd` والذى بمجرد كتابته يعرض لك المسار على الفور مثال :

```
oem@ubuntu:~$ pwd
/home/oem
```

والأمر `pwd` هو اختصار للعبارة التالية `print working directory` أو بمعنى قم بطبع الدليل الذى نعمل عليه.

لاحظ : أن كلمة `print` تعنى أظهر الخرج على الشاشة أو بمعنى `print to screen` وليس بمعنى اعطاء أمر ارسال الطباعة للطابعة .

## الأمر cd :

(Change to a Different Directory)

تكلّمنا سابقاً عن كيفية استعراض محتويات أى مجلد بسهولة ويسر مُستخدمين الأمر **ls** وذلك بمجرد تحديد المسار الذى يوجد فيه المجلد أو كتابة ال **relative path** ، ولكن ماذا لو أردنا تغيير المجلد الحالى الذى يوجد فيه إلى مجلد آخر ؟

هنا الأمر **cd** لينجز هذه المهمة حيث يقوم الأمر بتغيير الدليل الذى نوجد عليه إلى أى دليل آخر ( مسار أقصد ) على التوزيعية ومثال على ذلك نفترض أننا نوجد داخل المسار التالى **home/oem/Desktop/** ونريد تغيير هذا المسار إلى آخر وهو **var/cache/** كل ما علينا فعله التالى :

نقوم أولاً بالتأكد أننا على المسار الحالى:

```
oem@ubuntu:~/Desktop$ pwd
/home/oem/Desktop
```

بعد ذلك نقوم بكتابة المسار الذى نريده ومن ثمّ نضغط على زر : **enter**

```
oem@ubuntu:~/Desktop$ cd /var/cache
```

بالفعل تحولنا للمسار المطلوب كما بالشكل :

```
oem@ubuntu:/var/cache$
```

وللتأكد مرة أخرى أننا على المجلد المطلوب:

```
oem@ubuntu:/var/cache$ pwd
/var/cache
```

ولكن ماذا لو أردنا مرة أخرى الرجوع إلى ال **home directory** ؟



ذكرنا سابقاً أن العلامة ~ تساوى بالضبط ال **home directory** ولذا عندما نريد العودة مرة أخرى إلى المجلد **home directory** كل ما علينا فعله استخدام الأمر **cd** فقط مع العلامة ~ ليصبح الأمر بالشكل التالى وبعدها نضغط على الزر : **Enter**

نفترض مثلاً أننا على المسار **var/cache/** ونتأكد من ذلك أولاً :

```
oem@ubuntu:/var/cache$ pwd
/var/cache
```

بعد ذلك نقوم بكتابة الأمر **cd** مع العلامة ~ ونضغط على الزر **Enter** كما يلي:

```
oem@ubuntu:/var/cache$ cd ~
```

ليكون الناتج كالتالى :

```
oem@ubuntu:/var/cache$ cd ~
oem@ubuntu:~$
```

وللتأكد نقوم بكتابة الأمر **pwd** كما يلي :

```
oem@ubuntu:~$ pwd
/home/oem
```

من ضمن المميزات الرائعة التى يمكن إدراجها مع الأمر **cd** هى العلامة - أو ال **dash** والتى تعنى "**previous directory**" أى المجلد السابق أو الدليل السابق الذى كنا عليه .

ولفهم ذلك نفترض أننا كنا على المسار **var/cache/** وقمنا بالتغيير إلى مسار آخر لأداء مهمة معينة وليكن المسار **home/oem/** وبعد أن فرغنا من أداء المهمة أردنا الرجوع إلى المسار السابق ألا وهو **var/cache/** فكل ما علينا فعله التالى :

فى البداية كنا على المسار **var/cache/** كما بالشكل :

```
oem@ubuntu:/var/cache$
```

بعد ذلك قمنا بالتغيير إلى المسار `home/oem/` كما بالشكل :

```
oem@ubuntu:/var/cache$ cd /home/oem
```

نقوم بالضغط على زر `Enter` لنتحول إلى المسار الجديد كما يلي :

```
oem@ubuntu:/home/oem$
```

الآن نريد الرجوع مرة أخرى إلى المسار `var/cache/` دون كتابة المسار والتي تقوم بعمل ذلك هي ال - مع الأمر `cd` ونضغط على الزر `Enter` كما يلي :

```
oem@ubuntu:/home/oem$ cd -  
/var/cache  
oem@ubuntu:/var/cache$
```

وإن شاء الله سوف أكمل الرد في هذه المشاركة وعذرا على التأخير نظرا لظروف أملت  
بى فى الفترة الماضية أسأل الله أن يفرج همنا وهم المسلمين الموحدين آمين

## الأمر touch :

يعتبر الأمر **touch** من الأوامر التى نادراً ما نقوم باستخدامها خلال تعاملنا اليومي مع الطرفية ، ولكن إن شاء الله تعالى من خلال رحلتنا الشيقة فى هذه السلسلة سوف نتعرض له أكثر خلال المسيرة وإظهار الإمكانيات المتقدمة التى يقدمها الأمر لنا .

وقد يعتقد البعض أن الوظيفة الرئيسية التى يقوم بها الأمر **touch** هى تحديث وقت اللوج وتنفيذ المهام على ملف ما بواسطة هذا الأمر ، ولكن سيكون الاعتماد على استخدام هذا الأمر بواسطة الوظائف الفرعية التى يقوم بها والتى هى أهم من الوظيفة الرئيسية !! وسنستعرض ذلك بالتفصيل لاحقاً .

**لاحظ :** تستطيع استخدام الأمر **touch** على ملف ما وتغيير أوقات التعديل والولوج إذا كنت فقط تمتلك تصريح الكتابة على الملف وإلا فسيفشل الأمر فى تنفيذ المهمة!!

والآن نتطرق إلى الجزء العملي المتعلق بالأمر **touch** فى أبسط صورته دون إضافة أى خيارات أخرى معه ولنرى كيفية تعديل الوقت والتاريخ على ملف معين ، ونبدأ أولاً باستعراض الملفات باستخدام الأمر **ls** كما يلي :

```
shuffl@P0w3R:~$ ls -l ~
total 1167060
drwxr-xr-x 7 shuffl shuffl 4096 2007-02-21 19:03 aaaa
-rw----- 1 shuffl shuffl 865301 2007-02-21 18:45 arp.html
-rw-r--r-- 1 shuffl shuffl 31295 2006-11-27 10:07 arp.html.zip
```

هنا قمنا أولاً باستعراض الملفات الموجودة على المجلد **home directory** والتى ترمز العلامة **~** إليه وكانت النتيجة كما بأعلى .

والآن نريد مثلاً تغيير الوقت والتاريخ للمجلد ونلاحظ أن آخر وقت تم التعديل على الملف كانت الساعة **19:03** والتاريخ هو **21-02-2007** كما يلي :

```
shuffl@P0w3R:~$ touch aaaa
shuffl@P0w3R:~$ ls -l ~
total 1167060
drwxr-xr-x 7 shuffl shuffl 4096 2007-02-22 06:17 aaaa
-rw----- 1 shuffl shuffl 865301 2007-02-21 18:45 arp.html
-rw-r--r-- 1 shuffl shuffl 31295 2006-11-27 10:07 arp.html.zip
```

كما ترون تم تحديث الأوقات الخاصة بالمجلد **aaaa** لتُصبح هى الأوقات الفعلية الحالية لدينا .

ويتضح من ذلك أن الأمر **touch** يقوم بالتعديل على كل من الملفات والمجلدات على حد سواء بمعنى إذا كان لديك ملفات نصية أو أى نوع ترغب فى تغيير وقت التعديل عليه فما عليك إلا كتابة الأمر مصحوباً بإسم الملف المراد .

ولكن قد يتبادر إلى الأذهان سؤالاً بخصوص الأمر ، ألا وهو هل الأمر **touch** يقوم بتغيير وقت التعديل على الملفات إلى الأوقات الحالية فقط أم أى وقت وتاريخ أريده ؟؟

تعلمنا فيما مضى أن لينوكس عالمياً آخر ، ففيه تمتلك القدرة على تطويع أى شئ تريده لأنه نظام مرن بما فيه الكفاية ليفعل لك ما تريده .

ولذلك تستطيع بالطبع عند استخدام الأمر **touch** لتغيير أوقات التعديل على ملفات تريدها أن تغير أوقات التعديل إلى الأوقات التى تختارها أنت وبالتالي كل ما نحتاجه هو إضافة خيار إلى الأمر **touch** ليقوم بهذه المهمة وهذا الخيار هو **-t** والذى عنده إضافته للأمر يقوم بتغيير الأوقات التى تختارها بسلسلة ومرونة رائعة والجدول التالى يوضح الصيغة العامة لتغيير أوقات الملفات كالتالى :

**[CC]YY]MMDDhhmm[.ss]] :**

character	Meaning
CC	First two characters of a four-digit year
YY	Two-digit year: * if from (00-68) , assumes that first two digits are 20 * if form (69-99) , assumes that first two digits are 19 * if nothing , assumes current year
MM	Month (01-12)
DD	Day (01-31)
hh	Hour (01-23)
mm	Minute (00-59)
ss	Second (00-59)

والآن قد رأيت الجدول السابق لكيفية وضع القيم الخاصة بالتاريخ والوقت سواء كان التاريخ سنين ، شهور ، أيام وكذلك الوقت ساعات ، دقائق ، ثوان .

والآن إلى ملاحظة مهمة وهى متعلقة بالأرقام التى ليست على صورة ثنائية بمعنى ، إذا كان الرقم لديك مكون من خانة واحدة فقط كرقم 1 أو 2 أو 3 إلخ

ينبغي أن تقوم بكتابة صفر قبل الرقم حتى تقوم الطرفية بفهم صيغة تغيير الوقت والتاريخ بشكل سليم .

والآن نتطرق إلى بعض الأمثلة المتعلقة بالأمر **touch** بعد إضافته إلى الخيار **-t** لكي نقوم بوضع تاريخ ووقت معين من اختيارنا كالتالي :

مثال : قمنا بإنشاء ملف جديد باسم **wireless.log** والآن نريد استعراض كافة المعلومات الخاصة بالملف بداية من التصاريح الموضوعة عليه ، واستعراض المالك له والمجموعة كذلك آخر وقت تم التعديل فيه على الملف كالتالي :

```
shuffl@P0w3R:~/Desktop$ ls -l wireless.log
-rw-r--r-- 1 shuffl shuffl 0 2007-04-16 18:17 wireless.log
```

هنا في المثال الوقت الذي تم آخر تعديل فيه على الملف هو الساعة **18:17** والتاريخ هو **16-04-2007**.

نريد الآن تغيير ذلك الوقت والتاريخ لتكون الساعة **00:00** ويكون التاريخ **31-12-1992** ويكون ذلك كالتالي :

```
shuffl@P0w3R:~/Desktop$ touch -t 9212310000 wireless.log
shuffl@P0w3R:~/Desktop$ ls -l wireless.log
-rw-r--r-- 1 shuffl shuffl 0 1992-12-31 00:00 wireless.log
```

كما ترون تمت المهمة بنجاح وقمنا بتعديل كلاً من الوقت والتاريخ اللذان قمنا باختيارهما .

نأتي إلى مثال آخر ونريد تغيير الوقت إلى الساعة **02:34** والتاريخ هو **17-05-2034** عن طريق عمل التالي :

```
shuffl@P0w3R:~/Desktop$ touch -t 3405170234 wireless.log
shuffl@P0w3R:~/Desktop$ ls -l wireless.log
-rw-r--r-- 1 shuffl shuffl 0 2034-05-17 02:34 wireless.log
```

وبالفعل تم لنا ذلك أيضاً !! .

و نأتي إلى مثال أخير وهو تغيير الوقت إلى الساعة **17:03** والتاريخ إلى **19-10-2007** أي السنة الحالية :

```
shuffl@P0w3R:~/Desktop$ touch -t 10191703 wireless.log
shuffl@P0w3R:~/Desktop$ ls -l wireless.log
-rw-r--r-- 1 shuffl shuffl 0 2007-10-19 17:03 wireless.log
```

والآن نشرح كيفية وضع القيم في الأمثلة السابقة ، في المثال الأول أردنا تغيير التاريخ إلى 31-12-1992 قمنا أولا بكتابة السنة المرادة من اليسار وهى 92 ثم الشهر المراد وهو 12 ثم اليوم المراد وهو 31

ولكن قد يستوقفنى البعض ويقول لى لما لم تكتب صيغة السنة كاملة من أربعة أرقام لتصبح 1992 بدلا من 92 أو بصيغة أخرى كيف علمت الطرفية أن السنة التى أريدها هى 1992 وليست مثلا 2099 أو شىء من ذاك القبيل ؟؟

لو تتذكرون بأعلى وضعت جدول يحدد صيغ وضع الوقت والتاريخ فالبدائية كانت من الحرفين CC واللذان يرمزان إلى أول رقمين من أربعة أرقام من السنة وبعدهما الحرفان YY واللذان يرمزان إلى الرقمين الآخرين من أرقام السنة وذكرت أن للحرفين YY ثلاث حالات :

**الأولى :** أن يكون الرقمان فى المدة من سنة رقمها 00 إلى سنة رقمها 68 وهنا تعتبر الطرفية أن أول رقمين هما 20 وبالتالي يصبح نطاق الأرقام من 00 إلى 68 خاص بالألفية الثانية أو من سنة 2000 إلى سنة 2068 وهذه الحالة تمثل المثال رقم واحد.

**الثانية :** أن يكون الرقمان فى المدة من سنة رقمها 69 إلى سنة رقمها 99 وهنا تعتبر الطرفية أن أول رقمين هما 19 وبالتالي يصبح نطاق الأرقام من 69 إلى 99 خاص بالقرن التاسع عشر أو من سنة 1969 إلى سنة 1999 وهى الحالة تمثل المثال رقم اثنان.

**الثالثة :** إذا لم تضع أرقاما فتعتبر الطرفية أن السنة المرادة هى السنة الحالية وهذه الحالة تمثل المثال رقم ثلاثة.

**توضيح :** كما ذكرت أعلى أن للسنة تكتب أربعة أرقام أى من الممكن أن تكتب سنة 1992 بهذه الشكل أو أن تكتب الرقمين 92 فقط ، كذلك يمكنك كتابة سنة 2022 هكذا أو أن تكتب السنة مكثفيا بالرقمين 22 فقط.

وبديهى يكون الأمر بالمثل بالنسبة لكل من وضع الشهر ثم اليوم ثم الساعة وهكذا وأظن لا حاجة لشرح هذه الأمور فالأمر قد اتضح .

تعلمنا الآن أن الأمر touch يقوم بمهام رائعة تخص تعديل الوقت والتاريخ ولكن مازال الأمر يحوى الكثير فى جعبته !! نعم، ولكن حقيقةً المهام السابقة التى قام بها الأمر ليست هى المهام الرئيسية التى يقوم بها الأفراد عادة فالأمر touch له استخدام أكثر إثارة من المهام السابقة وهذا الإستخدام يكمن فى أنه يعطيك القدرة على إنشاء ملفات فارغة !!

قد يندهش البعض لذلك ولكن تعال معي لترى كيف يمكن عمل ذلك كالتالى :

```
shuffl@P0w3R:~/Desktop$ touch test.txt
shuffl@P0w3R:~/Desktop$ ls -l

-rw-r--r--  1 shuffl shuffl    0 2007-04-17 09:00 test.txt
-rw-r--r--  1 shuffl shuffl    0 1969-10-19 17:03 wireless.log
-rw-r--r--  1 shuffl shuffl 493 2006-10-02 18:08 xchat.desktop
```

نعم ! لا تندهش فعلا قام الأمر بإنشاء ملف جديد وهو الملف **test.txt** وبالتالى نستنتج من ذلك أنه يمكنك إنشاء أى ملف بأى صيغة تريدها باستخدام الأمر **touch** و كل ما عليك عمله هو مجرد فقط كتابة الأمر متبوعاً بإسم الملف الذى تود إنشاؤه.



# الأمر mkdir :

( Create a New Directory )

استعرضنا فى الأمر السابق **touch** بعض المهام الجيدة التى يقوم بها من تعديل وقت وتواريخ الملفات إلى إمكانية إنشائه ملفات فارغة وبعد أن تعرفنا على هذه المهام نريد التقدم خطوة أخرى وهى كيفية إنشاء مجلدات ؟؟

هنا يأتى دور الأمر **mkdir** والذي هو اختصار كلمتى **make directory** ، وكل ما عليك عمله لإنشاء مجلد فارغ هو مجرد كتابة الأمر **mkdir** متبوعاً بإسم المجلد بالشكل التالى :

```
shuffl@P0w3R:~/Desktop$ mkdir test
shuffl@P0w3R:~/Desktop$ ls -l

drwxr-xr-x  8 shuffl shuffl  4096 2007-03-11 07:56 terminologies
drwxr-xr-x  2 shuffl shuffl  4096 2007-04-17 09:32 test
-rw-r--r--  1 shuffl shuffl    0 2007-04-17 09:00 test.txt
```

**ملحوظة :** معظم التوزيعات عند إنشاء مجلدات باستخدام الأمر **mkdir** يكون التصريح الافتراضى للمجلد هو القراءة والكتابة والتنفيذ لصاحب الجلد أو المالك ، والقراءة والتنفيذ لكلاً المجموعة وأى مستخدم آخر على التوزيعة وقد بينت هذا الأمر فى موضوع التصاريح على الملفات والمجلدات بأنظمة **unix-like** فلتراجع على هذا الرابط:

<http://www.linuxac.org/forum/showthread.php?t=2437>

والأمر **mkdir** ذكى بما فيه الكفاية ، بمعنى حينما تود إنشاء ملف يحمل نفس الإسم فى مسار معين على الفور يُخبرك الأمر أنه لا يستطيع تكرار المجلد لأنه موجود مسبقاً ولنرى ذلك كالتالى :

```
shuffl@P0w3R:~/Desktop$ mkdir test
mkdir: cannot create directory `test': File exists
```

الآن تعلمنا خطوة أخرى وهى إنشاء مجلد فارغ باستخدام أمر جديد وهو الأمر **mkdir** ، ولكن كما قلنا فى البداية أن لكل أمر من أوامر الطرفية بعض الخيارات التى يتم إضافتها إليه لتكسب الأمر قوةً ومرونةً حقيقية .

وهذا ما يحدث للأمر **mkdir** فَيُوجد لدى الأمر بعض الخيارات المهمة نستعرض البعض منها على سبيل المثال :

# mkdir -p

(Create a New Directory and Necessary Subdirectories)

أول خيار نستعرضه مع الأمر **mkdir** هو الخيار **-p** والذي يرمز إلى **--parents** ، حيث يمكنك هذا الخيار عند دمجه مع الأمر من إنشاء المجلد الذي توده بالإضافة إلى مجلدات فرعية تحت هذا المجلد دون الحاجة إلى إنشاء كل مجلد فرعي على حده !!

دون أن نكثر الكلام نأخذ مثالاً عملياً على ذلك:

**مثال :** تود إنشاء مجلد مثلاً تريد فيه بعض الصور وتود أيضاً إنشاء مجلدات فرعية تحت هذا المجلد كل منها يشمل قسم معين ، نستطيع عمل ذلك كالتالي :

```
shuffl@P0w3R:~/Desktop$ mkdir -p pictures/personal/natural
shuffl@P0w3R:~/Desktop$ ls -l

drwxr-xr-x  3 shuffl shuffl  4096 2007-04-17 09:48 pictures
drwxr-xr-x  8 shuffl shuffl  4096 2007-03-11 07:56 terminologies
-rw-r--r--  1 shuffl shuffl    0 2007-04-17 09:00 test.txt

shuffl@P0w3R:~/Desktop$ cd pictures/
shuffl@P0w3R:~/Desktop/pictures$ ls -l

drwxr-xr-x 3 shuffl shuffl 4096 2007-04-17 09:48 personal

shuffl@P0w3R:~/Desktop/pictures$ cd personal/
shuffl@P0w3R:~/Desktop/pictures/personal$ ls -l

drwxr-xr-x 2 shuffl shuffl 4096 2007-04-17 09:48 natural
```

كما ترى قام الأمر باستخدام الخيار **-p** بإنشاء المجلد الرئيسي وهو **pictures** ، ثم قام بإنشاء مجلد فرعي تحت المجلد **pictures** وهو المجلد **personal** ، ثم قام أخيراً بإنشاء مجلد فرعي آخر تحت المجلد الفرعي **personal** وهو المجلد **natural**.

# mkdir -v

(Find Out What mkdir Is Doing As It Acts)

الخيار الذي سنتحدث عنه الآن مُفيد في بعض الحالات أثناء قيامك أداء مهمة معينة باستخدام أمر مُعين يمكن إضافة هذا الخيار معه ، بمعنى حينما قمنا في النقطة السابقة والتي تحدثنا فيها عن الخيار **-p** لم نعلم كيف بدأ الأمر **mkdir** في إنشاء المجلدات .

هل بدأ أولاً بإنشاء المجلد الرئيسى ، أم بدأ أولاً بإنشاء آخر مجلد فرعى وهكذا ، ولذلك الخيار **-v** يتيح لنا إمكانية معرفة أي المجلدات تم البدء في إنشائها أولاً والخيار **-v** يرمز إلى **--verbose**

وبالتالى نستطيع تكرار المثال السابق ولكن الجديد هنا إضافة خياران في نفس واحد هما الخيار **-p** والخيار **-v** كالتالى :

```
shuffl@P0w3R:~/Desktop$ mkdir -vp pictures/personal/natural
```

```
mkdir: created directory `pictures'
mkdir: created directory `pictures/personal'
mkdir: created directory `pictures/personal/natural'
```

كما تلاحظ قام الأمر **mkdir** أولاً بإنشاء المجلد **pictures** ثم قام بعد ذلك بإنشاء المجلد **personal** داخل المجلد **pictures** ثم قام أخيراً بإنشاء المجلد **natural** داخل المجلد **personal** والذي هو بدوره داخل المجلد **natural** .

ملحوظة : لا يفضل استخدام الخيار **-v** مع أوامر فك الضغط عن الملفات حيث يقوم الخيار بمضاعفة الوقت اللازم لعملية الفك.

إلى هذا الحد نكتفى في هذه المشاركة ونُكمل في المشاركات القادمة بإذنه تعالى وما أسألکم هو الدعاء لى بظاهر الغيب فقط.

وأتمنى من الإخوة عدم الرد في هذا الموضوع حيث أنى قمت بالتنبيه في أول الموضوع أن الردود والاستفسارات يتم وضعها في هذا الموضوع:

<http://www.linuxac.org/forum/showthread.php?t=1603>

تم بحمد الله تعالى

الاثنين الموافق 1428/12/15 هـ

وهذا الكتاب سوف أستمر في متابعة هذا الموضوع واطرح التحديث في هذا الملف

والشكر موصول للأخ أبو عبدالرحمن

واتمنى أن يكمل الموضوع حيث أنه مهم جداً

وأسأل الله تعالى بأن يوفقه ويرزقه من حيث لا يحتسب

والسلام عليكم ورحمة الله

اخوكم : الطارق