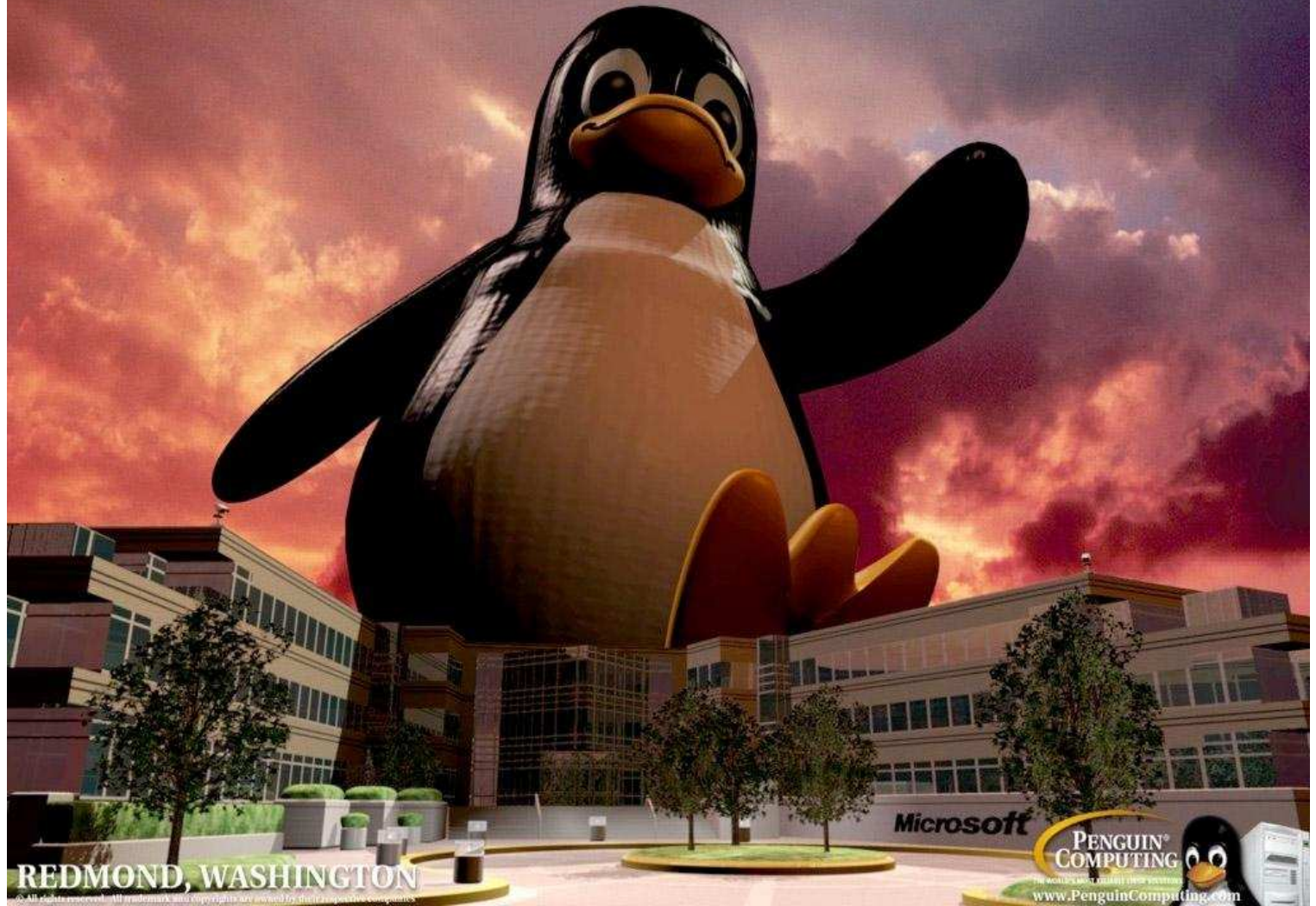


بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ  
مَنْ كَفَرَ بِاللَّهِ مِنْ بَعْدِ إِيمَانِهِ  
سَاءَ مَا يَحْكُمُهُ يَوْمَ الْقِيَامِ

# نظام الأوامر

وَقُلْ رَبِّيَ زِدْنِي عِلْمًا

Good evening Mr. Gates, I'll be your server today!



## مقدمة

بمجرد دخولك للنظام فإن مفسر الأوامر command interpreter يبدأ في عمله وهذا المفسر يدعى الصدفة shell .

ملاحظة:  
الصدفة shell توجد منها عدة أنواع سنعرف بعضها فيما بعد في فصل الصدفة shell .

و عندما تتعامل مع أي نظام فلا بد من أداة تمكنك من التعامل معه. وهذه الأداة تدعى أوامر النظام. فأنت من خلال هذه الأوامر تستطيع التعامل مع النظام وأنت بهذه الأوامر فأنت قد ملكت مفاتيحه، وبما أنك ملكت مفاتيح شيء فأنت قد ملكت الشيء.  
ونظام الأوامر في اللينوكس مريح وغير معقد من حيث كتابته وأيضا من حيث تذكره.

فيما بعد:  
ستتعرف على خواص نظام الأوامر في لينوكس، حيث أنه غير معقد وبسهل تذكره خلال هذا الفصل إن شاء الله

والأوامر عبارة عن سلسلة من الحروف على شكل كلمات متتالية وهذه الكلمات يفصل بينها مسافة أو عدة مسافات (space or Tab). وأول كلمة هي عبارة عن الأمر والذي يليها يسمى معاملات وخيارات الأمر.  
وأنت من خلال الأوامر تستطيع أن تهئي هذه الأوامر بحيث أنها تنفذ الذي تريد أنت تنفيذه. والأوامر في نظام لينوكس لها ميزتين:

- تستطيع أن تحدد أو تعيد توجيه مدخلات ومخرجات الأوامر.
- تستطيع أن تستخدم خيارات الأمر بحيث أن تمكنك من استخدام الأمر بصورة مناسبة.

فعندما تقوم بكتابة أمر ما فعندما تقوم بالضغط على زر enter فإن النظام لينوكس يأخذ هذا الأمر وينفذ (يعالج) هذا الأمر في المكان المخصص لذلك ثم يأخذ المخرجات الخاصة بمعالجة هذا الأمر ويعرضها لك سواء على الشاشة أو في ملف.

فيما بعد:  
سوف تعرف كيف يمكن أن تعرض مخرجات أمر ما في ملف في هذا الفصل إن شاء الله.

ملاحظة:  
خيارات الأمر فائدتها في أنها تنفذ (تعالج) لك الأمر بصورة محددة و أكثر تخصصا.

## خيارات الأمر

كما وضعنا سابقا عن خيارات الأمر فإنها تجعل لك الأمر ينفذ المهمة بصورة أدق وأكثر تحديدا. وسوف نأخذ هنا مثال لنوضح فيها استخدام خيارات الأمر.

مثال:

سنأخذ هنا مثال على أمر شائع استخدامه في اللينوكس وهو الأمر ls حيث يقوم هذا الأمر بسرد الملفات الموجودة في hard drive الخاصة بالنظام لينوكس. قم الآن بكتابة الأمر ls ثم اضغط enter، وانظر للنتيجة، لن تجد شيئاً، لماذا؟ لأن الأمر ls فقط يقوم بإظهار الملفات التي لا تبدأ بنقطة، أما لو استخدمنا خيارات الأمر ls فسوف تتلف النتيجة، جرب الآن هذا الأمر ls -a حيث أن -a يقوم عرض جميع الملفات حتى تلك التي تبدأ بنقطة. تعتبر -a أحد خيارات الأمر ls ولا توجد بين الحرف a والعلامة - مسافة ولكن توجد بين ls و -a مسافة. ويوجد للأمر ls العديد من خيارات الأمر. سوف تعرف لاحقاً كيف تعرف خيارات أمر ما وكيف تستخدمه في هذا الفصل إن شاء الله. يوجد خيار أمر آخر للأمر ls وهو -l حيث يقوم بعرض تفاصيل عن الملف، وتستطيع أن تدمج الخيارين -a, -l مع بعضهما بالطريقة التالية:

ls -a -l أو ls -la أو ls -al

فهذه الطرق الثلاثة مثل بعضهم في التنفيذ.

انتبه:

قد يحدث أن يكون اسم الملف طويلاً وبالتالي فإن النظام سيعرض بقية المعلومات عن الملف في السطر التالي مباشرة.

ملاحظة:

فائدة الإشارة - أنها تخبر النظام أن ما بعد الإشارة - عبارة عن خيارات أمر.

انتبه:

لا تستطيع دمج خيارات الأمر مع بعضهم في كل أمر كما فعلنا في الأمر ls -al حيث أن هناك خيارات أوامر تتكون من أكثر من حرف واحد.

ملاحظة:

عندما تقوم بتنفيذ الأمر ls -al فإنه يعرض لك كل الملفات بتفاصيلها بالترتيب الأبجدي للملفات، أما لو وضعت خيار الأمر -t فإنه سوف يعرض لك الملفات حسب تاريخ الإنشاء أو حسب تاريخ آخر تعديل.

فكرة:

يمكن استخدام الأمر ls بالصورة التالية: ls -alt

انتبه:

يمكن أن يوجد خيارين لأمر ما متشابهان في الحرف ومختلفان في الصفة، بحيث أنه يمكن أن يوجد لديك خيار أمر هكذا -r ويوجد أيضا خيار أمر هكذا -R وكل منهما ينفذ الأمر بصورة مختلفة عن الآخر، فلا بد أن تنتبه ما إذا كان الحرف كبيرا أم صغيرا.

ملاحظة:

لاحظت أن خيارات الأوامر تتكون من حروف ولكن هذه الحروف لها دلالة بحيث يسهل عليك تذكر خيارات الأوامر فمثلا عندما تستخدم -a فحرف a يدل على كلمة all files و -l تدل على long list و -t تدل t على sort by time وهكذا.

فيما بعد:

سنعرف في هذا الفصل كيف نبحث عن أمر معين وكيف لنا أن نستخدم هذا الأمر وكيف يمكن أن نستخدم خيارات أي أمر ، إن شاء الله.

## معاملات الأوامر

الأوامر في لينوكس يمكن أن تستخدم معاملات وهذه المعاملات لا تعتبر من خيارات الأمر وهذه المعاملات يمكن أن تكون اسم ملف أو فهرس ولا يسبق بعلامة - . فمثلا عندما تستخدم الأمر ls فإنه يقوم بعرض الملفات الموجودة في الفهرس الحالي، فتستطيع وأنت في أحد الفهارس أن تعرض ملفات لفهرس آخر وذلك حسب الشكل التالي:

```
ls /directoryname
```

فعندها تقوم أنت باستبدال directoryname باسم الفهرس الذي تريد أن تعرض ملفاته.  
فمثلا قم بتنفيذ الأمر التالي:

```
ls /bin
```

ولاحظ النتائج التي ستظهر على الشاشة.

فكرة:

قم بوضع خيارات الأمر ls مع ls /bin ، مثلا قم بتنفيذ الأمر ls -al . /bin

تستطيع أيضا أن تستخدم الأمر ls من خلال الشكل التالي:

ls filename

حيث تقوم أنت بوضع اسم الملف الذي تريد أن تعرض معلومات عنه حيث ستستبدل filename باسم الملف، وتستطيع أيضا أن تستخدم خيارات الأمر ls بالطريقة التالية: ls -al filename .

## إعادة توجيه المدخلات والمخرجات واستخدام الأنايب

\*\*\*\*\*

الكثير من الأوامر في لينوكس تدعك لتحديد الملف والفهرس الذي سيعمل عليه. كما وضعنا سابقا من خلال الأمر ls -l /bin .

عندما تقوم بتنفيذ أمر ما فإن هذا الأمر يأخذ المدخلات من خلال لوحة المفاتيح ويعرض المخرجات على الشاشة، وتستطيع أن توجه مخرجات أي أمر لتكون مدخلات لأمر آخر وتستطيع أيضا أن تجعل المخرجات تظهر في ملف ما. وحتى تكون مخرجات أمر ما مدخلات لأمر آخر استخدم تلك العلامة (ستجد تلك العلامة موجودة في أعلى زر العلامة \ ولكن عليك أن تضغط المفتاح shift) والشكل العام لاستخدام | هو كالتالي:

command\_1 | command\_2

من الأفضل:

يمكن أن لا يكون بين command\_1 والعلامة | أي مسافة ولكن من المفضل وضع مسافة حتى يسهل قراءة الأمر، وسوف تشعر بأهمية ذلك عندما نأخذ الصدف shell إن شاء الله.

من الأشياء التي يتيحها لينوكس لك من خلال إعادة توجيه المدخلات والمخرجات هو أنه يمكن أن تضع مخرجات أمر ما في ملف بدلا من أن يظهر على الشاشة. وتعتبر هذه العملية مفيدة جدا ولها استخدامات عدة ومنها أنك تحتفظ بنسخة من المخرجات لتستخدمها في وقت لاحق ، وأيضا يمكن أن تستخدمها مع الأوامر التي تأخذ وقتا طويلا في التنفيذ.

وحتى تستطيع أن ترسل مخرجات أمر ما في ملف قم باستخدام الإشارة التالية > ، فمثلا يمكن أن تضع مخرجات الأمر ls -l /bin في ملف يدعى مثلا test وذلك من خلال الأمر التالي:

ls -l /bin > test

من الأفضل:

من الأفضل أن توجد مسافة حول العلامة > ، ولكنها غير ضرورية ولكنها تسهل قراءة الأمر.

وعندما تقوم بتنفيذ الأمر السابق فإن الملف test سيتم إنشاؤه في الفهرس الحالي. وحتى تشاهد محتويات الملف قم بتنفيذ الأمر التالي:

more filename

حيث ستقوم أنت باستبدال كلمة filename بالملف الذي تريد أن ترى محتوياته، وهو في هذه الحالة الملف test، وبالتالي سيكون الأمر كالتالي:

more test

انتبه:

عندما تستخدم الأمر `ls /bin > test` فلو كان الملف `test` موجود أصلا فإن لينوكس سيقوم بمسح محتويات الملف السابقة ووضع المحتويات الجديدة فخذ حذرك وخاصة عندما تدخل إلى النظام ك `root`، وحتى تتأكد أنه لا يوجد ملف اسمه `test` في الفهرس الذي توجد فيه قم بكتابة الأمر التالي: `ls -l test` حيث سيعرض لك معلومات عن الملف إذا كان موجودا ولن يعرض معلومات إذا لم يكن موجودا.

طبعا فإن لينوكس لن يترك هذه المشكلة بدون حل، حيث أنك ومن خلال تلك العلامة `>` تخبر لينوكس أن يضع مخرجات الأمر في نهاية الملف بحيث يحتفظ بمحتويات الملف الأصلية.

فكرة:

جرب استخدام هذه العملية `>>` ولاحظ النتائج.

وإذا كان الملف كبير قم باستخدام الأمر `tail test` حيث أن هذا الأمر يعرض لك أواخر سطور الملف `test`.

فيما بعد:

سوف تعرف تفاصيل أكثر وأكثر عن إعادة وجيه المدخلات والمخرجات في فصل `bash` إن شاء الله.

## أشياء لا بد أن تعرفها عند التعامل مع أوامر لينوكس

عندما تتعامل مع أوامر لينوكس هناك عدة رموز لا بد أن تعرفها فهي تحدد لك خيار الأمر أو المعامل الذي لا بد أن تستخدمه أو خيار الأمر والمعامل الذي يمكن أن تستخدمه أو أن لا تستخدمه. وعندما تبحث عن أمر ما فإنك ترى هذا الأمر ومعه كل خياراته ومعاملاته الخاصة به ويترك لك لينوكس حرية الاختيار في التعامل مع الأمر. وإليك هذه القواعد في التعامل مع رموز الأوامر:

- 1- أي أمر يوجد معه نص مكتوب وهذا النص ليس داخل هذه الرموز `[ ]` `<>` { } فإن هذا النص يكتب كما هو.
- 2- أي نص مكتوب داخل `[ ]` فإنه نص اختياري، يعني يمكن كتابته أو تجاهله، ولا تضع مع النص `[ ]`.
- 3- إذا وجدت الرمز `<>` فهذا معناه أن تستبدل النص الذي يوجد داخل هذا الرمز بنص مناسب (اسم أو قيمة)، فمثلا الأمر `more <filename>` العامة له كالتالي: `more <filename>` فهذا معناه أن تستبدل `filename` باسم الملف الذي تريد أن تعرف محتوياته كما فعلنا مع الملف `test` سابقا من خلال الأمر التالي: `more test`. ولا تستخدم الرمز `<>`.
- 4- عندما تجد أشياء داخل { } فإنه عليه أن يختار أحد هذه القيم. فلو مثلا كانت الصيغة العامة لأمر ما كالتالي: `command {a|b}` فإن عليه أن يختار `a` أو `b`. فيكون الشكل لها كالتالي: `command -a` أو `command -b`

- 5- الرمز (...) معناه وهلم جرا، وهذا يستخدم مع المعاملات مثل لو طلب منك إدخال أسماء ملفات فيمكن أن تدخل اسم ملف واحد أو ملفين أو ثلاث ملفات وكذا.
- 6- يمكن أن ترى رمزين أو ثلاث مدمجين مع بعضهما فمثلا لو رأيت الأمر التالي: [ <filename>... ] more فهذا معناه أن يمكن أن يكتب اسم الملف أو يتجاهله وإذا كتبه فإنه يمكن أن يكتب ملف أو ملفين أو ثلاث ملفات.

انتبه:  
عندما يتم دمج أكثر من رمز فعليك أن تتعامل مع الرموز من الخارج إلى الداخل وليس العكس، كما فعلنا منذ قليل مع الأمر  
more [ <filename>... ]

## لينوكس يقدم لك المساعدة في الأوامر من خلال man

يقدم لك اللينوكس وسيلة المساعدة لكي تعرف كيف تستخدم أمر ما وذلك من خلال البرنامج man (كلمة man اختصار لكلمة manual ) ، وحتى تعرف كيف تستخدم أمر ما قم باستخدام الصيغة التالية: man <command> . فلو مثلا أردنا أن نعرف كيف نستخدم الأمر passwd قم كتابة الأمر التالي: man passwd وسوف تظهر النتيجة التالية:

COMMAND(1)

Linux Programmer's Manual

COMMAND(1)

NAME

command - summary of what command does

SYNOPSIS

&lt;complete syntax of command in the standard Linux form&gt;

---

## DESCRIPTION

More verbose explanation of what "command" does.

## OPTIONS

Lists each available option with description of what it does

## FILES

lists files used by, or related to, command

## SEE ALSO

`command_cousin(1)`, `command_spouse(1)`, etc.

## BUGS

## There are bugs in Linux commands??

## AUTHOR

J. S. Goobly (goobly@hurdly-gurdly.boondocks)

Linux 1.2

22 June 1994

1

من الملاحظ أن شرح الأوامر في man يتم فهمها بسهولة ولكن الشرح مختصر جدا حتى لا يأخذ الشرح مساحة كبيرة خصوصا أن أوامر لينوكس كثيرة إلى حد ما.

ملاحظة:  
قم بكتابة الأمر التالي: man ls ولاحظ كثرة خيارات الأمر ls ولاحظ طول الشرح الذي أخذه man!!!!!! .

فكرة:  
إذا واجهك أي أمر غريب فما عليك إلا أن تستدعي man .

في بعض الأحيان تعرف ما تريد أن تفعله ولكنك لا تعرف الأمر الذي يقوم بهذا العمل، فما عليك إلا أن تكتب هذا النموذج `man -k <keyword>` حيث أن keyword تدل على العمل الذي تريد أن تقوم به ولكن لا بد أن يكون وصف العمل مختصر جدا، وسوف يظهر لك man عدة أوامر التي تحتوي على هذا الوصف الذي كتبتته أنت، مع وصف ملخص لعمل كل أمر حتى تختار الذي يناسبك.

واليك هذا المثال:  
لنفترض أنك تريد أن تبحث عن أمر له علاقة بكلمة manual ، قم بكتابة الأمر التالي: `man -k manual` وسوف يخرج لك اللينوكس نتيجة مشابهة لذلك تقريبا.

man (1) - format and display the on-line manual pages .br manpath--

## determine user's search path for man pages

whereis (1) - locate binary, source, and manual page files for a command

xman (1) - Manual page display program for the X Window System

انتبه:

عندما تقوم بالبحث عن أمر يجب عليك أن تكتب كلمات محددة ولها معنى ولا تكن كلمات عامة، بمعنى لا تكتب كلمة file للبحث عن أمر ما يخضها لأن المخرجات سوف تكون كثيرة جداً.

استخدام الرمزين \*,?

يقدم لك الرمزين \*,? فائدة كبيرة في عملية البحث عن الملفات والفهارس. حيث أن الرمز \* يدل على أي عدد من الحروف أما الرمز ? فيدل على حرف واحد فقط، وإليك هذا المثال:

فلو كتبنا الأمر التالي: `ls /bin/c*` فهذا معناه أن يقوم النظام بالبحث عن ملفات أو فهارس يبدأ اسمها بالحرف c بأي عدد من الحروف بعد الحرف c .

أما لو كتبنا الأمر التالي: `ls /bin/c*t` فإنها تعني أن يقوم النظام بالبحث عن ملفات وفهارس تبدأ بالحرف c وتنتهي بالحرف t بأي عدد من الحروف بينهما.

أما لو كتبنا الأمر التالي: `ls /bin/c?` فإنها تعني أن يقوم النظام بالبحث عن ملفات أو فهارس يتكون من حرفين فقط وتبدأ بالحرف c .

انتبه:

الرمزين \*,? كما شرحناهما منذ قليل لا يكونا إلا مع أسماء الملفات والفهارس أي أنك لا يمكن أن تستخدمها مع الأوامر، يعني أنك لو كتبت الأمر التالي `man pass*` فهذا ليس معناه أن برنامج man سيبحث لك عن أوامر تبدأ بالحرف pass بل سيظهر لك رسالة خطأ.

انتبه:

خذ حذرك عندما تستخدم الرمزين ?,\* مع الأوامر الخطيرة مثل الأمر الذي يقوم بحذف الملفات والفهارس ، فعليك أن تنتبه جيدا قل استخدام الرمزين ?,\* مع تلك الأوامر.

انتبه:

نظام لينوكس مليء بالأوامر ومليء بأشياء أخرى كلها ممتعة وذكية ومليئة بالأفكار، وأنا من خلال شرحي هذا أعطيك الأفكار الأساسية للتعامل مع أي جزء من أجزاء النظام سواء في هذا الفصل الذي قمت بشرحه (نظام الأوامر) أو الفصول الأخرى التي تليه ، ولا أعطيك كل الأفكار الممكنة لأنها كثيرة جدا، وأيضا حتى أعطيك مجالاً لتفكر أنت لتأتي بأفكار جديدة حاول وجرب وكلما توصلت إلى فكرة جديدة قم بتدوينها على الفور حتى لا تنساها. لا تتوقف عند حد الشرح فقط لأنها ليست كل شيء. ولكن يجب أن يكون تفكيرك منطقي ومبني على أساس نظام لينوكس وحاول أن تتوقع النتائج مسبقا.