

Apuntes Básicos de Comandos de Unix

Resumen de comandos vistos y por ver:

- Comando *pwd*

| | |
|-----------------------|--|
| \$ <i>pwd</i> <Enter> | Permite saber en que directorio estás posicionado. |
|-----------------------|--|

- Comando *ls*

| | |
|-------------------------------|--|
| \$ <i>ls</i> <Enter> | Despliega el contenido del directorio actual. |
| \$ <i>ls -l</i> <Enter> | Despliega el contenido del directorio actual mostrando detalles. |
| \$ <i>ls -a</i> <Enter> | Despliega el contenido del directorio actual, incluyendo archivos "ocultos". |
| \$ <i>ls -la</i> <Enter> | Despliega el contenido del directorio actual mostrando detalles, incluyendo archivos "ocultos". |
| \$ <i>ls -la /etc</i> <Enter> | Despliega el contenido del directorio /etc , mostrando detalles, incluyendo archivos "ocultos". |

- Comando *cd*

| | |
|-----------------------------------|---|
| \$ <i>cd</i> <Enter> | Sin importar el directorio actual, se cambia al directorio "personal" del usuario. |
| \$ <i>cd /etc</i> <Enter> | Cambia al directorio /etc . |
| \$ <i>cd programacion</i> <Enter> | Cambia al directorio ./programacion . (Es decir, a un subdirectorio contenido en el directorio actual. |

- Comando *mkdir*

| | |
|---|---|
| \$ <i>mkdir /tmp/programación</i> <Enter> | Sin importar el directorio actual, se crea un subdirectorio llamado programacion dentro del directorio /tmp . |
| \$ <i>mkdir programacion</i> <Enter> | Dentro del directorio actual es creado un subdirectorio llamado programacion . |

- Comando cat

| | |
|---|--|
| \$ cat README <Enter> | Si el archivo README existe en el directorio actual, lo despliega del principio al final. |
| \$ cat /user/ie84adh/prueba.c <Enter> | Si el archivo prueba.c existe en el directorio /user/ie84adh , despliega su contenido. |

- Comando more

| | |
|--|--|
| \$ more README <Enter> | Si el archivo README existe en el directorio actual, lo despliega por pantalla. Es decir, al llegar al final de la pantalla, espera que el usuario presione la barra de espacio para continuar. |
| \$ more /user/ie84adh/prueba.c <Enter> | Si el archivo prueba.c existe en el directorio /user/ie84adh , despliega su contenido. Al llegar al final de la pantalla, espera que el usuario presione la barra de espacio para continuar. |

- Comando pipe (|)

| | |
|----------------------------|--|
| \$ ls -l /etc more <Enter> | Este directorio contiene una gran cantidad de archivos, por lo que no es posible verlos en una sola pantalla. Lo que hace el pipe (), es redireccionar la salida del comando ls para que sea procesada por el comando more , lo que permite ver el contenido del directorio deseado pantalla por pantalla. |
|----------------------------|--|

- Comando cp

| | |
|----------------------------|---|
| \$ cp .profile profile.bak | Copia el archivo .profile en otro archivo llamado profile.old dentro del mismo directorio. |
| \$ cp /tmp/profile ./ | Copia el archivo profile ubicado en el directorio /tmp al directorio actual. |
| \$ cp *.c ./programacion | Copia todos los archivos cuyo nombre finalice con .c dentro del subdirectorio programacion (suponiendo que dicho directorio exista).. |

- Comando mv

| | |
|--|---|
| \$ mv <i>profile.new</i> <i>.profile</i> | Renombra el archivo profile.new como .profile . |
| \$ mv <i>prueba.c</i> <i>/tmp</i> | Traslada en archivo prueba.c del directorio actual al directorio /tmp (no hace copia; cambia el archivo de lugar). |
| \$ mv <i>*.c</i> <i>./programacion</i> | Mueve todos los archivos cuyo nombre finalice con .c dentro del subdirectorio programacion (suponiendo que dicho directorio exista).. |

- Comando rm

| | |
|----------------------------|---|
| \$ rm <i>prueba.c</i> | Borra el archivo prueba.c |
| \$ rm <i>/tmp/prueba.c</i> | Borra el archivo prueba.c , ubicado en el directorio /tmp . |

- Comando passwd

| | |
|-----------|---|
| \$ passwd | Cambian su password actual por uno nuevo. Primero les pide el password actual. Después, el password nuevo y la confirmación del mismo. Es recomendable que utilicen dígitos y/o caracteres especiales, como \$, %, &, etc. Esto hace que su clave sea más segura. |
|-----------|---|

- Compilador gcc

| | |
|--|--|
| \$ gcc <i>prueba.c -o prueba</i> | Compila el archivo prueba.c . En el caso de que no existan errores, generará un archivo ejecutable llamado prueba . |
| \$ gcc <i>./programacion/prueba.c -o corre</i> | Compila el archivo prueba.c ubicado en el subdirectorio programación . Si no hay errores, generará un archivo ejecutable llamado corre en el directorio actual. |