

Nociones sobre Unix/Linux que todo Ingeniero debe conocer

Para empezar, un aspecto fundamental: los sistemas operativos basados en Unix (o Linux) distinguen entre mayúsculas y minúsculas. Sin embargo, las sentencias de ForTran se pueden escribir indistintamente en minúsculas o mayúsculas. Por otra parte, las extensiones de archivo (*.txt, *.dat, *.f, ...) en sistemas UNIX (o Linux) forman parte del nombre y no indican, necesariamente, el tipo de archivo (ASCII, binario, bmp,...). Asimismo, se recomienda que los nombres de archivos y carpetas no incorporen espacios en blanco, tildes, ñ, etc...

Generalidades

Entrar en la Cuenta de Usuario (o "Login")

```
Username: mnp_001
Password: la_que_sea
```

Salir de la Cuenta (o "exit")

```
$ exit
```

Cambiar el "Password" de la Cuenta

```
$ passwd
Old password:
New password:
Confirm new password:
```

Ayuda sobre comandos en Sistemas UNIX/LINUX

```
$ man comando
- Barra espaciadora --> Avanza página
- Tecla "Enter" --> Avanza línea
- Tecla "q" --> Sale de la ayuda
```

Función de autocompletado

Unix/Linux dispone de una función de autocompletado de nombres de archivos en la línea de comandos. Para escribir el nombre completo de un archivo existente bastará con escribir las primeras letras y pulsar el tabulador para autocompletar el nombre. Si se pulsa el tabulador dos veces consecutivas se listarán todas los nombres de archivos que comienzan por las letras que se han tecleado.

Sobre Directorios y carpetas

Directorio principal de cada usuario: /home/alumnos/*usuario*

Para ver la carpeta actual:

```
$ pwd
```

Para ver lo que hay en una carpeta o directorio:

```
$ ls          Lista simple de ficheros de la carpeta
$ ls -ls fichero Lista detallada de las propiedades de fichero
$ ls -lsa     Lista detallada de todos los archivos de la carpeta
```

Para Crear un Directorio

```
$ mkdir NombreCarpeta
```

Para moverse dentro del árbol de directorios o carpetas

```
$ cd NombreCarpeta Moverse a la carpeta NombreCarpeta desde la posición actual
$ cd ..             Regresar a un nivel anterior en el árbol de directorios
$ cd               Regresar a la carpeta raíz del usuario (/home/alu/usuario)
$ cd /RutaCompleta/ Moverse a la carpeta que se indica en RutaCompleta
```

Borrado de archivos

Precaución amigo conductor, los sistemas Unix/Linux NO INCORPORAN de forma natural herramientas de recuperación de operaciones anteriores (papelera de reciclaje, comando deshacer, ...) Si se borra un archivo, borrado está y será muy difícil de recuperar.

```
$ rm NombreArchivo Borra el archivo NombreArchivo
!! $ rm *           Borra todos los archivos existentes en la carpeta actual
$ rm *.txt         Borra todos los archivos cuyo nombre finaliza con .txt
$ rm Solucion?.txt Borra todos los archivos cuyo nombre coincide con el indicado
                    salvo el caracter ? que puede tomar cualquier valor.
```

Borrado de una carpeta

El borrado de una carpeta requiere:

- 1) Estar completamente seguro de que se quiere borrar la carpeta (y todo su contenido)
- 2) Tener en cuenta que el borrado de archivos y carpetas es un proceso IRREVERSIBLE

\$ rm -r *Carpeta* Borra la carpeta *Carpeta* y todo su contenido

Volcado en pantalla del contenido de un archivo de texto sin editarlo

\$ cat *Archivo*

Volcado en pantalla de las primeras/últimas líneas de un archivo de texto sin editarlo

\$ tail *Archivo* Vuelca las últimas 10 líneas de *Archivo*

\$ tail -99 *Archivo* Vuelca las últimas 99 líneas de *Archivo*

\$ head -999 *Archivo* Vuelca las primeras 999 líneas de *Archivo*

Copiar o duplicar el contenido de un archivo en otro archivo nuevo

\$ cp *ArchivoOriginal* *ArchivoNuevo*

\$ cp */RutaOriginal/ArchivoOriginal* */RutaNueva/ArchivoNuevo*

Ej. \$ cp ../hola.txt ./copia.txt --> Copia el archivo hola.txt del nivel anterior en el nivel actual con el nombre copia.txt

Ej. \$ cp -r ../Work1 ./Work2 --> Copia la carpeta Work1 y todo su contenido del nivel anterior en el nivel actual con el nombre work2

Mover un Archivo de una carpeta a otra/Cambio de Nombre de un Archivo

\$ mv *ArchivoOrigen* *ArchivoNuevo* --> Cambia el nombre del archivo

\$ mv */RutaOrigen/ArchivoOrigen* */RutaNueva/ArchivoNuevo* --> Cambia el nombre y la ubicación del archivo

Ej. \$ mv ../Hola.txt ./NuevoHola.txt --> Mueve el archivo Hola.txt del nivel anterior al nivel actual y lo llama NuevoHola.txt

Creación de un Archivo de Texto y/o Edición de uno ya existente

\$ nano *NombreFichero* (*En sistemas más antiguos este editor se llama pico*)

Cuando se invoca el editor de texto se puede escribir y desplazarse libremente en las distintas líneas del documento. Para preparar “programas fuente” de FORTRAN se recomienda emplear solamente códigos ASCII, y por ello no acentuar palabras ni emplear la letra “ñ”.

En la parte inferior de la ventana de edición se indican los comandos de ayuda a la edición más habituales.

Para guardar los cambios y abandonar el modo de edición de texto --> ‘Control+O’ y ‘Control+X’

Los sistemas Unix incorporan además de forma habitual editores de texto más avanzados pero también más complejos en cuanto a su utilización. Los editores más habituales, además del mencionado, son EMACS y VI.

Protecciones de los Archivos/Carpetas

\$ ls -ls *Archivo* --> Muestra información sobre *archivo*. Por ejemplo:

```
4 drwxr-xr-x 2 Einstein users 4096 nov 25 1915 Theory_of_Relativity
```

donde drwxr-xr-x hace referencia a las autorizaciones permitidas a 3 tipos genéricos de usuarios. La primera letra indica que es un directorio. A continuación y en grupos de 3 letras se muestran los permisos para el Usuario propietario, Grupo y Otros, respectivamente. Cada grupo de tres letras indica en orden de aparición: permisos de lectura (r), de escritura o modificación (w) y de ejecución (x).

Para cambiar los permisos asignados a un archivo

\$ chmod XYZ *Archivo*

Ej. \$ chmod 755 programa.out --> rwxr-xr-x

Cada uno de los códigos numéricos (X,Y,Z) se establecen de acuerdo con la siguiente tabla:

Octal	Texto	Binario	Permisos
0	---	000	Todo tipo de acceso denegado
1	--x	001	Sólo se permite la ejecución
2	-w-	010	Sólo se permite la escritura
3	-wx	011	Se permite la escritura y la ejecución
4	r--	100	Sólo se permite la lectura
5	r-x	101	Se permite la lectura y la ejecución
6	rw-	110	Se permite la lectura y la escritura
7	rwx	111	Todos los accesos permitidos

Compilación, Linkado y Ejecución de Programas FORTRAN
--

Edición del “programa fuente” en Lenguaje FORTRAN

```
$ nano trabajo.f
.....
  (Se escribe el programa)
.....
Para salir del editor:  Control+O          (Guardar cambios)
                       Control+X          (Salir del editor)
```

Sólo compilación del programa “trabajo.f”

```
$ gfortran trabajo.f -c
```

Se crea un archivo “objeto” que el sistema automáticamente asigna la terminación “.o”. Este fichero no se puede editar ni imprimir dado que se trata de un fichero binario encriptado.

Si no hay errores, entonces se puede pasar a la fase de “linkado” y creación del archivo ejecutable.

Linkado de ficheros objeto

```
$ gfortran trabajo.o -o trabajo.exe
```

Se crea un archivo ejecutable con el nombre trabajo.exe. Este archivo no se puede editar ni imprimir dado que se trata de un fichero binario encriptado. Si se omiten las sentencias “-o trabajo.exe” el compilador le asigna por defecto el nombre a.out al fichero ejecutable resultante.

Compilación y linkado del programa “trabajo.f”

```
$ gfortran trabajo.f -o trabajo.exe
```

Esta opción crea el fichero ejecutable en un sólo paso.

Ejecución de un programa o archivo ejecutable

```
$ ./trabajo.exe          --> Ejecuta el programa que se encuentra
                           en la propia carpeta
$ /RutaCompleta/trabajo.exe --> Ejecuta el programa ubicado en /RutaCompleta
```

Los caracteres “./” son necesarios para que el sistema operativo UNIX busque el archivo ejecutable en la propia carpeta en la que se encuentra para su ejecución. Si no se indica la ruta en la que se encuentra el fichero ejecutable el sistema operativo no los encontrará.

Para interrumpir la ejecución de un programa: teclear ‘‘Control+C’’**Para compilar y linkar un programa con subrutinas editadas en ficheros independientes**

```
$ gfortran trabajo.f -c
$ gfortran subrutina1.f -c
$ gfortran subrutina2.f -c
$ gfortran trabajo.o subrutina1.o subrutina2.o -o trabajo.exe
```

El resultado es un ejecutable llamado “trabajo.exe”.

Opciones de compilación más habituales

```
-Wall          --> Habilita que se muestren todas las advertencias del compilado
-fbounds-check --> Comprueba que no se exceden los límites de las dimensiones de
                   matrices, vectores, ...
-0X           --> Indica un nivel de optimización X del programa ejecutable resultante
-00          --> Sin optimización (posible ayuda para depurar errores de programación)
-01          --> Optimización de nivel 1
-02          --> Optimización de nivel 2 (más intensa, mayor velocidad de ejecución)
-03          --> Nivel de optimización más agresivo para mayor velocidad de ejecución.
```

Para invocar la aplicación de Ayuda de FORTRAN

```
$ man gfortran  --> Abre el archivo de ayuda con las opciones de compilación

- Barra espaciadora --> Avanza página a página
- Tecla ‘‘Enter’’   --> Avanza línea a línea
- Tecla ‘‘q’’       --> Sale de la ayuda
- Teclas ↑ y ↓     --> Se desplaza una línea arriba o abajo
```

EJEMPLO

Crearemos en primer lugar una carpeta llamada TRABAJO y nos moveremos a ella

```
$ mkdir TRABAJO
$ cd TRABAJO
$ nano factorial.f
```

Y en el editor escribimos teniendo en cuenta los criterios de escritura de sentencias FORTRAN en cada línea, es decir empezando a escribir por el carácter número 7 de la primera línea:

```
123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890
```

```

    program calculo_factorial
    implicit real*8(a-h,o-z),integer*4(i-n)
100 write(6,*)' Dime el numero '
    read(5,*)numero
    if (numero.gt.12)then
        write(6,*)' El numero no puede ser mayor de 12'
        goto 100
    endif
    nfactorial=1
    do i=1,numero
        nfactorial=nfactorial*i
    enddo
    write(6,99)numero,nfactorial
99 format(' El factorial de ',i5,' es ',i5)
end
```

Finalizamos la edición del archivo con las teclas: “Control+O” para guardar los cambios y “Control+X” para salir de la edición del archivo. Tecleamos el comando “ls” y observamos que se ha guardado el archivo correctamente.

A continuación ya jugaremos un poco compilando, linkando, ejecutando, ...

That's All Folks