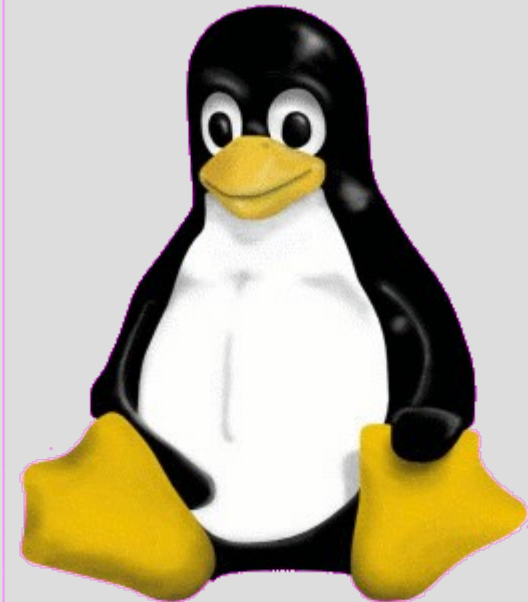


Introducción a Linux

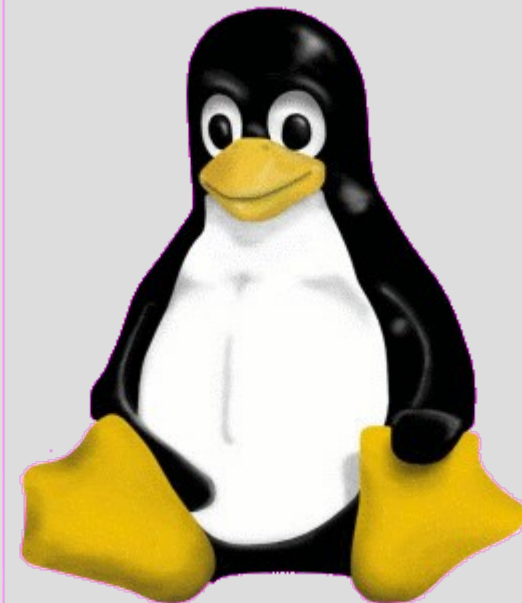
El regreso a los años 70: la consola (III)



Introducción a Linux

El regreso a los años 70: la consola (III)

- ✓ ¿Qué es y cómo acceder a una consola?
- ✓ Algunos comandos para moverse
- ✓ Sintaxis general de parámetros en BASH
- ✓ Encadenamiento de comandos
- ✓ Otros comandos GNU
- ✓ Editores de textos
- Permisos y usuarios
- Scripting en BASH



Introducción a Linux

El regreso a los años 70: la consola (III)

Permisos y usuarios

Para todos los ficheros se definen 3 permisos básicos:

lectura, escritura y ejecución (r , w , x)

Estos permisos se definen por usuario (propietario del fichero), grupo y otros (u,g,o)

Cada uno de estos permisos puede estar activo o inactivo (0 ó 1)



Introducción a Linux

El regreso a los años 70: la consola (III)

Permisos y usuarios

Se usan dos comandos que permiten cambiar permisos y propietarios de cada fichero:

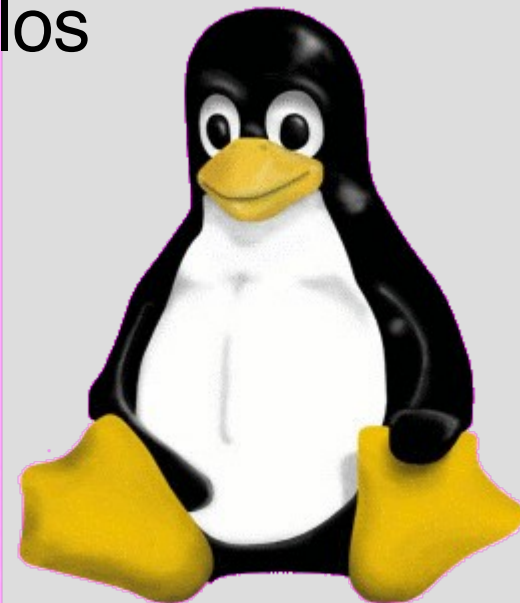
- **chmod** , *change mode*, permite cambiar los permisos de un fichero.

Ejemplos:

```
chmod +x fich
```

```
chmod 755 fich
```

```
chmog ug-x fich
```



Introducción a Linux

El regreso a los años 70: la consola (III)

Permisos y usuarios

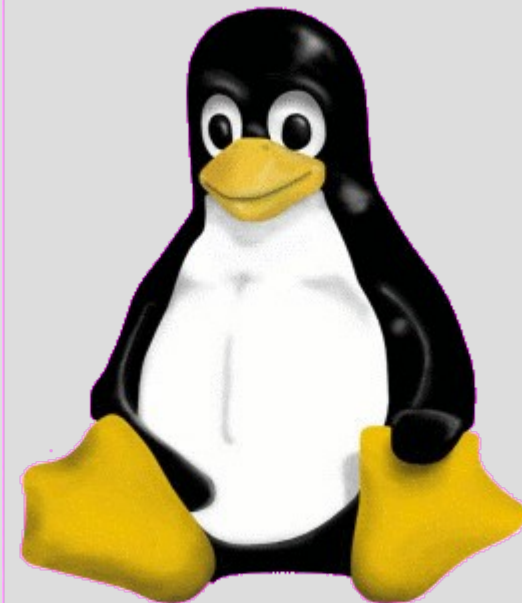
- **chown** , *change owner*, permite cambiar el usuario propietario de un fichero.

Ejemplos:

```
chown usuario2 fich
```

```
chown :grupo3 fich
```

```
chown usuario2:grupo3 fich
```



Introducción a Linux

El regreso a los años 70: la consola (III)

Permisos y usuarios

Podemos añadir y eliminar usuarios desde la línea de comandos, también cambiarle sus passwd, con 3 comandos:

- **useradd** (sólo para root), crea un usuario
- **userdel** (sólo para root), borra un usuario
- **usermod** (sólo para root), modifica un usuario, los grupos a los que pertenece,...
- **passwd** , sin parámetros, cambia la propia contraseña. Indicando un usuario, cambia la suya (sólo para root)



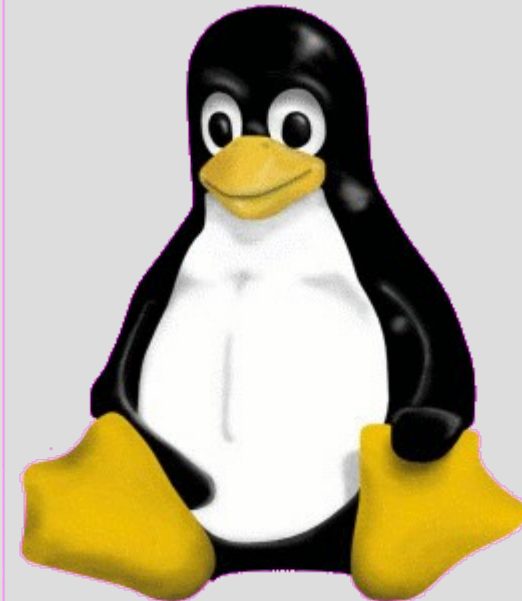
Introducción a Linux

El regreso a los años 70: la consola (III)

Permisos y usuarios

De manera similar podemos gestionar los grupos con estos dos comandos:

- `groupadd`
- `groupdel`



Introducción a Linux

El regreso a los años 70: la consola (III)

Permisos y usuarios

Al crear usuarios, se crea una carpeta de usuario (\$HOME) que contiene los ficheros por defecto que existen en /etc/skel

Para el BASH, se incluyen dos ficheros de configuración:

- ~/.bashrc , se ejecuta al iniciar BASH
- ~/.bash-profile , se ejecuta al iniciar la sesión un usuario, contiene el perfil de la sesión.



Introducción a Linux

El regreso a los años 70: la consola (III)

Scripting en BASH

Un script es un fichero de texto con permisos de ejecución que contiene una lista de comandos del intérprete que usemos.

La primera línea de un script debe indicar el intérprete que debe ejecutarlo:

```
#!/bin/bash
```



Introducción a Linux

El regreso a los años 70: la consola (III)

Scripting en BASH

Para ejecutar un script desde la línea de comandos lo llamaremos como si fuera un comando externo cualquiera.

Al escribir un comando en BASH, éste lo busca en los directorios indicados en \$PATH. Si no lo encuentra, nos mostrará un error.

Si queremos ejecutar un script, deberemos copiarlo en algún sitio de la ruta o indicarle la ruta completa de donde está.



Introducción a Linux

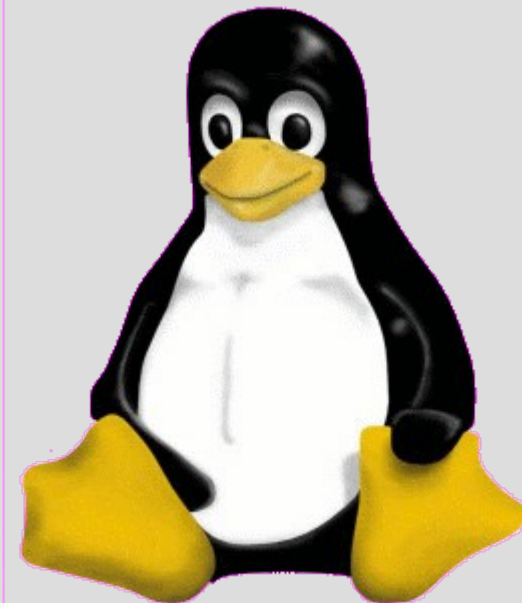
El regreso a los años 70: la consola (III)

Scripting en BASH

Primer ejemplo:

```
[usr@host ~]$ cat > script1
#!/bin/bash
#esto es un comentario

echo "Hola mundo"<CTRL>+<D>
[usr@host ~]$ chmod +x script1
[usr@host ~]$ ./script1
```



Introducción a Linux

El regreso a los años 70: la consola (III)

Scripting en BASH

Primer ejemplo:

```
[usr@host ~]$ cat > script1 (iniciamos un fichero nuevo)
```

```
#!/bin/bash
```

```
#esto es un comentario
```

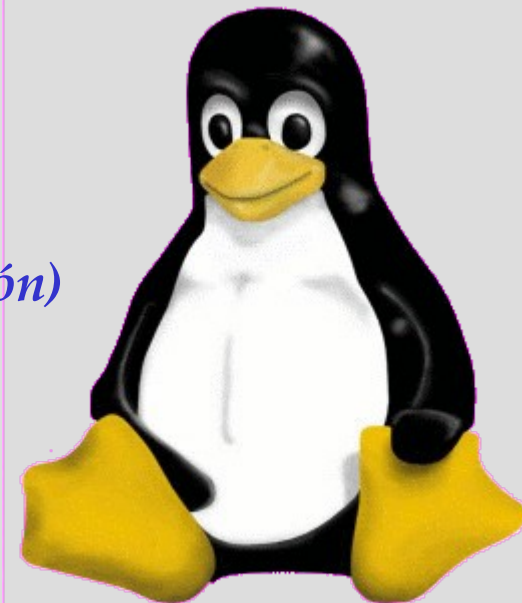
```
echo "Hola mundo" <CTRL>+<D> (fin de edición)
```

```
[usr@host ~]$ chmod +x script1
```

(hacemos ejecutable nuestro fichero)

```
[usr@host ~]$ ./script1
```

(ejecutamos nuestro fichero indicando su ruta relativa)



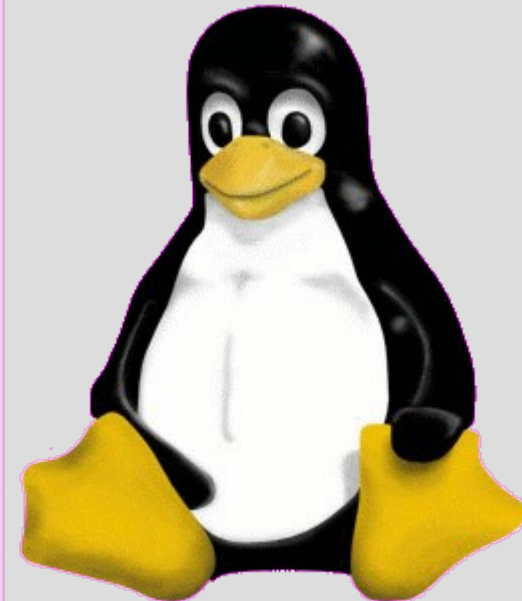
Introducción a Linux

El regreso a los años 70: la consola (III)

Scripting en BASH

Ejemplo2:

```
[usr@host ~]$ cat > script2
#!/bin/bash
for a in X Y Z;
do
    echo $a;
done;
[usr@host ~]$ chmod +x script2
[usr@host ~]$ ./script2
```



Introducción a Linux

El regreso a los años 70: la consola (III)

Scripting en BASH

Ejemplo2:

```
[usr@host ~]$ cat > script2
```

```
#!/bin/bash
```

```
for a in X Y Z; (X,Y,Z son valores)
```

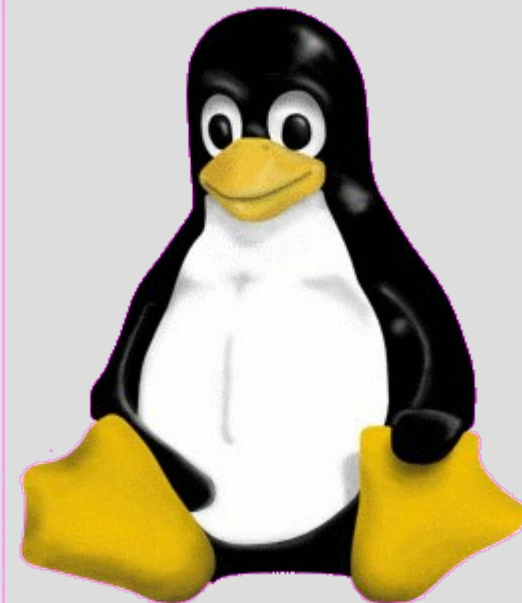
```
do (en cada iteración, se le asigna a la variable 'a' un valor)
```

```
    echo $a;
```

```
done;
```

```
[usr@host ~]$ chmod +x script2
```

```
[usr@host ~]$ ./script2
```



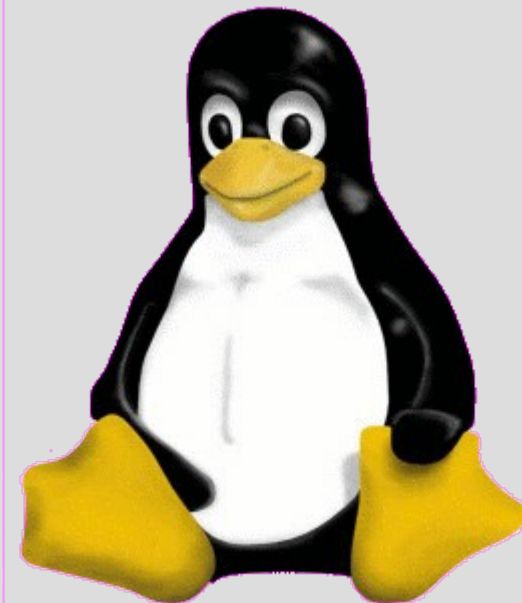
Introducción a Linux

El regreso a los años 70: la consola (III)

Scripting en BASH

Ejemplo3:

```
[usr@host ~]$ cat > script3
#!/bin/bash
A=1
while ( test $A -lt 5 );
do
    echo $A;
    A=$((A+1));
done;
[usr@host ~]$ chmod +x script3
[usr@host ~]$ ./script3
```



Introducción a Linux

El regreso a los años 70: la consola (III)

Scripting en BASH

Ejemplo3:

```
[usr@host ~]$ cat > script3
```

```
#!/bin/bash
```

```
A=1 (el punto y coma final no es imprescindible aquí)
```

```
while ( test $A -lt 5 );
```

```
do (el comando test devuelve true/false)
```

```
    echo $A;
```

```
    A=$((A+1)); (se evalua el contenido como una expresión)
```

```
done;
```

```
[usr@host ~]$ chmod +x script3
```

```
[usr@host ~]$ ./script3
```



Introducción a Linux

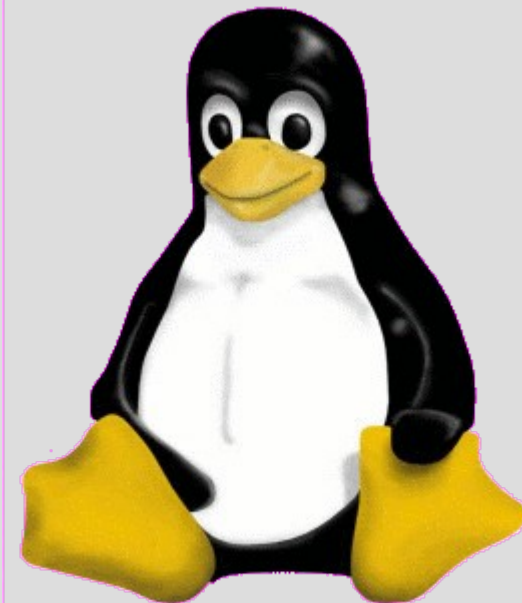
El regreso a los años 70: la consola (III)

Scripting en BASH

Ejemplo4:

```
[usr@host ~]$ cat > script4
#!/bin/bash

while [ -n "$1" ];
do
    echo $1;
    shift;
done;
[usr@host ~]$ chmod +x script4
[usr@host ~]$ ./script4
```



Introducción a Linux

El regreso a los años 70: la consola (III)

Scripting en BASH

Ejemplo4:

```
[usr@host ~]$ cat > script4
```

```
#!/bin/bash
```

(la variable \$1 corresponde al primer parámetro de la línea de comandos, \$0 es el propio comando)

```
while [ -n "$1" ]; (los corchetes equivalen a test)
```

```
do
```

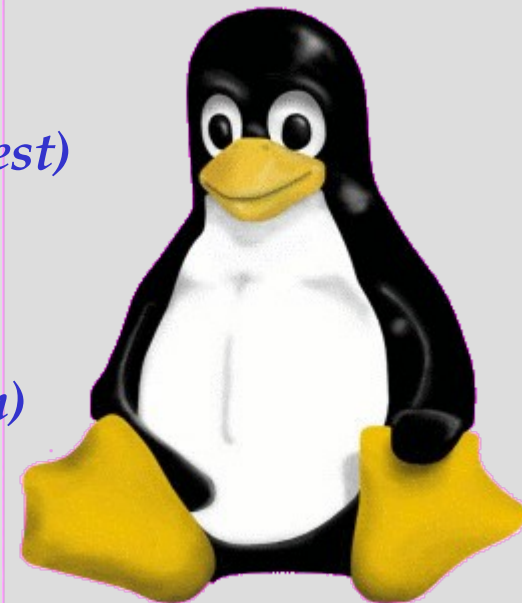
```
    echo $1;
```

```
    shift; (elimina el 1er parámetro, el resto se desplazan)
```

```
done;
```

```
[usr@host ~]$ chmod +x script4
```

```
[usr@host ~]$ ./script4
```



Introducción a Linux

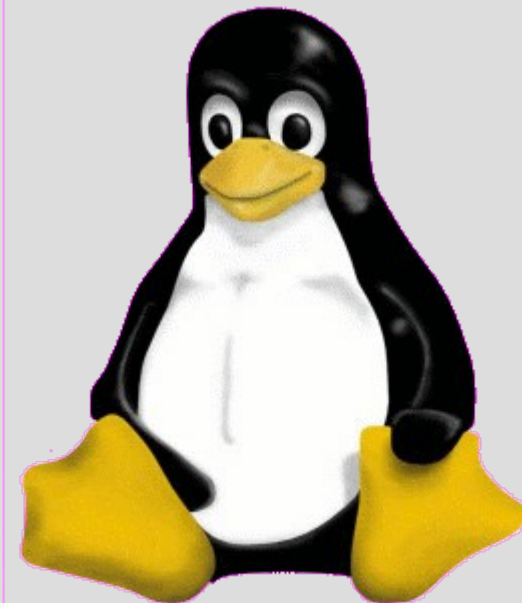
El regreso a los años 70: la consola (III)

Scripting en BASH

Ejemplo5:

```
[usr@host ~]$ cat > script5
#!/bin/bash

files=`ls -l`;
nf=`ls -l|wc`;
echo $files;
echo "Hay $nf ficheros";
[usr@host ~]$ chmod +x script5
[usr@host ~]$ ./script5
```



Introducción a Linux

El regreso a los años 70: la consola (III)

Scripting en BASH

Ejemplo5:

```
[usr@host ~]$ cat > script5  
#!/bin/bash
```

```
files=`ls -l`; (ejecutando el comando y captura la salida)
```

```
nf=`ls -l|wc`; (podemos combinar comandos)
```

```
echo $files;
```

```
echo "Hay $nf ficheros";
```

```
[usr@host ~]$ chmod +x script5
```

```
[usr@host ~]$ ./script5
```

