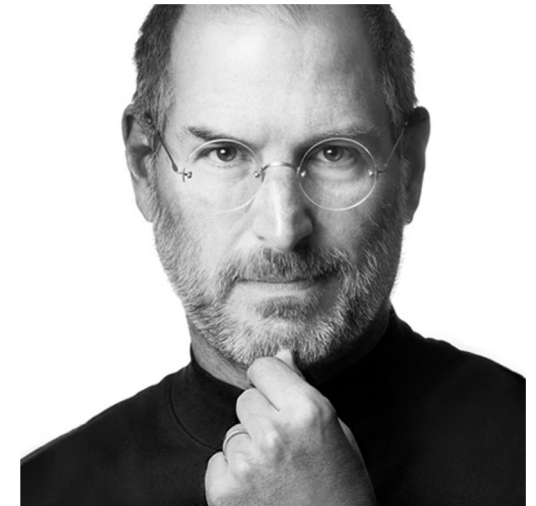
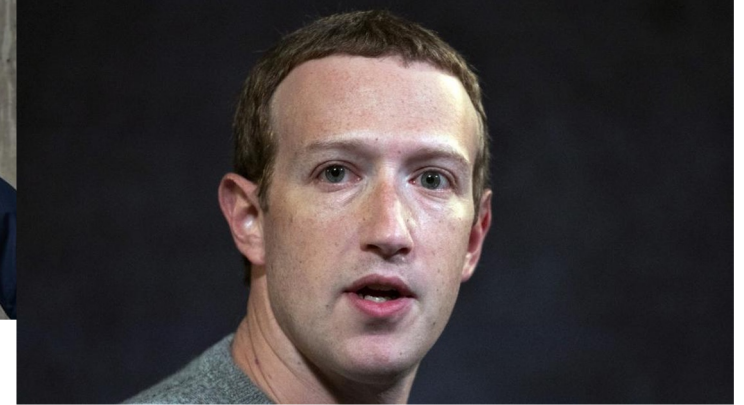


Unix / Linux

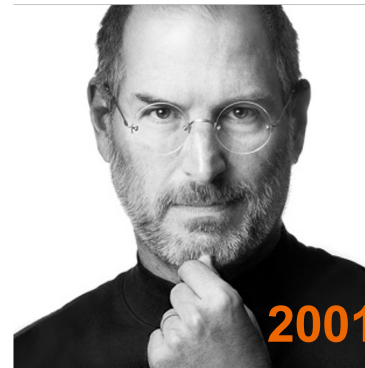
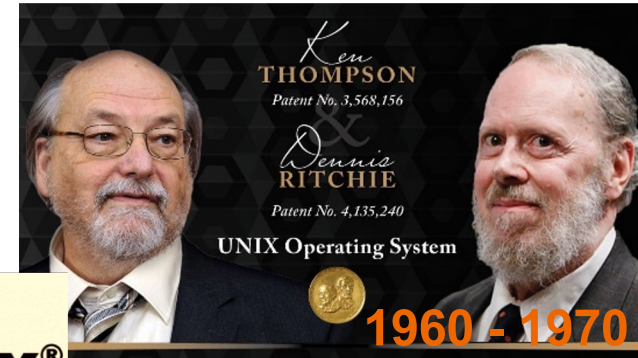
onlyfy

Who is Who?

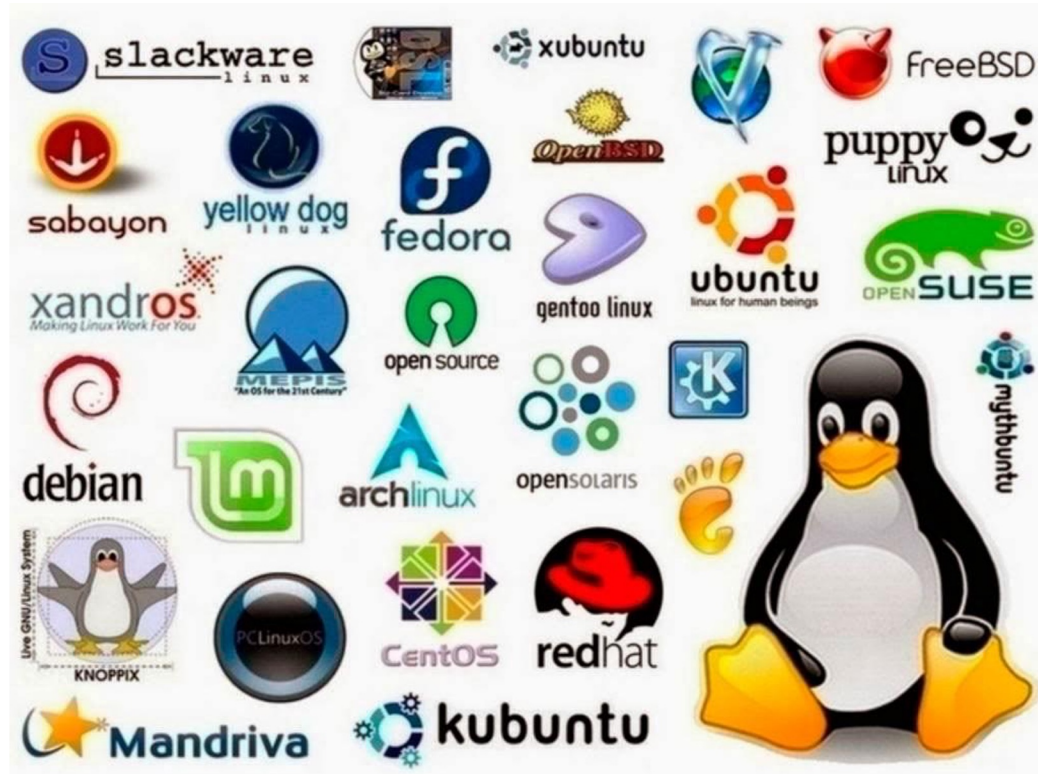


Step by step...

Tim Paterson



Distributions



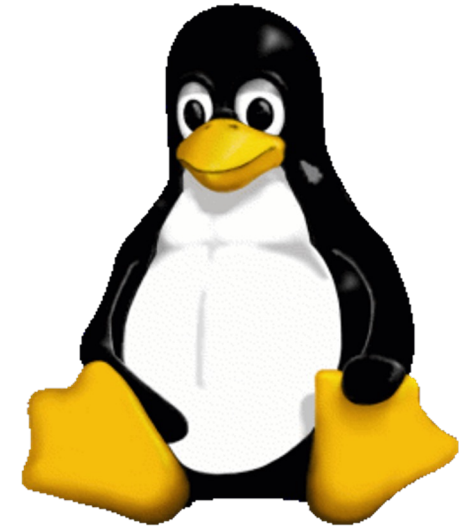
Qué es Unix?

Hoy en día los sistemas operativos Unix son ampliamente utilizados en multitud de dispositivos que abarcan desde los supercomputadores más capaces hasta los teléfonos móviles más populares, pasando por los ordenadores que utilizamos diariamente en nuestros escritorios. La filosofía de los sistemas Unix se caracteriza por:

- Un sistema de ficheros jerárquico,
- Una gran colección de pequeños programas que pueden trabajar en paralelo,
- El uso de ficheros de texto para almacenar los datos,
- Tratar los dispositivos como ficheros.

[Linux](#) y MacOS X son ejemplos de sistemas Unix.

[Unix](#) es un sistema operativo portable, multitarea y multiusuario desarrollado a partir de 1969.



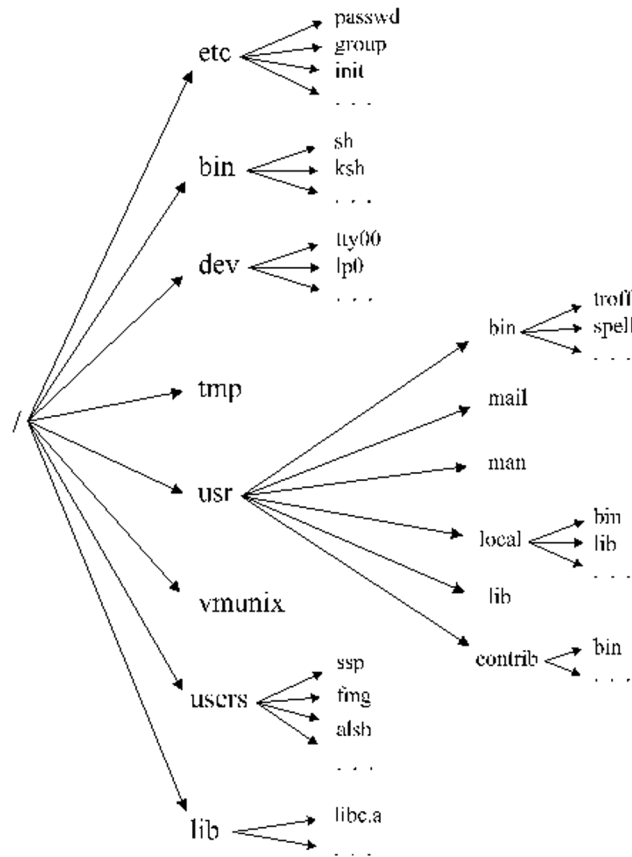
Tux

Pantalla negra



<https://bellard.org/jslinux/>

Sistema de Ficheros



• Directorio raíz (/)

Todos los archivos y directorios dependen de un único directorio denominado **directorio raíz** o *root*, el cual se representa por el símbolo */*. En el caso de que en el sistema tengamos varios dispositivos físicos de almacenamiento secundario (normalmente discos), todos deben depender del directorio raíz, y el usuario tratará cada uno de los discos como un subdirectorio que depende del directorio raíz (*root*).

• Directorio de conexión (~)

Se trata de un directorio que el administrador del sistema crea o asigna cuando da de alta a un usuario. El propósito de este directorio es suministrar un punto en la jerarquía de directorios del sistema de archivos UNIX a partir del cual el usuario puede almacenar y estructurar sus propios archivos y directorios. Este directorio suele ser tomado como **directorio de trabajo** inicial cuando el usuario establece una sesión de trabajo. Se representa mediante el símbolo *~*.

• Directorio de trabajo (.)

El **directorio de trabajo** o act el directorio en el que nos encontramos operando en cada momento de la sesión. Se representa mediante un punto *.*

Términos

Comando

Parámetro

Argumento

```
ls -l carpeta1
```

```
ls -l carpeta1
```

 Fácil?

```
ls -l carpeta1 -t -r
```

```
ls -l carpeta1 -t -r
```

 Fácil?

```
ls -ltr carpeta1
```

```
ls -ltr carpeta1
```

```
ls -ltr carpeta1 carpeta2
```

```
ls -ltr carpeta1 carpeta2
```


Sistema de Ficheros

Listado de acciones:

- Crear fichero: **touch fichero1.txt**
- Copiar Fichero: **cp origen destino**
- Mover Fichero: **mv origen destino**
- Borrar Fichero: **rm fichero1.txt**
- Listar Fichero: **ls fichero1.txt**
- Volcar contenido: **cat fichero1.txt**
- Cambiar Carpeta: **cd ruta / cd ..**

File Commands
ls - directory listing
ls -al - formatted listing with hidden files
cd <i>dir</i> - change directory to <i>dir</i>
cd - change to home
pwd - show current directory
mkdir <i>dir</i> - create a directory <i>dir</i>
rm <i>file</i> - delete <i>file</i>
rm -r <i>dir</i> - delete directory <i>dir</i>
rm -f <i>file</i> - force remove <i>file</i>
rm -rf <i>dir</i> - force remove directory <i>dir</i> *
cp <i>file1 file2</i> - copy <i>file1</i> to <i>file2</i>
cp -r <i>dir1 dir2</i> - copy <i>dir1</i> to <i>dir2</i> ; create <i>dir2</i> if it doesn't exist
mv <i>file1 file2</i> - rename or move <i>file1</i> to <i>file2</i> if <i>file2</i> is an existing directory, moves <i>file1</i> into directory <i>file2</i>
ln -s <i>file link</i> - create symbolic link <i>link</i> to <i>file</i>
touch <i>file</i> - create or update <i>file</i>
cat > <i>file</i> - places standard input into <i>file</i>
more <i>file</i> - output the contents of <i>file</i>
head <i>file</i> - output the first 10 lines of <i>file</i>
tail <i>file</i> - output the last 10 lines of <i>file</i>
tail -f <i>file</i> - output the contents of <i>file</i> as it grows, starting with the last 10 lines

Ejercicios

1. ¿Cuáles son los ficheros y directorios presentes en el directorio raíz?
2. ¿Cuáles son todos los archivos presentes en nuestro directorio de usuario?
3. Crea un directorio llamado experimento.
4. Crea con touch los archivos datos1.txt y datos2.txt dentro del directorio experimento.
5. Vuelve al directorio principal de tu usuario y desde allí lista los archivos presentes en el directorio experimento usando rutas absolutas y relativas
6. Borra todos los archivos que contengan un 2 en el directorio experimento.
7. Copia el directorio experimento a un nuevo directorio llamado exp_seguridad.
8. Borra el directorio experimento.
9. Renombra el directorio exp_seguridad a experimento.
10. Copia el fichero /etc/passwd al directorio experimento
11. Copia el fichero /etc/passwd al directorio experimento con el nombre usuarios.txt

Comandos Básicos

Listado de acciones:

- Mostrar contenido fichero: **cat fichero1.txt**
- Escribir pantalla: **echo "hola"**
- Escribir a Fichero: **echo "hola" > fichero1.txt**
- Buscar en Fichero: **grep a fichero1.txt**
- Histórico: **history**

Sistema

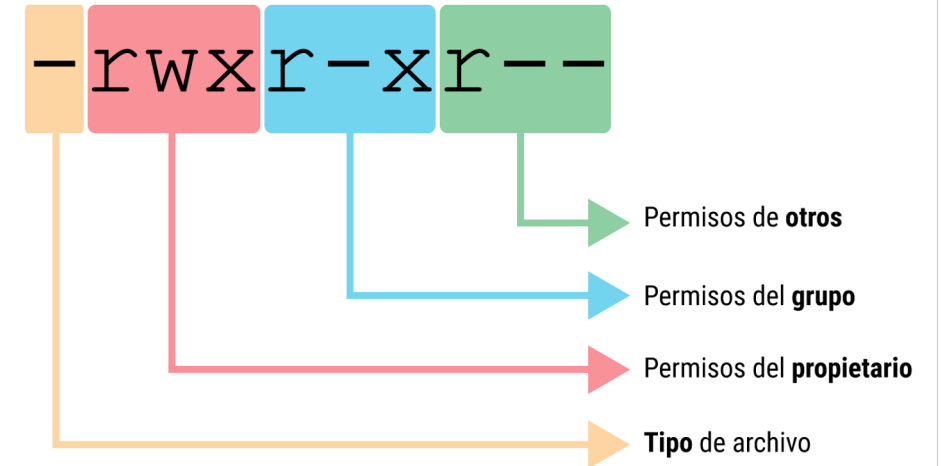
onlyfy

Permisos

Unix desde su origen ha sido un sistema multiusuario. Para conseguir que cada usuario pueda trabajar en sus archivos, pero que no pueda interferir accidental o deliberadamente con los archivos de otros usuarios se estableció desde el principio un sistema de permisos.

Por defecto un usuario tiene permiso para leer y modificar sus propios archivos y directorios, pero no los de los demás. En los sistemas Unix los ficheros pertenecen a un usuario concreto y existen unos permisos diferenciados para este usuario y para el resto. Además el usuario pertenece a un grupo de trabajo. Por ejemplo, imaginemos que la usuaria alicia puede pertenecer al grupo de trabajo “diagnostico”. Si alicia crea un fichero este tendrá unos permisos diferentes para alicia, para el resto de miembros de su grupo y para el resto de usuarios del ordenador. Podemos ver los permisos asociados a los ficheros utilizando el comando `ls` con la opción `-l` (Long)

<https://chmod-calculator.com/>



Número	Binario	Lectura (r)	Escritura (w)	Ejecución (x)
0	000	✗	✗	✗
1	001	✗	✗	✓
2	010	✗	✓	✗
3	011	✗	✓	✓
4	100	✓	✗	✗
5	101	✓	✗	✓
6	110	✓	✓	✗
7	111	✓	✓	✓

Unidades de Disco

El sistema de archivos puede abarcar una o más [particiones](#). Una partición es una región de un disco o de cualquier otro medio de almacenamiento. Las instalaciones de Windows tienen normalmente una partición por disco, pero en Linux esto no es tan habitual. Cada partición tiene un sistema de archivos propio, pero en Unix estos sistemas deben estar montados en algún lugar dentro de la jerarquía que cuelga de la raíz. En Windows cada partición tiene por defecto una jerarquía independiente.

```
himanshu@ansh:~/htf-daily$ df
Filesystem      1K-blocks    Used Available Use% Mounted on
udev            1935948         0   1935948  0% /dev
tmpfs           391428         6184   385244  2% /run
/dev/sda7       201902024 6971940 184651020  4% /
tmpfs           1957124         1512   1955612  1% /dev/shm
tmpfs           5120           4        5116  1% /run/lock
tmpfs           1957124          0   1957124  0% /sys/fs/cgroup
/dev/loop1       81408        81408          0 100% /snap/core/2312
/dev/loop2       12928        12928          0 100% /snap/wormhole/14
/dev/loop0       81408        81408          0 100% /snap/core/1689
tmpfs           391428          64   391364  1% /run/user/1000
```

Vamos a crear una máquina



Pasos:

```
docker create -it --name unix ubuntu:latest
```

```
Unable to find image 'ubuntu:latest' locally
latest: Pulling from library/ubuntu
c391327a0f1d: Pull complete
Digest: sha256:9b8dec3bf938bc80fbc758d856e96fdfab5f56c39d44b0cff351e847bb1b01ea
Status: Downloaded newer image for ubuntu:latest
4516327a6a049c20d2cdd4c2ac628ec09390231f9d3f76b14c1937eeb9e3a18e
```

```
docker start unix
```

```
docker exec -it unix /bin/bash
```

```
➤ onlyfy-infrastructure git:(CAPE-364-RDS-Audit-Logs) docker exec -it unix /bin/bash
root@4516327a6a04:/#
```

Instalar apps

¿Qué estamos haciendo aquí?

```
$ sudo apt update  
$ sudo apt install openjdk-8-jdk
```

<http://ports.ubuntu.com/ubuntu-ports/dists/jammy/>

¿Cómo conseguimos que funcione el siguiente commando?

```
root@4516327a6a04:/# git --version  
git version 2.34.1
```



Titel (Eingabe über "Einfügen > Kopf- und Fußzeile")

Bash Variables

Variables de Entorno

→ Mostrar variables:

`env` -> muestra todas las variables

`echo $A` -> muestra el valor de una variable

→ Establecer variables:

`export A=1`

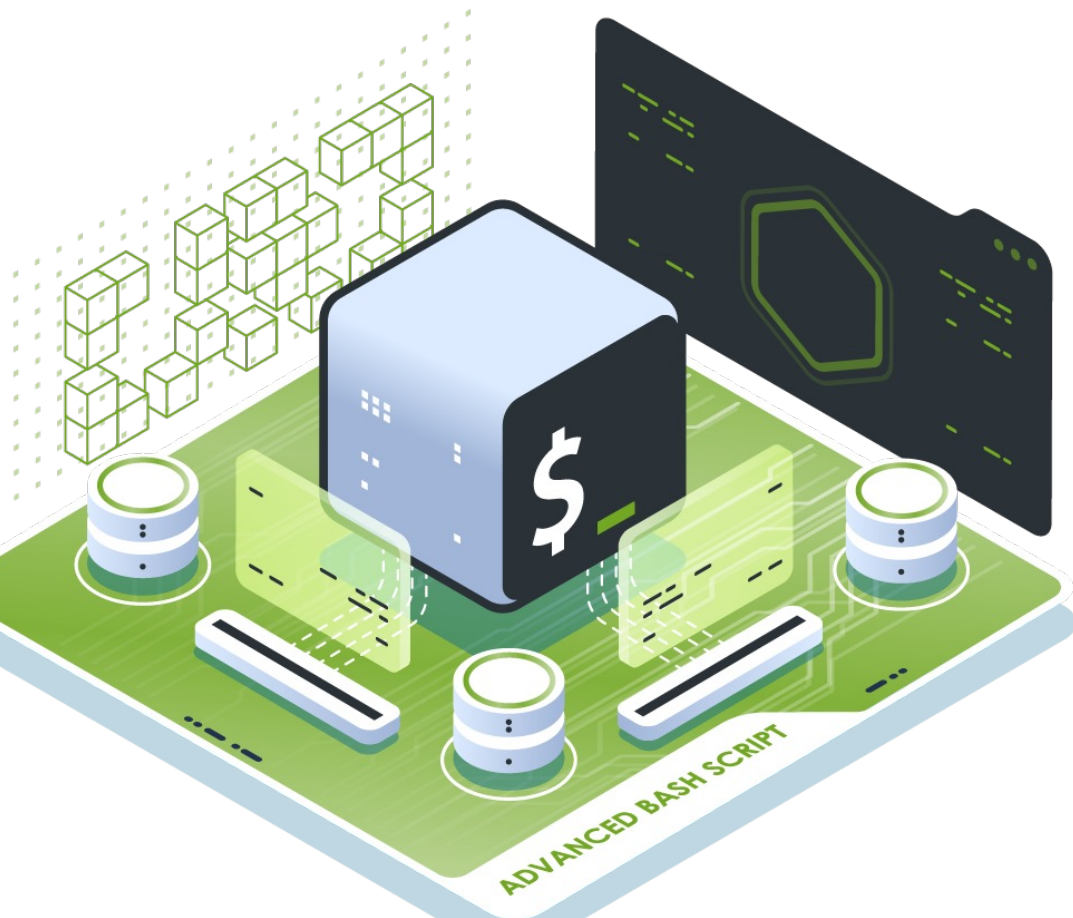
→ Establecer variables:

`.bashrc`

Ejercicios

1. ¿Cuáles son los permisos de los directorios presentes en el directorio raíz y en nuestro directorio de usuario? ¿A quién pertenecen los ficheros y qué permisos tienen los distintos usuarios del ordenador?
2. Crea un directorio en tu home y muestra los permisos que tiene.
3. Cambia los permisos para que sólo tu usuario pueda acceder al nuevo directorio
4. Crea un fichero nuevo y dale permisos de ejecución para todos los usuarios
5. Último fichero modificado en el directorio /etc.
6. Lista los ficheros de /etc con su tamaño y ordénalos por tamaño.
7. Copia todos los ficheros y directorios del directorio /etc cuyo nombre comience por s. ¿Has podido copiarlos todos?
8. ¿Cuánto espacio libre queda en las distintas particiones del sistema?
9. ¿Cuánto espacio ocupan todos los ficheros y subdirectorios de tu \$HOME?

Scripting



Bash es un intérprete de órdenes que generalmente se ejecuta en una ventana de texto donde el usuario escribe órdenes en modo texto. Bash también puede leer y ejecutar órdenes desde un archivo, llamado guion o 'script'. Al igual que todos los intérpretes de Unix, es compatible con el agrupamiento de nombres de archivo (coincidencia de comodines), tuberías, sustitución de comandos, variables y estructuras de control para pruebas de condición e iteración. Las palabras reservadas, la sintaxis, las variables de ámbito dinámico y otras características básicas del lenguaje se copian de sh. Otras características, por ejemplo, el historial, se copian de csh y ksh. Bash es un intérprete de órdenes compatible con POSIX, pero con varias extensiones.

1. Instalar un editor de texto.



2. Crear un fichero llamado hola.sh

3. Escribir dentro este contenido

```
#!/bin/bash
echo "hello world"
```

4. Seguid vosotros... 😊

Thank you

onlyfy