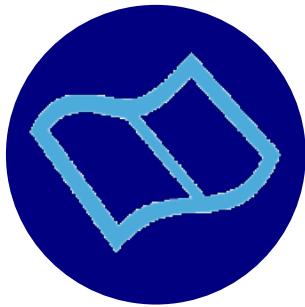


# **Repaso de LINUX**



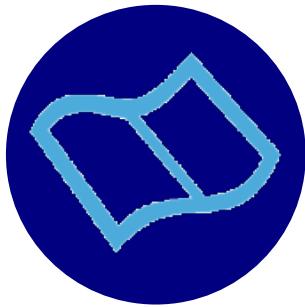
# INTRODUCCIÓN

- Una gran parte de las aplicaciones actuales se despliegan sobre sistemas basados en Linux
- Antes de poder hacer despliegue sobre estos sistemas es necesario saber moverse y manejar Linux correctamente
- En el módulo de Despliegue de Aplicaciones Web nos basaremos en distribuciones de Linux, tales como Debian y sus variantes



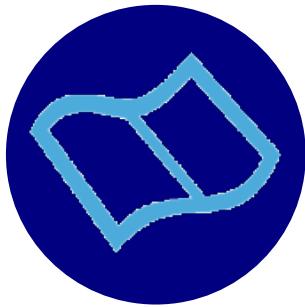
# ARQUITECTURA LINUX

- Linux maneja muy bien el hardware
  - Mejor compatibilidad que Windows y MAC
  - Se basa en la Hardware Abstraction Layer – HAL
    - HAL se compone de : sysfs, udev, hald, dbus
- HAL dispone de un demonio, **hald**
  - Mantiene actualizada la base de datos del hardware instalado
- HAL se comunica mediante **dbus**
  - Comunicación interproceso (IPC)
  - Puede establecer automáticamente puntos de montaje
- HAL usa un gestor de dispositivos a nivel de núcleo, **udev**
  - Consulta conexiones por si se producen eventos, añade dispositivos...



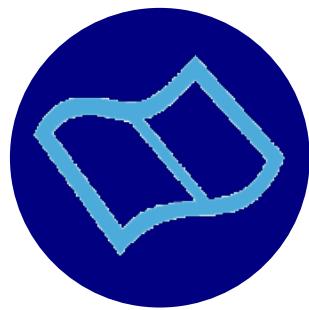
# ARQUITECTURA LINUX

- HAL hace uso de un sistema de archivos virtual, **sysfs**
  - **/sys** incluyendo sus subdirectorios
    - **block/** representa a los dispositivos de bloques
    - **class/** representa las clases de dispositivos
    - **devices/** contiene la jerarquía de dispositivos detectados
    - **firmware/** para los drivers
    - **module/** para los módulos de núcleo cargados



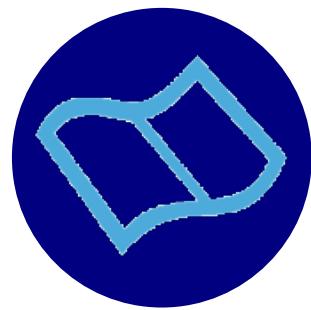
# ARQUITECTURA LINUX

- Configuraciones dinámicas en **/proc**
  - Otro sistema de archivos virtual
  - Información sobre la CPU : **/proc/cpuinfo**
  - Encaminamiento de paquetes (Linux como router) :  
**/proc/sys/net/ipv4/ip\_forward**
  - Otros ejemplos : **/proc/modules**, **/proc/mounts/**,  
**/proc/meminfo**

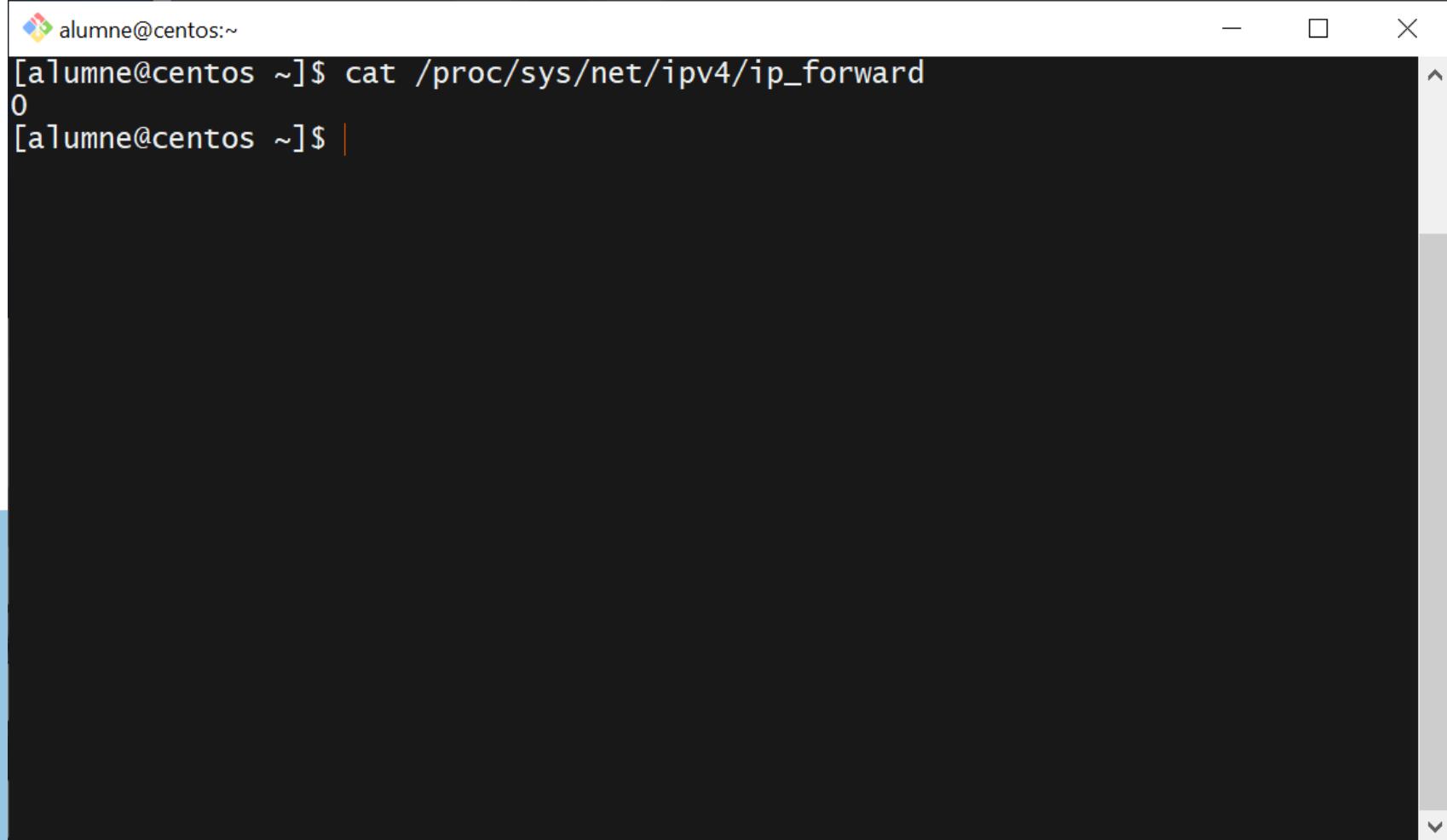


# ARQUITECTURA LINUX

```
alumne@centos:~$ cat /proc/cpuinfo
processor       : 0
vendor_id      : GenuineIntel
cpu family    : 6
model          : 158
model name    : Intel(R) Core(TM) i7-7700HQ CPU @ 2.80GHz
stepping       : 9
microcode     : 0x8e
cpu MHz        : 2808.005
cache size    : 6144 KB
physical id   : 0
siblings       : 1
core id        : 0
cpu cores     : 1
apicid         : 0
initial apicid: 0
fpu            : yes
fpu_exception  : yes
cpuid level   : 22
wp             : yes
flags          : fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov
pat pse36 clflush mmx fxsr sse sse2 ss syscall nx pdpe1gb rdtscp lm constant_tsc
arch_perfmon nopl xtopology tsc_reliable nonstop_tsc eagerfpu pni pclmulqdq sse
e3 fma cx16 pcid sse4_1 sse4_2 x2apic movbe popcnt tsc_deadline_timer aes xsave
```

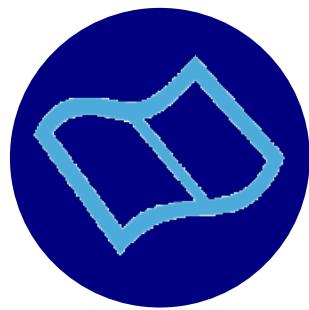


# ARQUITECTURA LINUX



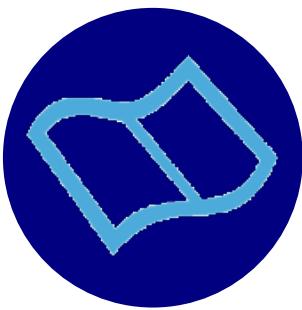
A screenshot of a terminal window titled "alumne@centos:~". The window shows the command `cat /proc/sys/net/ipv4/ip_forward` being run, with the output "0" displayed. The terminal has a dark theme with light-colored text. The window has standard operating system window controls (minimize, maximize, close) in the top right corner.

```
[alumne@centos ~]$ cat /proc/sys/net/ipv4/ip_forward
0
[alumne@centos ~]$ |
```



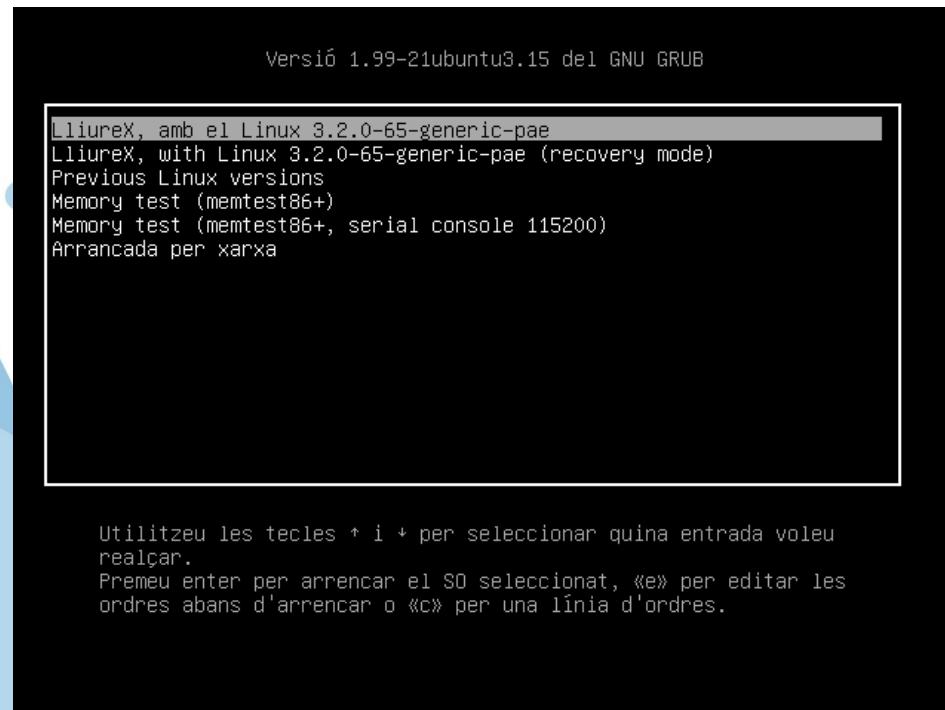
# ARQUITECTURA LINUX

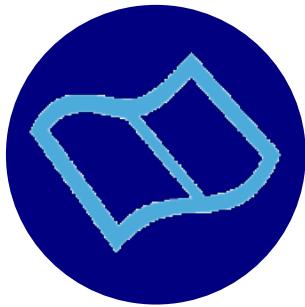
```
alumne@centos:~$ ls /proc
1      1950  3041  4966  6234  98      ioports      partitions
10     1959  3042  4971  6259  acpi    irq          sched_debug
11     2      3043  4976  6271  asound   kallsyms   schedstat
13     20     3044  4977  6615  buddyinfo kcore        scsi
13387  21     3045  4983  6941  bus      keys         self
14     22     3046  4989  7       cgroups   key-users   slabinfo
15     23     3047  4993  703    cmdline   kmsg         softirqs
16     24     31     4997  7159  consoles  kpagecount  stat
16926  25     3114  5       7160    cpuinfo   kpageflags  swaps
16930  2988  3131  5008   7162    crypto    loadavg    sys
16931  2989  3143  5009   718     devices   locks      sysrq-trigger
16961  3      32     5010   724    diskstats mdstat      sysvipc
17     30     33     5012   730    dma       meminfo   timer_list
1724   3003  4      5701   737    driver    misc      timer_stats
1728   3004  41     6      7525  execdomains modules   tty
1729   3022  42     60     7539  fb       mounts   uptime
1776   3027  43     6134   7540  filesystems mpt      version
1782   3032  44     6151   8      fs       mtrr      vmallocinfo
18     3035  45     6177   9      interrupts net      vmstat
19     3038  47     6207   91    iomem    pagetypeinfo zoneinfo
alumne@centos ~]$
```



# ARQUITECTURA LINUX

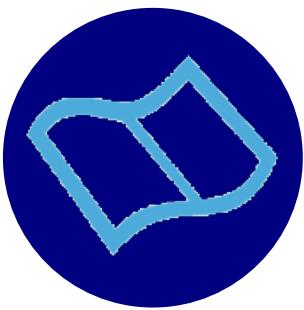
- Cuando Linux Arranca suele :
  - ... presentar un cargador del sistema con un menú (GRUB)
  - ... mostrar una secuencia de mensajes por pantalla que se suceden muy rápidamente (en modo recovery)





# ARQUITECTURA LINUX

- Cuando Linux Arranca suele :
  - Los eventos de arranque los podemos visualizar con el comando **dmesg**
  - **Algunos tipos de mensajes :**
    - Kernel panic, hardware no detectado, corrupción del sistema
    - Hay muchísimos eventos registrados
      - Para encontrar algún evento concreto es conveniente utilizar **dmesg** junto con algún filtro, como **grep** (se estudiará más adelante)
      - Ejemplo: **dmesg | grep IPv6**



# ARQUITECTURA LINUX

- **Runlevels**

- Indican en qué modo arrancará Linux
- Están numeradas del 0 al 6
  - Algunas distribuciones usan letras en lugar de números
- En las nuevas distribuciones ya no se suele usar el "demonio" **init**
  - Pero suelen mantener el sistema de runlevels
- Cuando el sistema entra en uno de los runlevels, ejecuta :
  - ... al arrancar los scripts (enlaces) que empiezan por **S**...
  - ... al cerrar el sistema, los scripts (enlaces) que empiezan por **K**...
  - ... del directorio **/etc/rcX.d**, siendo **X** el runlevel
- Podemos forzar un runlevel ejecutando **init X**, siendo **X** el runlevel



# ARQUITECTURA LINUX

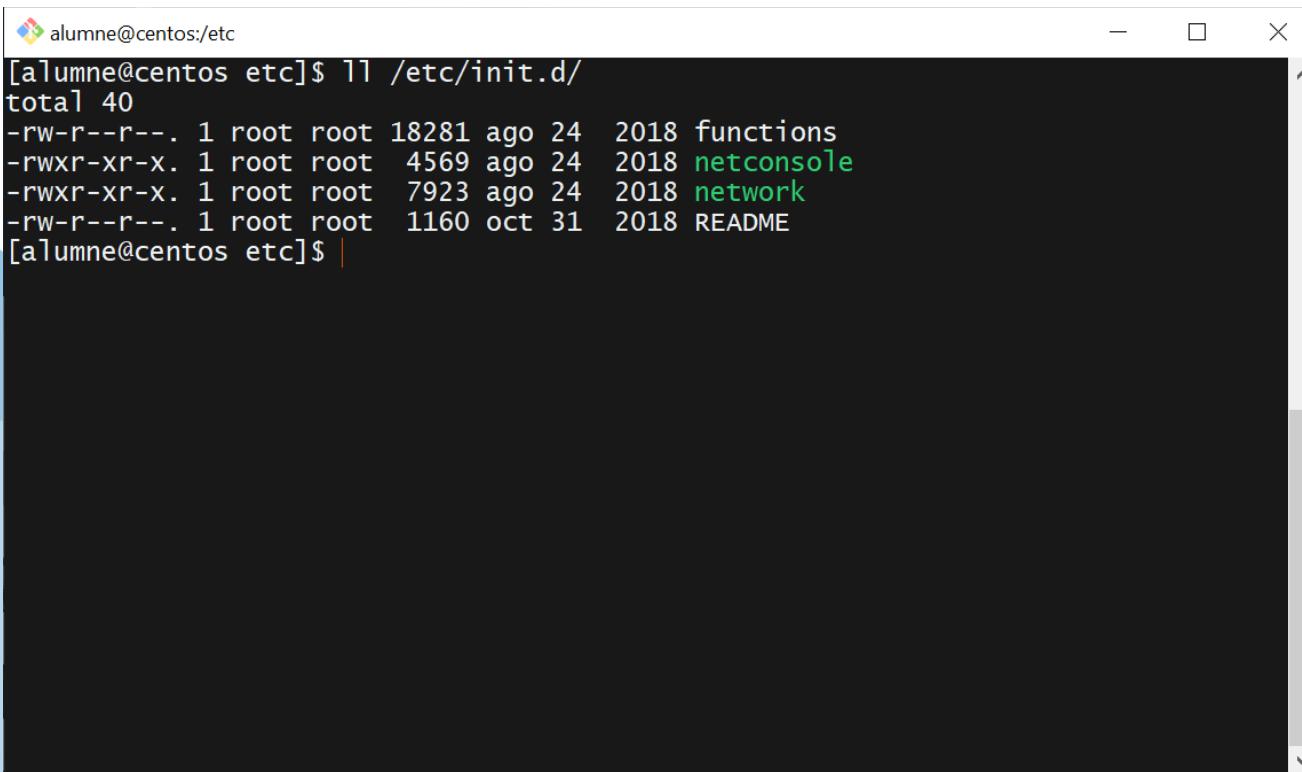
- **Runlevels (2)**

- En algunas distribuciones no se usan todos los runlevels
- En Debian/Ubuntu no están disponibles el 3, el 4 y el 5
- Runlevel 6 = reiniciar el sistema
  - **init 6 = reboot, shutdown -r now**
- Run 0 = apagar
  - **init 0 = halt, shutdown -h now**

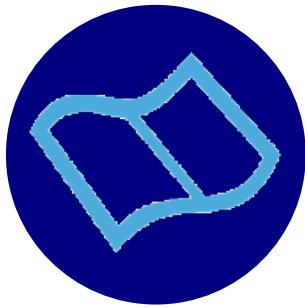


# ARQUITECTURA LINUX

- Scripts de inicio y fin
  - Se encuentran ubicados en **/etc/init.d**
  - Se crean enlaces a ellos dentro de los directorios **/etc/rcX.d**, para los diferentes runlevel

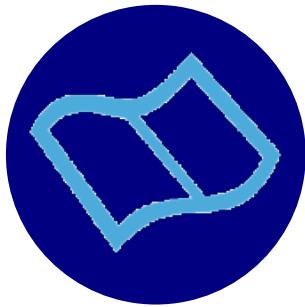


```
[alumne@centos etc]$ ll /etc/init.d/
total 40
-rw-r--r--. 1 root root 18281 ago 24 2018 functions
-rwxr-xr-x. 1 root root 4569 ago 24 2018 netconsole
-rwxr-xr-x. 1 root root 7923 ago 24 2018 network
-rw-r--r--. 1 root root 1160 oct 31 2018 README
[alumne@centos etc]$ |
```



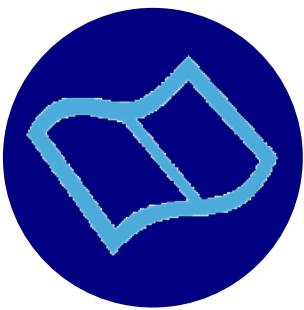
# ARQUITECTURA LINUX

- Scripts de inicio y fin (2)
  - Los scripts de **/etc/init.d** los podemos lanzar a mano con distintas opciones:
    - **start**: arranca el servicio
    - **stop**: para el servicio
    - **restart**: reinicia el servicio
    - **reload**: lee de nuevo el fichero de configuración del servicio, sin pararlo
      - A veces hace falta utilizar **force-reload**
    - **status**: muestra el estado del servicio



# ARQUITECTURA LINUX

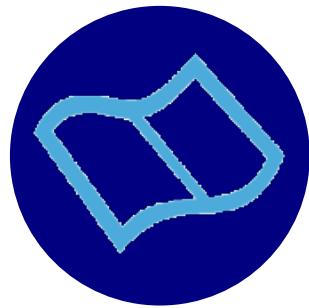
- Scripts de inicio y fin (3)
  - En las nuevas distribuciones se suele emplear **service** para iniciar o parar los servicios
    - Ejemplo : **/etc/init.d/ssh start = service ssh start**



# ARQUITECTURA LINUX

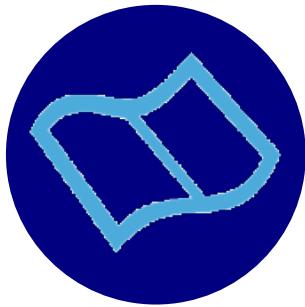
- **Targets**

- En las nuevas distribuciones Linux, el sistema de **runlevels** antiguo se sustituye por el sistema de **targets** de **Systemd**
- Existe una retrocompatibilidad con el sistema antiguo (equivalencias)
- Para ver la equivalencia entre **targets** y **runlevels** se puede ejecutar el comando:  
`ls -l /usr/lib/systemd/system/runlevel?.target`
- **Systemd** incluye una mayor granularidad



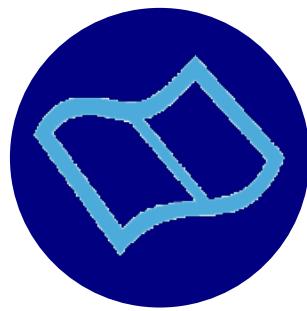
# ARQUITECTURA LINUX

- **Targets (II)**
  - Para ver todos los target disponibles, ejecutar el comando  
**systemctl list-units --type=target --all**



# ARQUITECTURA LINUX

- En las versiones más modernas de Linux la tendencia es la de sustituir los scripts de inicio por los servicios registrados en **systemctl**
- Mediante el comando **systemctl list-unit-files** podemos saber el estado de los servicios registrados junto con sus correspondientes ficheros
- A continuación se muestra uno de los mensajes de Linux al respecto de esta cuestión
- También se muestra parte de la salida de **systemctl list-unit-files**



# ARQUITECTURA LINUX

```
alumne@centos:~/etc
[alumne@centos etc]$ systemctl list-unit-files
UNIT FILE                                     STATE
proc-sys-fs-binfmt_misc.automount           static
dev-hugepages.mount                         static
dev-mqueue.mount                            static
proc-sys-fs-binfmt_misc.mount               static
sys-fs-fuse-connections.mount              static
sys-kernel-config.mount                     static
sys-kernel-debug.mount                      static
tmp.mount                                    disabled
brandbot.path                               disabled
systemd-ask-password-console.path          static
systemd-ask-password-plymouth.path          static
systemd-ask-password-wall.path              static
session-1.scope                            static
session-2.scope                            static
session-3.scope                            static
auditt.service                             enabled
autovt@.service                           enabled
blk-availability.service                  disabled
brandbot.service                          static
chrony-dnssrv@.service                   static
chrony-wait.service                      disabled
chronyd.service                           enabled
console-getty.service                    disabled
console-shell.service                   disabled
container-getty@.service                 static
cpupower.service                          disabled
crond.service                            enabled
```

# CREACIÓN DE USUARIOS

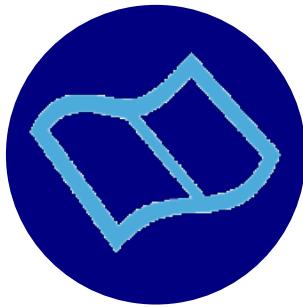
## AÑADIR UN USUARIO A LOS “SUDDERS”



# ACTIVIDAD

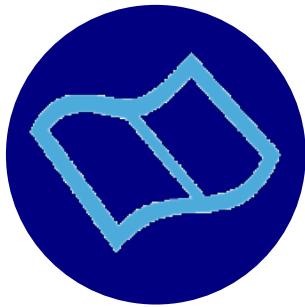
En la máquina virtual DEBIAN que tenéis instalada:

- Instalar el paquete **sudo**
- Hacer que el usuario **administrador** que se creó durante la instalación, pertenezca al grupo de **sudoers**
- Crear un nuevo usuario con vuestro nombre y hacer que éste también sea **sudoer**



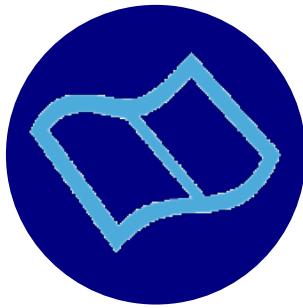
# SISTEMA DE FICHEROS

- Sistema de ficheros, volúmenes y directorios
  - Sistema de ficheros local:
    - Parcición formateada, volumen, RAID
  - Un sistema de ficheros se monta siempre sobre un directorio
  - La raíz del sistema de ficheros de Linux se representa por /
  - Un comando que usaremos habitualmente para examinar el sistema de ficheros: **ls**



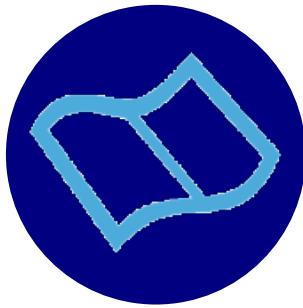
# SISTEMA DE FICHEROS

- Algunos directorios / puntos de montaje
  - **/boot** – contiene el kernel y el GRUB
    - Recomendable montar una partición primaria
  - **/home** – punto de montaje para las carpetas de los usuarios
  - **/var** – usado por varios servicios, por ejemplo algunos servidores web o ftp



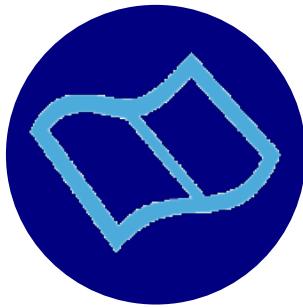
# SISTEMA DE FICHEROS

- Partición especial: **SWAP**
  - Habitualmente usaremos una partición del doble del tamaño de la RAM
  - Podemos decidir hacerla de menor tamaño, sobre todo en sistemas con gran cantidad de RAM



# SISTEMA DE PAQUETES DE DEBIAN / UBUNTU

- Debian / Ubuntu utilizan las herramientas **dpkg** y **apt** para las operaciones de instalación y desinstalación de paquetes
  - **dpkg** = Debian Package Management Tool
  - Los paquetes de software tienen la extensión **.deb**
- Uso básico de **dpkg**
  - **dpkg -l** muestra la lista de paquetes instalados
  - **dpkg -l nombrepaquete** muestra los detalles de instalación
  - **dpkg -i nombrepaquete**, instala el paquete indicado
  - **dpkg -r nombrepaquete**, elimina el paquete, manteniendo sus ficheros de configuración
  - **dpkg -P nombrepaquete**, elimina el paquete, eliminando sus ficheros de configuración



# SISTEMA DE PAQUETES DE DEBIAN / UBUNTU

- Uso de **dpkg** (2)
  - **dpkg -i nombrepaquete** puede que no funcione si hay dependencias con otros paquetes
  - Podemos resolver dependencias con las utilidades **apt**
- **dpkg** puede requerir la intervención del usuario para ciertos detalles de la configuración (ej. solicitar contraseñas, IPs, ...)
  - **dpkg-reconfigure nombrepaquete**, vuelve a lanzar prompts para reconfigurar un paquete ya instalado

# dpkg

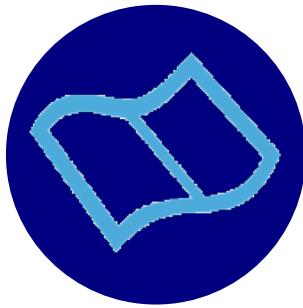
GESTIÓN MANUAL  
DE PAQUETES



.deb

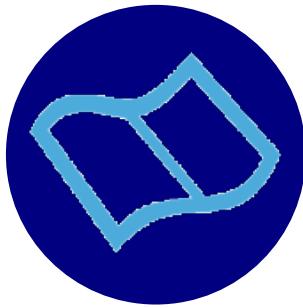
# ACTIVIDAD 1.01

- Realizar la instalación del SGBD **Firebird3** en DEBIAN, y su cliente en línea de comandos (isql)
- Para realizar esta instalación, emplear **únicamente** la utilidad **dpkg**
- La gestión de las **dependencias** debe realizarse **manualmente**
- Esta actividad debe resolverse **realizando un vídeo** en el que el/la alumno/a describa el proceso realizado



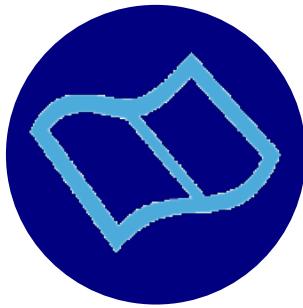
# SISTEMA DE PAQUETES DE DEBIAN / UBUNTU

- Repositorios y comandos **apt**
  - **apt** gestiona las dependencias entre paquetes
  - Los repositorios son los lugares en los que se buscarán los paquetes
    - Pueden ser locales o remotos
  - Comandos **apt**
    - **apt install nombrepaquete**, instala un paquete y todas sus dependencias
    - **apt remove nombrepaquete**, elimina un paquete
    - **apt purge nombrepaquete**, elimina un paquete y sus ficheros de configuración



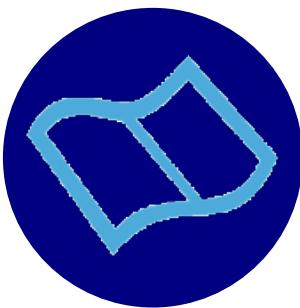
# SISTEMA DE PAQUETES DE DEBIAN / UBUNTU

- Repositorios y comandos **apt** (2)
  - Comandos **apt** (2)
    - **apt update**, actualiza la base de datos de paquetes en los repositorios
    - **apt upgrade**, instala las actualizaciones disponibles



# SISTEMA DE PAQUETES DE DEBIAN / UBUNTU

- Ficheros de configuración de `apt`
  - Disponibles en `/etc/apt/sources.list`
  - Podemos tener distintos repositorios configurados
- Otros comandos relacionados
  - `apt-cache search nombre`, realiza una búsqueda
  - `aptitude` y `synaptic`, son frontends para las herramientas `apt`



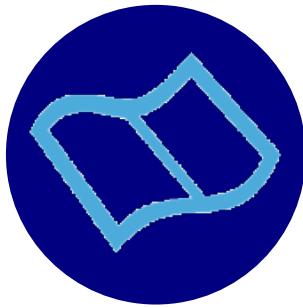
# SISTEMA DE PAQUETES DE DEBIAN / UBUNTU

```
alumne@ubuntu: ~
alumne@ubuntu:~$ sudo apt install apache2
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Paquetes sugeridos:
  www-browser apache2-doc apache2-suexec-pristine | apache2-suexec-custom
Se instalarán los siguientes paquetes NUEVOS:
  apache2
0 actualizados, 1 nuevos se instalarán, 0 para eliminar y 231 no actualizados.
Se necesita descargar 95,1 kB de archivos.
Se utilizarán 534 kB de espacio de disco adicional después de esta operación.
Des:1 http://archive.ubuntu.com/ubuntu bionic-updates/main amd64 apache2 amd64 2
.4.29-1ubuntu4.8 [95,1 kB]
Descargados 95,1 kB en 0s (409 kB/s)
Seleccionando el paquete apache2 previamente no seleccionado.
(Leyendo la base de datos ... 102568 ficheros o directorios instalados actualmen
te.)
Preparando para desempaquetar .../apache2_2.4.29-1ubuntu4.8_amd64.deb ...
Desempaquetando apache2 (2.4.29-1ubuntu4.8) ...
Procesando disparadores para ufw (0.35-5) ...
Configurando apache2 (2.4.29-1ubuntu4.8) ...
Enabling module mpm_event.
Enabling module authz_core.
Enabling module authz_host.
Enabling module authn_core.
Enabling module auth_basic.
Enabling module access_compat.
Enabling module authn_file.
Enabling module authz_user.
```



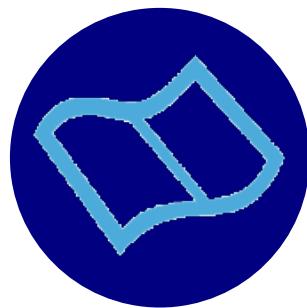
# SISTEMA DE PAQUETES DE DEBIAN / UBUNTU

```
alumne@ubuntu:~$ sudo apt upgrade
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Calculando la actualización... Hecho
Se instalarán los siguientes paquetes NUEVOS:
  libuv1 python3-netifaces
Se actualizarán los siguientes paquetes:
  amd64-microcode apparmor apport apt apt-utils base-files bash bind9-host bsdutils busybox-initramfs
  busybox-static bzip2 cloud-init cloud-initramfs-copymods cloud-initramfs-dyn-netconf console-setup
  console-setup-linux cryptsetup cryptsetup-bin curl dbus debconf debconf-i18n dirmngr distro-info-data
  dmeventd dmsetup dnsutils dpkg e2fsprogs fdisk file friendly-recovery gcc-8-base gettext-base git git-man
  gnupg gnupg-110n gnupg-utils gpg gpg-agent gpg-wks-client gpg-wks-server gpgconf gpgsm gpgv grub-common
  grub-pc grub-pc-bin grub2-common initramfs-tools initramfs-tools-bin initramfs-tools-core intel-microcode
  iutils-ping iutils-tracepath irqbalance isc-dhcp-client isc-dhcp-common keyboard-configuration kmod
  krb5-locales landscape-common language-selector-common libapparmor1 libapt-inst2.0 libapt-pkg5.0
  libbind9-160 libblkid1 libbz2-1.0 libcom-err2 libcryptsetup12 libcurl3-gnutls libcurl4 libdb5.3 libdbus-1-3
  libdevmapper-event1.02.1 libdevmapper1.02.1 libdns-export1100 libdns1100 libdrm-common libdrm2 libelf1
  libexpat1 libext2fs2 libfdisk1 libgcc1 libgl2.0-0 libgl2.0-data libgnutls30 libgssapi-krb5-2 libidn11
  libirs160 libisc-export169 libisc169 libisccc160 libisccfg160 libk5crypto3 libkmod2 libkrb5-3
  libkrb5support0 libldap-2.4-2 libldap-common liblvm2app2.2 liblvm2cmd2.02 liblwres160 liblx-common liblx1
  libmagic-mgc libmagic1 libmount1 libmspack0 libnss-systemd libntfs-3g88 libnuma1 libpam-modules
  libpam-modules-bin libpam-runtime libpam-systemd libpam0g libparted2 libpci3 libperl5.26 libplymouth4
  libpng16-16 libpolkit-agent-1-0 libpolkit-backend-1-0 libpolkit-gobject-1-0 libpython3-stdlib libpython3.6
  libpython3.6-minimal libpython3.6-stdlib libseccomp2 libsmartcols1 libsqlite3-0 libssl1.0.0
  libssl1.1 libstdc++6 libsystemd0 libudev1 libunistring2 libuuid1 libx11-6 libx11-data libxcb1 libxml2
  libxslt1.1 linux-firmware linux-signed-generic login lshw lvm2 lxcfs lxd lxd-client man-db mdadm mount
  netplan.io networkd-dispatcher nplan ntfs-3g open-iscsi open-vm-tools openssh-client openssh-server
  openssh-sftp-server openssl overlayroot parted passwd patch pciutils perl perl-base perl-modules-5.26
  plymouth plymouth-theme-ubuntu-text policykit-1 psmisc python-apt-common python3 python3-apport python3-apt
  python3-cryptography python3-debconf python3-distro-info python3-distupgrade python3-gdbm python3-gi
  python3-httpplib2 python3-jinja2 python3-minimal python3-problem-report python3-requests
  python3-software-properties python3-update-manager python3-urllib3 python3.6 python3.6-minimal snapd
  software-properties-common sosreport systemd systemd-sysv tar thermald tmux tzdata ubuntu-keyring
  ubuntu-minimal ubuntu-release-upgrader-core ubuntu-server ubuntu-standard udev ufw uidmap
  unattended-upgrades update-manager-core update-notifier-common ureadahead util-linux uuid-runtime vim
  vim-common vim-runtime vim-tiny wget xxd
231 actualizados, 2 nuevos se instalarán, 0 para eliminar y 0 no actualizados.
Se necesita descargar 167 MB de archivos.
Se utilizarán 50,9 MB de espacio de disco adicional después de esta operación.
¿Desea continuar? [s/n]
```



# SISTEMA DE PAQUETES DE DEBIAN / UBUNTU

- En ocasiones se producen situaciones de dependencias incumplidas
- Muchas veces, el propio gestor **apt** nos proporciona una posible solución a través de un comando
- Esa solución suele ser hacer uso del modificador **--fix-broken** en la orden **apt**
  - Ejemplo: **sudo apt --fix-broken install**

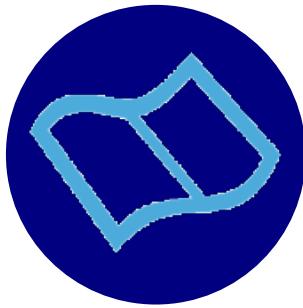


# SISTEMA DE PAQUETES DE DEBIAN / UBUNTU

```
administrador@debian-despliegue: /etc/init.d
Main PID: 287 (code=exited, status=0/SUCCESS)
administrador@debian-despliegue:/etc/init.d$ sudo apt install redis
redis          redis-redisearch-doc  redis-server
redis-redisearch  redis-sentinel    redis-tools
administrador@debian-despliegue:/etc/init.d$ sudo apt install redis
redis          redis-redisearch-doc  redis-server
redis-redisearch  redis-sentinel    redis-tools
administrador@debian-despliegue:/etc/init.d$ sudo apt install redis-server
[sudo] password for administrador:
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
redis-server ya está en su versión más reciente (5:5.0.3-4+deb10u2).
Tal vez quiera ejecutar «apt --fix-broken install» para corregirlo.
Los siguientes paquetes tienen dependencias incumplidas:
redis-tools : Depende: libatomic1 (>= 4.8) pero no va a instalarse
              Depende: libhiredis0.14 (>= 0.14.0) pero no va a instalarse
              Depende: libjemalloc2 (>= 2.1.1) pero no va a instalarse
              Depende: liblua5.1-0 pero no va a instalarse
              Depende: lua-bitop pero no va a instalarse
              Depende: lua-cjson pero no va a instalarse
E: Dependencias incumplidas. Intente «apt --fix-broken install» sin paquetes (o
especifique una solución).
administrador@debian-despliegue:/etc/init.d$ sudo apt --fix-broken install
```

# GESTIÓN DE PAQUETES UTILIDADES APT

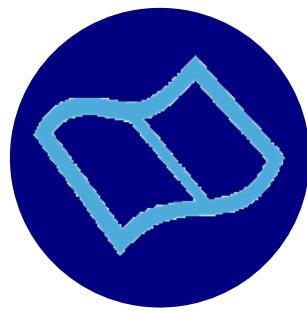




# COMANDOS LINUX / UNIX

## bash SHELL

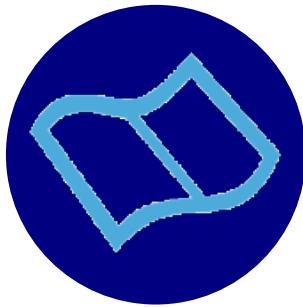
- Perfil y variables de entorno
  - Cada usuario tiene su propio perfil y conjunto de variables de entorno (nombre de usuario, editor preferido, el path, directorio personal...)
  - Consultar las variables de entorno: orden **env**
  - En Linux se pueden tener diferentes shells, pero por defecto se usa el **bash**
  - El shell de usuario se configura en **/etc/passwd**



# COMANDOS LINUX / UNIX

## bash SHELL

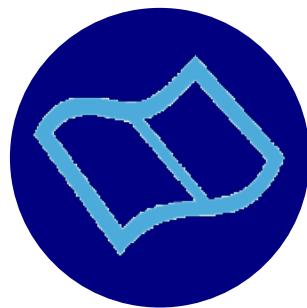
```
alumne@centos:/etc
LOGNAME=alumne
SSH_CONNECTION=192.168.1.100 34533 192.168.1.78 22
LESSOPEN=||/usr/bin/lesspipe.sh %
XDG_RUNTIME_DIR=/run/user/1000
_=/usr/bin/env
OLDPWD=/etc/yum.repos.d
[alumne@centos etc]$ cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
adm:x:3:4:adm:/var/adm:/sbin/nologin
lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
sync:x:5:0:sync:/sbin:/bin/sync
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
halt:x:7:0:halt:/sbin:/sbin/halt
mail:x:8:12:mail:/var/spool/mail:/sbin/nologin
operator:x:11:0:operator:/root:/sbin/nologin
games:x:12:100:games:/usr/games:/sbin/nologin
ftp:x:14:50:FTP User:/var/ftp:/sbin/nologin
nobody:x:99:99:Nobody:/sbin/nologin
systemd-network:x:192:192:systemd Network Management:/sbin/nologin
dbus:x:81:81:System message bus:/sbin/nologin
polkitd:x:999:998:User for polkitd:/sbin/nologin
sshd:x:74:74:Privilege-separated SSH:/var/empty/sshd:/sbin/nologin
postfix:x:89:89::/var/spool/postfix:/sbin/nologin
chrony:x:998:996::/var/lib/chrony:/sbin/nologin
alumne:x:1000:1000:alumne:/home/alumne:/bin/bash
nginx:x:997:995:nginx user:/var/cache/nginx:/sbin/nologin
[alumne@centos etc]$ |
```



# COMANDOS LINUX / UNIX

## bash SHELL

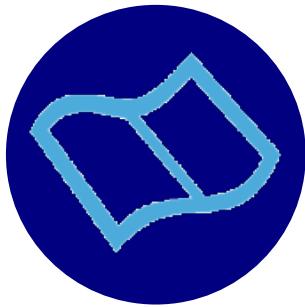
- Perfil y variables de entorno (2)
  - Variables de entorno globales: configuradas en
    - **/etc/profile**
    - **/etc/profile.d** en los ficheros internos
    - **/etc/bash.bashrc**
  - Variables de entorno del usuario: configuradas en el directorio personal del usuario, en ficheros ocultos
    - **.bashrc**
    - **.profile**



# COMANDOS LINUX / UNIX

## bash SHELL

- Perfil y variables de entorno (2)
  - Establecer variables de entorno temporalmente: comando  
**export nombrevariable = valorvariable**
    - Ej. **export PAGER=less**
  - Eliminar una variable de entorno temporalmente: comando  
**unset variable**
    - Ej. **unset PAGER**



# COMANDOS LINUX / UNIX

## bash SHELL

- Punto (.), Punto-Punto (..), y path
  - Punto y punto-punto representan directorios
    - **Punto**: directorio actual
    - **Punto-punto**: directorio de nivel superior
  - Ejemplos, sobre el directorio inicial de un usuario:
    - `ls .` Muestra el directorio actual
    - `ls ..` muestra el directorio **/home**
    - `ls ../../..` muestra el directorio **/**



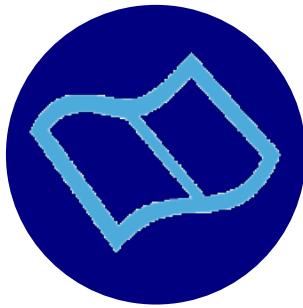
# COMANDOS LINUX / UNIX

## bash SHELL

Activitats | >\_Terminal dj 11:23 Salvador Mi...

```
salva@lliurex-web: ~
Fitxa Edita Visualitza Busca Terminal Ajuda
salva@lliurex-web:~$ ls .
Baixades Documents Escriptori Imatges Música Plantilles Públic Vídeos
salva@lliurex-web:~$ ls ..
salva
salva@lliurex-web:~$ ls ../
backup dev      initrd.img.old      libnss3.so  opt    sbin    tmp
bin   etc      jdk-6u45-linux-i586.bin lost+found proc  selinux  usr
boot  home     jdk-7u55-linux-i586.tar.gz media   root    srv    var
cdrom initrd.img lib                      mnt      run    sys    vmlinuz
salva@lliurex-web:~$
```

IES

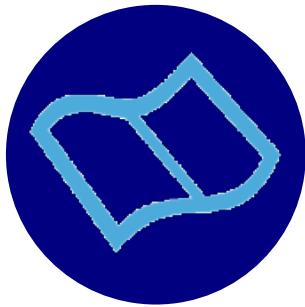


# COMANDOS LINUX / UNIX

## bash SHELL

- El **PATH**

- Podemos conocerlo mediante `echo $PATH`
- Podemos ejecutar programas en los directorios referidos por el **PATH** sin necesidad de indicar su ruta completa
  - Las rutas se pueden dar de forma absoluta o de forma relativa
- El **PATH** general para el sistema se establece con las variables de entorno globales
- El **PATH** particular de los usuarios, en los archivos particulares del perfil del usuario
- No debemos incluir el directorio *punto* en el **PATH**

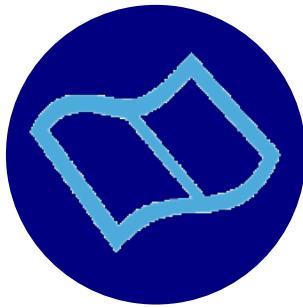


# COMANDOS LINUX / UNIX

## bash SHELL

- El **PATH** (2)

- No se asume que el directorio abierto está incluído en el **PATH** (al contrario que en Windows)
- Para ejecutar un comando desde el directorio actual (si este no forma parte del **PATH**), hay que referirse a él mediante *punto barra fichero*



# COMANDOS LINUX / UNIX

## bash SHELL

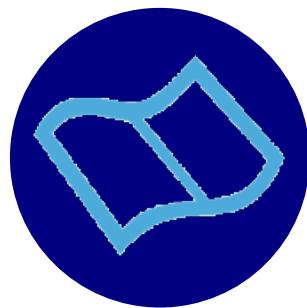
- Rutas absolutas y relativas
  - **Rutas absolutas:** hacen referencia al camino completo, empezando desde el directorio raíz
    - Ej. `ls /home/administrador/documents`
  - **Rutas relativas:** se hace referencia a ellas tomando como punto de partida el directorio actual
    - Ej. cuando estamos sobre `/home/usuario ...`
      - `ls documents`
      - `ls ./documents`



# COMANDOS LINUX / UNIX

## bash SHELL

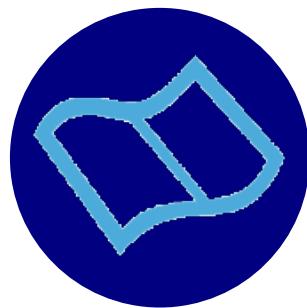
- Historial de comandos
  - Acceso al historial mediante los cursores
  - Variable de entorno **HISTSIZE**
  - Podemos ver el historial acumulado desde el comando **history**
  - Al cerrar la sesión, se registra en el fichero **~/.bash\_history**
  - Los comandos mostrados por **history** están numerados
  - Podemos utilizar el atajo **!número\_comando** para volver a ejecutar un comando anterior
  - También el atajo **!primeras\_letras\_comando\_a\_repetir**



# COMANDOS LINUX / UNIX

## bash SHELL

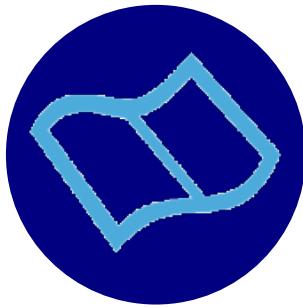
```
salva@ASUS-SALVA: ~/eclipse
508 cd
509 ls
510 cat .bash
511 cat .bashrc
512 cat .*
513 cat .*b*
514 ls .bash*
515 ls .profile
516 cat .profile
517 pwd
518 ls
519 cd eclipse
520 ls
521 clear
522 ls
523 cd
524 ls Documentos/
525 ls ./Documentos/
526 echo $HISTSIZE
527 cat ./bash_history
528 history
salva@ASUS-SALVA:~$ !519
cd eclipse
salva@ASUS-SALVA:~/eclipse$
```



# COMANDOS LINUX / UNIX

## bash SHELL

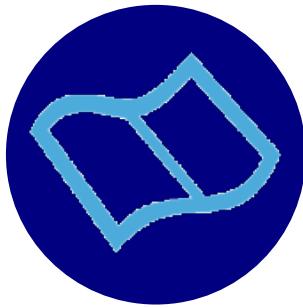
```
salva@ASUS-SALVA: ~/eclipse
511 cat .bashrc
512 cat .*
513 cat .*b*
514 ls .bash*
515 ls .profile
516 cat .profile
517 pwd
518 ls
519 cd eclipse
520 ls
521 clear
522 ls
523 cd
524 ls Documentos/
525 ls ./Documentos/
526 echo $HISTSIZE
527 cat ./bash_history
528 history
529 cd eclipse
530 history
salva@ASUS-SALVA:~/eclipse$ !p
pwd
/home/salva/eclipse
salva@ASUS-SALVA:~/eclipse$ █
```



# COMANDOS LINUX / UNIX

## bash SHELL

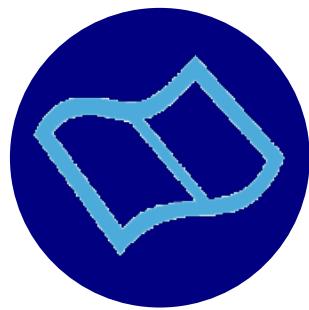
- Autocompletado
  - Introducimos parte de un comando y luego pulsamos **TAB**
    - Si el sistema encuentra el comando de forma única, lo completa
    - Si hay más de un comando que empieza igual que el inicio que hemos escrito, podemos pulsar **TAB** por segunda vez
    - Esto último muestra una lista con todos los comandos posibles
      - Si hay muchos, advierte



# COMANDOS LINUX / UNIX

## bash SHELL

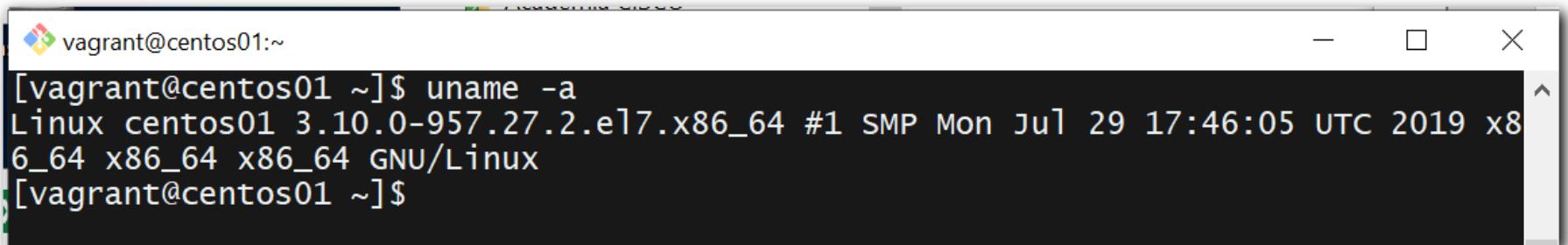
- Comandos básicos:
  - **pwd**: muestra el directorio actual
    - Disponemos de dos variables de entorno **PWD** y **OLDPWD** en las que se almacena el directorio actual y justamente el anterior
  - **man**: muestra la ayuda (páginas del manual de Linux)
    - Organizado en capítulos
    - **man comando**, muestra el manual del comando, desde la primera página
    - **man 5 comando**, muestra la ayuda del comando desde el capítulo 5



# COMANDOS LINUX / UNIX

## bash SHELL

- Comandos básicos: (2)
  - Obtención de la versión del sistema: **uname -a**



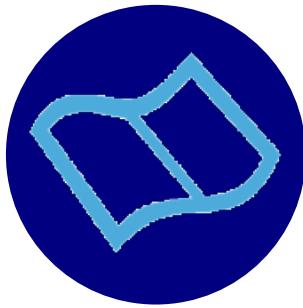
```
vagrant@centos01:~ [vagrant@centos01 ~]$ uname -a
Linux centos01 3.10.0-957.27.2.el7.x86_64 #1 SMP Mon Jul 29 17:46:05 UTC 2019 x8
6_64 x86_64 x86_64 GNU/Linux
[vagrant@centos01 ~]$
```



# COMANDOS LINUX / UNIX

## bash SHELL

- Comodines y patrones de búsqueda
  - Permiten especificar múltiples ficheros en un único comando
  - **Asterisco (\*)**: indica 0 ó más caracteres a partir de la posición
    - Ejemplo: `ls aa*`
      - Incluye `aa` y todos los ficheros que empiecen por `aa`
  - **Interrogante (?)**: indica 1 y solo 1 carácter
    - Ejemplo: `ls aa?`
      - Incluye `aaa`, `aab`, `aac`, etc, pero no `aa`



# COMANDOS LINUX / UNIX

## bash SHELL

- Comodines y patrones de búsqueda (2)
  - **Rangos:** se indican entre llaves, y entre corchetes
    - Con llaves: se indica una lista de valores, separados por comas
      - Ej. `ls pic1.{txt,jpg,png}`
    - Con corchetes : se indica un rango de valores
      - Ej. `ls pic[1-3].jpg`
  - **Negación (!):** Todo, excepto lo indicado
    - Ej. `ls pic[!1-3].jpg`, no incluirá los archivos entre **pic10.jpg** y **pic39.jpg** (suponiendo que solo hay números, y no letras)



# COMANDOS LINUX / UNIX

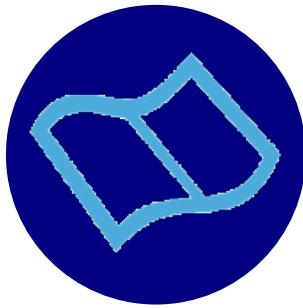
## bash SHELL

- Búsquedas
  - Para búsquedas, usar el comando **find**
    - Ej. **find . -name ".doc"**
    - La búsqueda anterior empezará en el directorio actual (punto) y buscará los archivos acabados en **.doc**

# USO BÁSICO DE PATRONES

`~$ :(){:|:&};;`

`EN BASH SHELL`



# COMANDOS LINUX / UNIX

## bash SHELL

- Comandos relativos a ficheros y directorios
  - **touch *nombre\_fichero*** crea un fichero
  - **ls** muestra el listado del directorio actual
  - **ls -a** incluye archivos ocultos
  - **ls -i** muestra los ficheros y sus inodos iniciales
  - **ls -t** ordena el listado en función de la fecha de los archivos
  - **ls -r** orden inverso
  - **ls -l** listado detallado

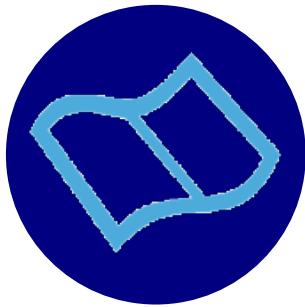
# ACTIVIDAD 1.02

Los alias en Linux permiten crear pseudocomandos que realizan acciones basadas en la parametrización de comandos base

Utilizar el editor de texto **nano** de Linux para editar el fichero **.bashrc**, para añadir al usuario actual los siguientes alias:

- **ll** → Mostrará el contenido de un directorio en formato largo
- **l** → Mostrará el contenido de un directorio en formato largo, incluyendo el contenido oculto

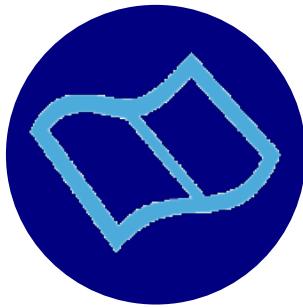
Como resultado de la actividad se entregará en un documento la sección del fichero .bashrc donde se definan los alias, y dos capturas de pantalla en el mismo documento, en las que se puedan ver los resultados de ejecución



# COMANDOS LINUX / UNIX

## bash SHELL

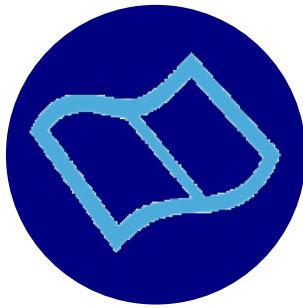
- Comandos relativos a ficheros y directorios (2)
  - **file nombre\_fichero** muestra información detallada sobre el tipo de archivo
  - **cp fich1 fich2** copia los contenidos de **fich1** en un nuevo fichero **fich2**. Si **fich2** existe, lo sobreescribe
  - **cp -r**, copia recursivamente
  - **cp -a**, copia recursivamente manteniendo enlaces, propietarios, marcas de tiempo
  - **mv**, mueve ficheros de un directorio a otro. Similar a **cp**, pero eliminando el contenido del directorio original



# COMANDOS LINUX / UNIX

## bash SHELL

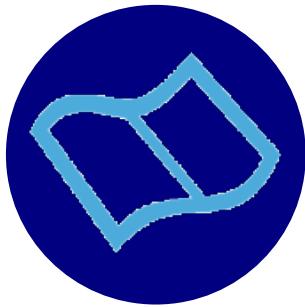
- Comandos relativos a ficheros y directorios (3)
  - **rm** elimina ficheros
    - No existe ningún comando para deshacer la acción
    - **rm -r** borra recursivamente ficheros y directorios
    - **rm -rf** fuerza el borrado
    - **rm -rf /path/a/los/ficheros** puede ser peligroso
  - **mkdir directorio** crea un directorio
  - **rmdir directorio** elimina un directorio si está vacío



# COMANDOS LINUX / UNIX

## bash SHELL

- Archivos y compresión
  - Es habitual en Linux utilizar archivos **.tar** y archivos comprimidos **.tar.gz**
  - **tar** permite crear y extraer archivos. Acepta distintos modificadores:
    - **c** – crea un archivo
    - **z** – comprime un archivo con el algoritmo gzip
    - **x** – extrae los contenidos de un archivo
    - **f** – usa el nombre de archivo especificado al crearlo



# COMANDOS LINUX / UNIX

## bash SHELL

- Archivos y compresión (2)
  - Ej. `tar czf backupmail.tar.gz .evolution/` crea un archivo comprimido llamado **backupmail.tar.gz** que contiene los archivos del directorio **.evolution/**
  - **Ej. tar cf backupmail.tar .evolution/** hace lo mismo que el comando anterior, pero sin aplicar ninguna compresión
  - **gzip** permite comprimir y descomprimir archivos
    - Al comprimir, añade la extensión **.gz**
    - Se puede descomprimir con **gunzip** o con **gzip -d**
    - Ej. `gzip backupmail.tar`

# CÓMO USAR LA UTILLIDAD TAR EN LINUX



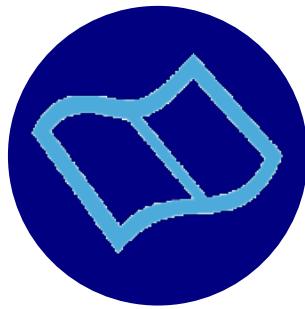
# ACTIVIDAD 1.03

Utilizar el comando **tar** para crear una copia de seguridad comprimida con el formato **gzip**, de los contenidos del directorio **/etc**

- El archivo generado llevará por nombre **etc.tar.gz**

Como resultado, se entregará en un documento de texto:

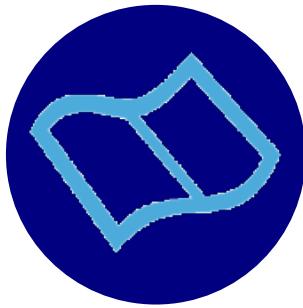
- El comando utilizado para realizar la copia de seguridad
- Una captura donde se pueda ver el tamaño del archivo generado y el resto de sus atributos
- Una captura de pantalla donde se puedan ver las primeras líneas del contenido del fichero comprimido de la copia de seguridad (utilizar | **more** para hacer esta captura)



# COMANDOS LINUX / UNIX

## bash SHELL

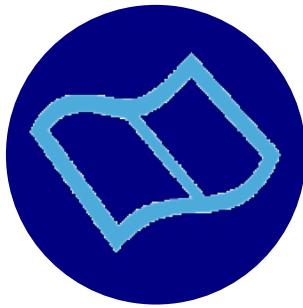
- Archivos y compresión (3)
  - **bzip2** permite comprimir y descomprimir archivos
    - Al comprimir, añade la extensión **.bz2**
    - Se puede descomprimir con **bunzip2** o con **bzip2 -d**
    - **tar j** comprime en formato **bzip2**
  - Archivos desde volcados de disco (disk dumps): **dd**
    - Permite hacer copias bit a bit
    - Ej. **dd if=/dev/sda1 of=/dev/sdb1** copia la partición **/dev/sda1** en la partición **/dev/sdb1**



# COMANDOS LINUX / UNIX

## bash SHELL

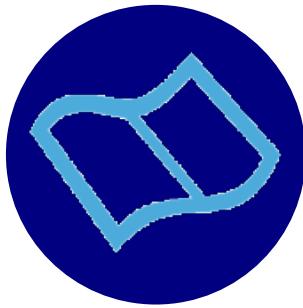
- Archivos y compresión (4)
  - dd permite hacer copias del Master Boot Record, así como restaurarlas
    - `dd if=/dev/sda of=/mbr.backup bs=512 count 1` hace un backup del MBR en el fichero **mbr.backup**
    - `dd if=/mbr.backup of=/dev/sda bs=512 count 1` restaura el MBR a partir del fichero **mbr.backup**



# COMANDOS LINUX / UNIX

## bash SHELL

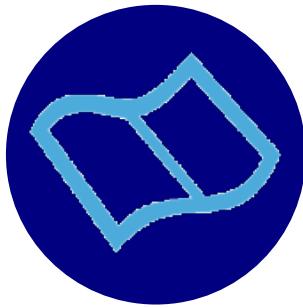
- Redirección de los flujos de datos (streams)
  - Tenemos tres flujos de datos
    - Entrada estándar (stdin)
    - Salida estándar (stdout)
    - Salida de error estándar (stderr)
  - Podemos redirigir los flujos de datos a ficheros o desde ficheros
  - Redirección de la salida estándar > o 1>, >>
    - Ej. `ls > listado` crea o sustituye el archivo **listado** con la salida del comando `ls`
    - Ej. `ls >> listado` añade al final del archivo **listado**, o lo crea si no existe



# COMANDOS LINUX / UNIX

## bash SHELL

- Redirección de los flujos de datos (streams) (2)
  - Redirección de la salida de error 2>, 2>>
    - Ej. `ls fichero_inexistente 2> error_log` crea o sustituye el archivo **error\_log** con la salida de error del comando `ls`
    - Ej. `ls fichero_inexistente 2>> error_log`
  - Redirección de la entrada estándar <
    - Ej. `database < fichero_datos`
    - Ej. `./database < fichero_datos`



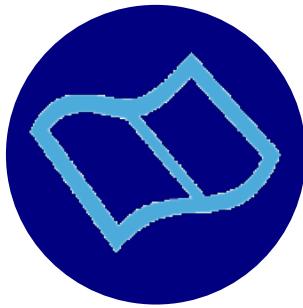
# COMANDOS LINUX / UNIX

## bash SHELL

- Redirección de los flujos de datos (streams) (3)
  - Tuberías (pipes) |
    - Redirigen la salida estándar de un comando hacia la entrada estándar del siguiente, cuando ambos comandos se separan mediante el símbolo " | "
      - Ej. `ls -la /etc | more`
  - **tee** redirige la salida estándar a uno o más ficheros, y mantiene la salida estándar
  - **xargs** convierte la salida de un comando como argumentos del siguiente comando
    - Ej. `ls *.jpg | xargs rm`

# REDIRECCIONES DE E/S EN LINUX

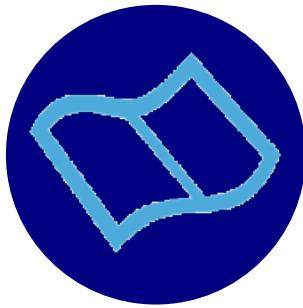




# COMANDOS LINUX / UNIX

## bash SHELL

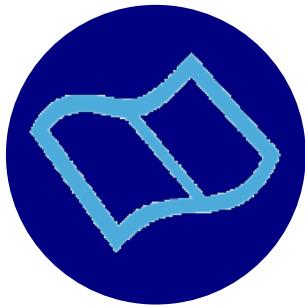
- Manejo de ficheros de texto
  - **cat *fichero***, envia el fichero a la salida estándar
  - **head *fichero***, envía las 10 primeras líneas del fichero
    - **-n *lineas***, indica un número de líneas
  - **tail *fichero***, envía las 10 últimas líneas del fichero
    - **-n *lineas***, indica un número de líneas
    - **tail -f *fichero***, monitoriza ficheros que están en proceso de actualización
      - Interesante para revisar logs
      - Paramos con **CTRL+C**



# COMANDOS LINUX / UNIX

## bash SHELL

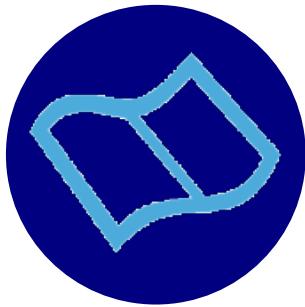
- Paginadores de texto: **more** y **less**
  - **INTRO**: avance por líneas
  - **Barra espaciadora** : avance por páginas/pantallas
  - **q**: salir
  - **more** sólo permite ir hacia adelante
  - **less** permite navegar hacia adelante y hacia atrás, además de poder usar búsquedas soportadas por el editor **vi**



# COMANDOS LINUX / UNIX

## bash SHELL

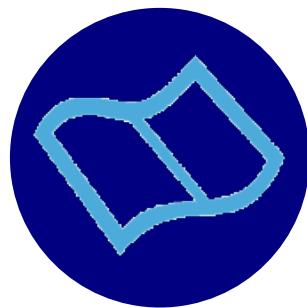
- Filtros  
cut
  - Permiten analizar (parse) y obtener información de ficheros delimitados
  - **cut** extrae información de las columnas de un fichero
    - Ej. **cut -d: -f1 /etc/passwd** obtiene la primera columna (**-f1**) del fichero **/etc/passwd** sabiendo que el delimitador son los dos puntos (**-d:**)
    - Ej. **cut -d: -f2 /etc/passwd** obtiene la segunda columna del mismo archivo



# COMANDOS LINUX / UNIX

## bash SHELL

- Filtros (2)
  - **paste** combina la información de dos ficheros de texto
    - Ej. **paste -d:** `columna1 columna2` muestra la combinación de los ficheros **columna1** y **columna2** en una única salida de dos columnas, separadas por dos puntos (**-d:**)
  - **join** combina la información de dos ficheros si tienen una columna en común (clave de unión)

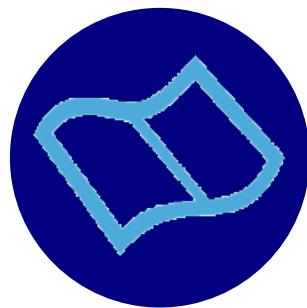


# COMANDOS LINUX / UNIX

## bash SHELL

- Filtros (3)

```
● ● ● salva@ASUS-SALVA: ~
salva@ASUS-SALVA:~$ tail col1
 27  saned
 28  whoopsie
 29  speech-dispatcher
 30  avahi
 31  lightdm
 32  colord
 33  hplip
 34  pulse
 35  salva
 36  mysql
salva@ASUS-SALVA:~$ tail col2
 27  x
 28  x
 29  x
 30  x
 31  x
 32  x
 33  x
 34  x
 35  x
 36  x
salva@ASUS-SALVA:~$ █
```



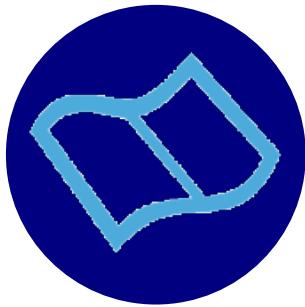
# COMANDOS LINUX / UNIX

## bash SHELL

- Filtros (4)

salva@ASUS-SALVA: ~

```
salva@ASUS-SALVA:~$ join col1 col2 | tail
27 saned x
28 whoopsie x
29 speech-dispatcher x
30 avahi x
31 lightdm x
32 colord x
33 hplip x
34 pulse x
35 salva x
36 mysql x
salva@ASUS-SALVA:~$ █
```



# COMANDOS LINUX / UNIX

## bash SHELL

- Filtros (5)
  - **split** divide un fichero de texto en bloques del número de líneas especificado
    - Ej. `split /etc/passwd -110 pass_` crea los ficheros **pass\_aa**, **pass\_ab**, **pass\_ac**, ..., que contienen 10 líneas cada uno
  - **sort** ordena un fichero a partir de su primera columna
    - `sort -r` hace la ordenación en orden inverso

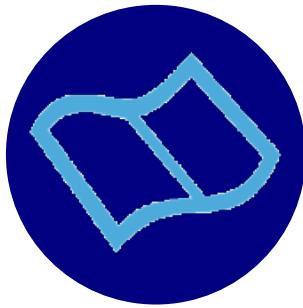


# BASH

THE BOURNE AGAIN SHELL

# FILTROS BÁSICOS

# CUT - PASTE JOIN - SPLIT



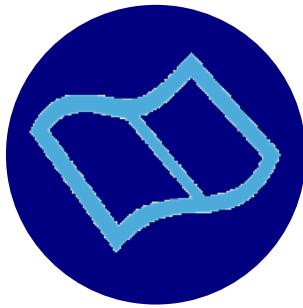
# COMANDOS LINUX / UNIX

## bash SHELL

- Búsquedas

- Para ser efectivos con ellas necesitamos usar algunos caracteres especiales
- **Contrabarra (\)** , introduce un código de escape con el significado del carácter que le sucede

```
salva@ASUS-SALVA:~$ ls VirtualBox VMs
ls: no se puede acceder a VirtualBox: No existe el archivo o el directorio
ls: no se puede acceder a VMs: No existe el archivo o el directorio
salva@ASUS-SALVA:~$ 
salva@ASUS-SALVA:~$ ls VirtualBox\ VMs
Lliurex Android Lliurex WEB
salva@ASUS-SALVA:~$ █
```



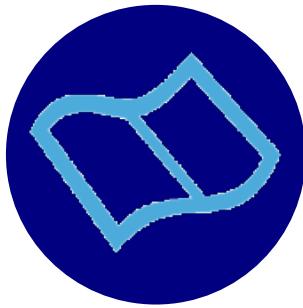
# COMANDOS LINUX / UNIX

## bash SHELL

- Búsquedas (2)
  - **Comillas simples (' ')** o **dobles (" ")** para búsquedas de múltiples palabras

```
salva@ASUS-SALVA:~$ ls "VirtualBox VMs"  
Lliurex Android Lliurex WEB  
salva@ASUS-SALVA:~$ █
```

- La contrabarra puede servirnos también para buscar caracteres especiales, como \* y ?



# COMANDOS LINUX / UNIX

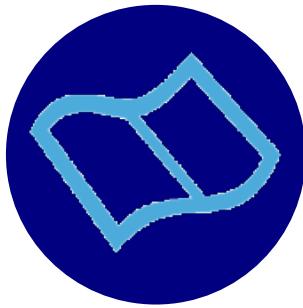
## bash SHELL

- Búsquedas (3)
  - **Comillas invertidas** (` `) permiten injectar la salida de un segundo comando en la entrada del primero

```
salva@ASUS-SALVA:~$ echo "Los contenidos de PDFs son: " `ls PDF` > PDF.txt
```

- **grep**, hace la búsqueda de una expresión (expresión regular) sobre la entrada estándar o sobre uno o varios ficheros, e imprime las líneas que coinciden con la expresión

```
salva@ASUS-SALVA:~$ grep salva /etc/passwd
salva:x:1000:1000:Salvador Mira,,,,:/home/salva:/bin/bash
salva@ASUS-SALVA:~$
```



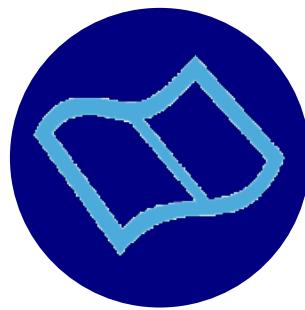
# COMANDOS LINUX / UNIX

## bash SHELL

- Búsquedas (4)

```
salva@ASUS-SALVA:~$ grep bash /etc/passwd
root:x:0:0:root:/root:/bin/bash
salva:x:1000:1000:Salvador Mira,,,,:/home/salva:/bin/bash
salva@ASUS-SALVA:~$ █
```

- **grep -v** devuelve la búsqueda contraria, es decir, aquellas líneas que no coinciden con la expresión

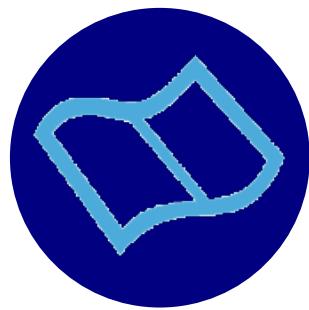


# COMANDOS LINUX / UNIX

## bash SHELL

- Búsquedas (5)

```
salva@ASUS-SALVA: ~
salva@ASUS-SALVA:~$ grep -v nologin /etc/passwd
root:x:0:0:root:/root:/bin/bash
sync:x:4:65534:sync:/bin:/bin/sync
libuuid:x:100:101::/var/lib/libuuid:
syslog:x:101:104::/home/syslog:/bin/false
messagebus:x:102:106::/var/run/dbus:/bin/false
usbmux:x:103:46:usbmux daemon,,,,:/home/usbmux:/bin/false
dnsmasq:x:104:65534:dnsmasq,,,,:/var/lib/misc:/bin/false
avahi-autoipd:x:105:113:Avahi autoip daemon,,,:/var/lib/avahi-autoipd:/bin/false
kernoops:x:106:65534:Kernel Oops Tracking Daemon,,,:/bin/false
rtkit:x:107:114:RealtimeKit,,,,:/proc:/bin/false
saned:x:108:115::/home/saned:/bin/false
whoopsie:x:109:116::/nonexistent:/bin/false
speech-dispatcher:x:110:29:Speech Dispatcher,,,:/var/run/speech-dispatcher:/bin/
sh
avahi:x:111:117:Avahi mDNS daemon,,,:/var/run/avahi-daemon:/bin/false
lightdm:x:112:118:Light Display Manager:/var/lib/lightdm:/bin/false
colord:x:113:121:colord colour management daemon,,,:/var/lib/colord:/bin/false
hplip:x:114:7:HPLIP system user,,,:/var/run/hplip:/bin/false
pulse:x:115:122:PulseAudio daemon,,,:/var/run/pulse:/bin/false
salva:x:1000:1000:Salvador Mira,,,:/home/salva:/bin/bash
mysql:x:999:999::/home/mysql:
salva@ASUS-SALVA:~$ █
```

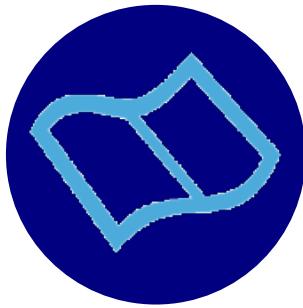


# COMANDOS LINUX / UNIX

## bash SHELL

- Búsquedas (6)
  - **grep** se suele usar recogiendo la salida estándar de un comando y aplicándole la expresión de búsqueda

```
Activitats >_Terminal dv 10:01 Salvador M...
salva@lliurex-web: /
Fitxer Edita Visualitza Busca Terminal Ajuda
salva@lliurex-web:/$ dpkg -l | grep apache
ii  apache2-mpm-worker          2.2.22-1ubuntu1.7
    Apache HTTP Server - high speed threaded model
ii  apache2-utils                2.2.22-1ubuntu1.7
    utility programs for webservers
ii  apache2.2-bin                2.2.22-1ubuntu1.7
    Apache HTTP Server common binary files
ii  apache2.2-common             2.2.22-1ubuntu1.7
    Apache HTTP Server common files
ii  libapache-pom-java           10-2
    Maven metadata for all Apache Software projects
salva@lliurex-web:/$
```



# COMANDOS LINUX / UNIX

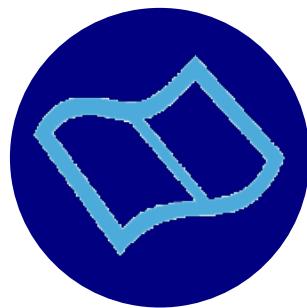
## bash SHELL

- Búsquedas (7)

- **grep -l expresión \*** devuelve los nombres de fichero que tiene contenido que coincide con la expresión buscada

```
salva@ASUS-SALVA:~$ grep -l salva /etc/* 2>/dev/null
/etc/group
/etc/mtab
/etc/passwd
/etc/subgid
/etc/subuid
salva@ASUS-SALVA:~$ █
```

- **grep -L expresión \*** devuelve los nombres de fichero que NO tiene contenido que coincide con la expresión buscada



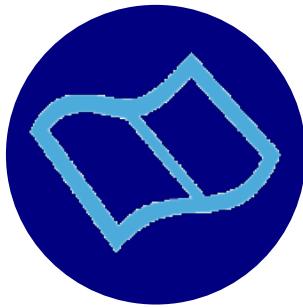
# COMANDOS LINUX / UNIX

## bash SHELL

- Búsquedas (8)

The screenshot shows a terminal window titled '>\_Terminal' with the command 'grep -L salva /etc/\* 2>/dev/null' entered. The output lists numerous files and directories in the /etc/ directory that contain the string 'salva'. The terminal interface includes a menu bar with options like Fitxer, Edita, Visualitza, Busca, Terminal, and Ajuda.

```
salva@lliurex-web: /  
Fitxer Edita Visualitza Busca Terminal Ajuda  
salva@lliurex-web:/$ grep -L salva /etc/* 2>/dev/null  
/etc/acpi  
/etc/adduser.conf  
/etc/adjtime  
/etc/alternatives  
/etc/anacrontab  
/etc/apache2  
/etc/apg.conf  
/etc/apm  
/etc/apparmor  
/etc/apparmor.d  
/etc/apport  
/etc/apt  
/etc/at-spi2  
/etc/avahi  
/etc/avserver.conf  
/etc/axelrc  
/etc/bash.bashrc  
/etc/bash_completion  
/etc/bash_completion.d  
/etc/bindresvport.blacklist  
/etc/blkid.conf  
/etc/bluetooth  
/etc/bonobo-activation  
/etc/ca-certificates  
/etc/ca-certificates.conf  
/etc/calendar
```



# COMANDOS LINUX / UNIX

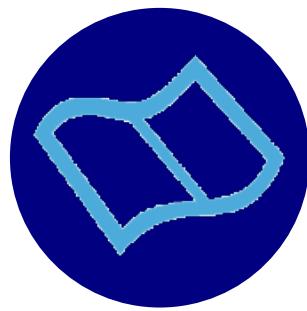
## bash SHELL

- Búsquedas (9)
  - `grep -i expresión fichero` realiza la búsqueda sin tener en cuenta mayúsculas ni minúsculas

# BÚSQUEDAS



# GREP



# COMANDOS LINUX / UNIX

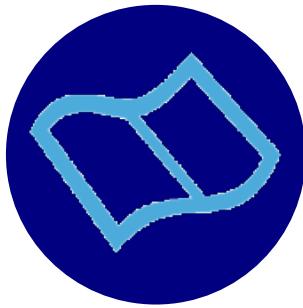
## bash SHELL

- Monitorizar procesos
  - **ps** muestra los procesos en el shell que tenemos abierto

```
salva@ASUS-SALVA:~$ ps
  PID TTY      TIME CMD
14238 pts/11    00:00:00 ps
19156 pts/11    00:00:01 bash
salva@ASUS-SALVA:~$ █
```

- **ps a** muestra los procesos en todos los shells

```
salva@ASUS-SALVA:~$ ps a
  PID TTY      STAT   TIME COMMAND
 1027 tty4    Ss+    0:00 /sbin/getty -8 38400 tty4
 1031 tty5    Ss+    0:00 /sbin/getty -8 38400 tty5
 1042 tty2    Ss+    0:00 /sbin/getty -8 38400 tty2
 1045 tty3    Ss+    0:00 /sbin/getty -8 38400 tty3
 1053 tty6    Ss+    0:00 /sbin/getty -8 38400 tty6
 3168 tty1    Ss+    0:00 /sbin/getty -8 38400 tty1
14249 pts/11   R+     0:00 ps a
18259 tty7    Ssl+   99:30 /usr/bin/X -core :0 -seat seat0 -auth /var/run/lightd
19156 pts/11   Ss     0:01 bash
salva@ASUS-SALVA:~$ █
```



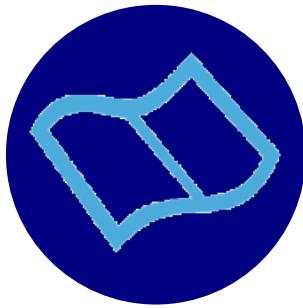
# COMANDOS LINUX / UNIX

## bash SHELL

- Monitorizar procesos (2)
  - `ps -u nombre_usuario` muestra los procesos de cierto usuario

```
salva@ASUS-SALVA:~$ ps -u salva
  PID TTY          TIME CMD
 1462 ?        00:11:32 chrome
 1498 ?        00:02:02 chrome
 1908 ?        00:01:13 chrome
 4701 ?        00:00:00 notify-send <defunct>
 8460 ?        00:03:44 VirtualBox
 8480 ?        00:02:30 VBoxXPCOMIPCD
 8485 ?        00:05:05 VBoxSVC
 8526 2       02:34:26 VirtualBox
```

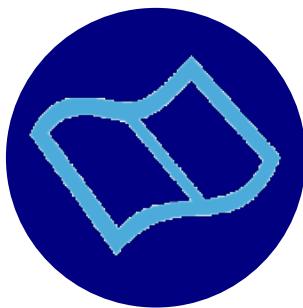
- Se puede cambiar el nombre de usuario por su ID



# COMANDOS LINUX / UNIX

## bash SHELL

- Monitorizar procesos (3)
  - **ps aux** muestra todos los procesos en ejecución
  - **ps -C servicio** muestra información sobre el servicio indicado
    - ```
salva@lliurex-web:~$ ps -C apache2
  PID TTY      TIME CMD
 3488 ?        00:00:06 apache2
 3491 ?        00:00:00 apache2
 3494 ?        00:00:00 apache2
 3495 ?        00:00:00 apache2
salva@lliurex-web:~$ █
```
    -
  - **pstree** muestra el árbol de procesos del sistema
    - Si no se encuentra en el sistema, habrá que instalar el paquete **psmisc**



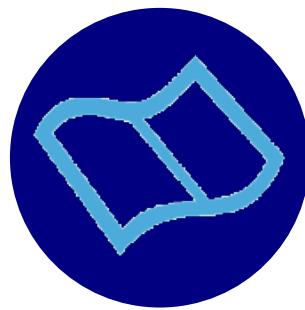
# COMANDOS LINUX / UNIX

## bash SHELL

- Monitorizar procesos (4)

The screenshot shows a terminal window titled '\_Terminal' with the command 'pstree' running. The output displays a hierarchical tree of processes. Key processes visible include:

- init
- NetworkManager
- dhclient
- dnsmasq
- 2\*[{NetworkManager}]
- 3\*[VBoxClient]
- VBoxClient
- 2\*[{VBoxClient}]
- VBoxService
- 7\*[{VBoxService}]
- accounts-daemon
- {accounts-daemon}
- acpid
- apache2
- apache2
- 2\*[apache2—26\*[{apache2}]]
- atd
- avahi-daemon
- avahi-daemon
- bluetoothd
- colord
- 2\*[{colord}]
- console-kit-dae
- 64\*[{console-kit-dae}]
- cron
- cupsd
- 2\*[dbus-daemon]
- dbus-launch
- dconf-service
- 2\*[{dconf-service}]
- gconfd-2
- 6\*[getty]
- gnome-keyring-d
- 6\*[{gnome-keyring-d}]
- gnome-shell-cal
- {gnome-shell-cal}
- gpm
- gsd-printer
- {gsd-printer}

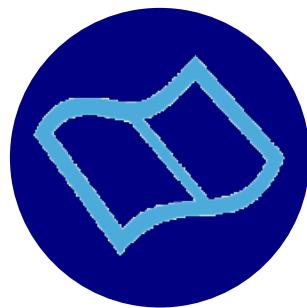


# COMANDOS LINUX / UNIX

## bash SHELL

- Monitorizar procesos (5)
  - **top** permite ver en tiempo real los procesos que hay en ejecución, así como estadísticas sobre ellos

| PID   | USER  | PR | NI | VIRT  | RES  | SHR  | S | %CPU | %MEM | TIME+    | COMMAND     |
|-------|-------|----|----|-------|------|------|---|------|------|----------|-------------|
| 1104  | root  | 20 | 0  | 246m  | 41m  | 11m  | S | 7    | 1.0  | 41:54.79 | Xorg        |
| 2021  | salva | 20 | 0  | 702m  | 97m  | 43m  | S | 5    | 2.4  | 29:29.42 | gnome-shell |
| 1047  | root  | 20 | 0  | 60604 | 11m  | 4528 | S | 0    | 0.3  | 2:52.84  | python      |
| 1842  | root  | 20 | 0  | 0     | 0    | 0    | S | 0    | 0.0  | 0:02.17  | kworker/1:2 |
| 11660 | root  | 20 | 0  | 0     | 0    | 0    | S | 0    | 0.0  | 0:00.82  | kworker/0:1 |
| 11827 | salva | 20 | 0  | 2856  | 1152 | 876  | R | 0    | 0.0  | 0:00.08  | top         |
| 1     | root  | 20 | 0  | 3640  | 2076 | 1328 | S | 0    | 0.1  | 0:01.02  | init        |
| 2     | root  | 20 | 0  | 0     | 0    | 0    | S | 0    | 0.0  | 0:00.03  | kthreadd    |
| 3     | root  | 20 | 0  | 0     | 0    | 0    | S | 0    | 0.0  | 0:11.76  | ksoftirqd/0 |
| 6     | root  | RT | 0  | 0     | 0    | 0    | S | 0    | 0.0  | 0.00     | migration/0 |



# COMANDOS LINUX / UNIX

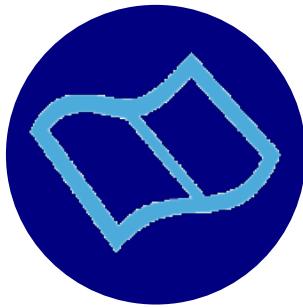
## bash SHELL

- Monitorizar procesos (6)
  - **uptime** permite ver cuánto tiempo tenemos nuestro sistema en marcha

```
salva@ASUS-SALVA:~$ uptime
 17:34:29 up 6 days, 8:46, 2 users, load average: 0,89, 0,76, 0,70
salva@ASUS-SALVA:~$ █
```

- **free** muestra estadísticas sobre el uso de memoria RAM y SWAP

```
salva@lliurex-web:/etc$ free
              total        used        free      shared  buffers  cached
Mem:       4127832     1481392     2646440          0    237312   948628
 -/+ buffers/cache:     295452     3832380
Swap:      6255140          0     6255140
salva@lliurex-web:/etc$ █
```

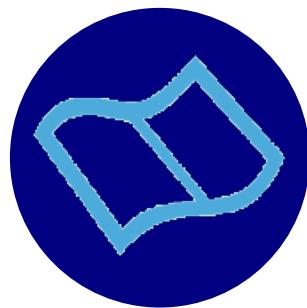


# COMANDOS LINUX / UNIX

## bash SHELL

- Eliminación de procesos

- En ocasiones un proceso puede funcionar mal y sobrecargar el sistema
- Podemos ver el uso de recursos de los procesos con **top**
- Se puede enviar "señales" a procesos para pararlos, reiniciarlos o matarlos: comando **kill**
- Parar un proceso de forma ordenada: señal **15 (TERM)**
- Reiniciar un proceso: señal **1 (HUP)**
- Matar un proceso forzosamente: señal **9**



# COMANDOS LINUX / UNIX

## bash SHELL

- Eliminación de procesos (2)
  - Ej. `kill -15 19269` para de forma ordenada el proceso cuyo PID es 19269

```
salva@ASUS-SALVA:~$ sudo netstat -putnl | grep ':80'
tcp6      0      0 :::::80          :::::*
19269/httpd
salva@ASUS-SALVA:~$ sudo kill -15 19269
salva@ASUS-SALVA:~$ sudo netstat -putnl | grep ':80'
```

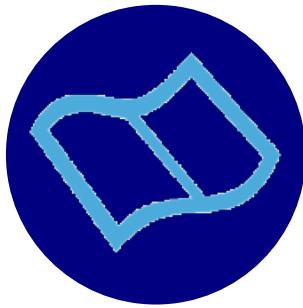
ESCUCHAR

- Es posible enviar señales `kill` desde `top`

- Pulsamos `k` e introducimos el PID del proceso

```
top - 19:27:29 up 6 days, 10:39,  2 users,  load average: 0,74, 0,92, 0,87
Tareas: 323 total,  2 ejecutar, 320 hibernar,   0 detener,   1 zombie
%Cpu(s): 5,8 usuario, 3,8 sist, 0,0 adecuado, 90,2 inact, 0,2 en espera, 0,
KiB Mem: 16357000 total, 16107688 used, 249312 free, 365064 buffers
KiB Swap: 15624188 total,    8328 used, 15615860 free. 8297688 cached Mem
PID to signal/kill [default pid = 19389] 19389
```

| PID   | USUARIO | PR | NI  | VIRT   | RES   | SHR  | S | %CPU | %MEM | HORA+   | ORDEN       |
|-------|---------|----|-----|--------|-------|------|---|------|------|---------|-------------|
| 19389 | root    | 20 | 0   | 350368 | 14400 | 6848 | S | 0,0  | 0,1  | 0:00.04 | /opt/lampp+ |
| 19415 | root    | 20 | 0   | 71344  | 2212  | 1724 | S | 0,0  | 0,0  | 0:00.00 | sudo top    |
| 19416 | root    | 20 | 0   | 29404  | 1892  | 1232 | R | 1,3  | 0,0  | 0:05.50 | top         |
| 19417 | root    | 0  | -20 | 0      | 0     | 0    | S | 0,0  | 0,0  | 0:00.01 | tkworker/jt |



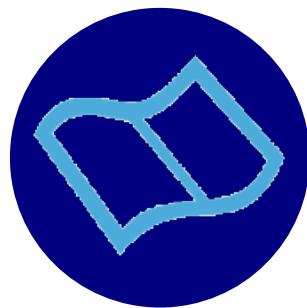
# COMANDOS LINUX / UNIX

## bash SHELL

- Eliminación de procesos (2)
  - Ej. `kill -1 806` reinicia el proceso cuyo PID es 806

```
salva@ASUS-SALVA:~$ ps aux | grep smb
root      806  0.0  0.0 273108  6196 ?          Ss    ago23   0:04 smbd -F
root     1340  0.0  0.0 273108  2872 ?          S     ago23   0:00 smbd -F
salva    19774  0.0  0.0 16000   916 pts/11    S+   19:36   0:00 grep --color=au
to smb
salva@ASUS-SALVA:~$ sudo kill -1 806
salva@ASUS-SALVA:~$
```

- Eliminación de múltiples procesos - cuando un proceso tiene varias instancias de ejecución : `killall proceso`



# COMANDOS LINUX / UNIX

## bash SHELL

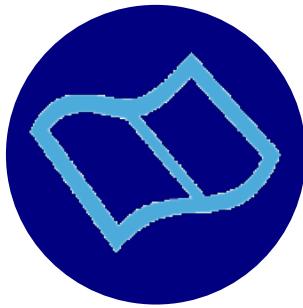
- Eliminación de procesos (3)

The screenshot shows a terminal window titled '>\_Terminal' with the status bar indicating 'dv 19:45'. The window title bar also shows 'Salvador M...'. The terminal content is as follows:

```
Activitats >_Terminal dv 19:45 Salvador M...
salva@lliurex-web: /etc
Fitxer Edita Visualitza Busca Terminal Ajuda
salva@lliurex-web:/etc$ ps aux | grep apache
root      3488  0.0  0.0   6216  2760 ?        Ss   07:06   0:07 /usr/sbin/apache2 -k s
tart
www-data  3491  0.0  0.0   5996  1924 ?        S    07:06   0:00 /usr/sbin/apache2 -k s
tart
www-data  3494  0.0  0.0 228660  2264 ?        S1   07:06   0:00 /usr/sbin/apache2 -k s
tart
www-data  3495  0.0  0.0 229732  2716 ?        S1   07:06   0:00 /usr/sbin/apache2 -k s
tart
salva    11889  0.0  0.0   4412   824 pts/0    S+   19:45   0:00 grep --color=auto apache
salva@lliurex-web:/etc$ sudo killall apache2
[sudo] password for salva:
salva@lliurex-web:/etc$ ps aux | grep apache
salva    11893  0.0  0.0   4408   824 pts/0    S+   19:45   0:00 grep --color=auto apache
salva@lliurex-web:/etc$
```

# GESTIÓN DE PROCESOS EN LINUX





# COMANDOS LINUX / UNIX

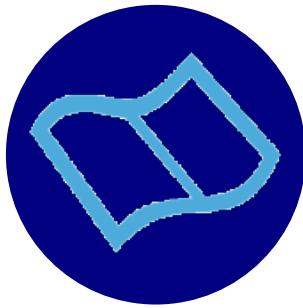
## bash SHELL

- Ejecutar procesos en segundo plano
  - Poner el símbolo & tras el comando a ejecutar
    - Ej. `gedit &` lanza el editor de texto y nos devuelve el prompt para seguir usando el terminal

```
salva@ASUS-SALVA:~$ gedit &
[1] 16857
salva@ASUS-SALVA:~$
```

- Conocer los procesos en segundo plano : `jobs` o `jobs -p`

```
salva@ASUS-SALVA:~$ jobs
[1]+  Ejecutando                  gedit &
salva@ASUS-SALVA:~$ jobs -p
16857
```



# COMANDOS LINUX / UNIX

## bash SHELL

- Ejecutar procesos en segundo plano (2)
  - Devolver un proceso a primer plano : **fg**
    - Si hay varios procesos, averiguar el número de orden mediante **jobs**, y pasarlo como parámetro a **fg**

```
salva@ASUS-SALVA:~$ jobs
[1]-  Ejecutando                      gedit &
[2]+  Ejecutando                      calibre &
salva@ASUS-SALVA:~$ fg 2
calibre
```

- Enviar un proceso de primer plano a segundo plano
  - Primer paso : Pulsar **CTRL+Z**
  - Segundo paso : Comando **bg**



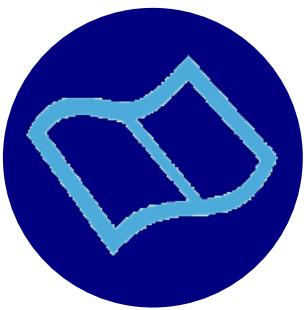
# COMANDOS LINUX / UNIX

## bash SHELL

- Ejecutar procesos en segundo plano (3)
  - Mantener un proceso en ejecución aunque se cierre la shell desde la que se lanzó : `nohup comando_a_ejecutar`

# PROCESOS EN SEGUNDO PLANO

A collage of various Linux distributions and their logos, including Arch Linux, Gentoo, Slackware, Mandriva, Mageia, SUSE, Red Hat, Ubuntu, and openSUSE.

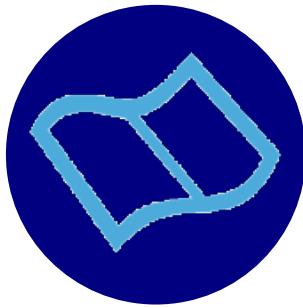


# SISTEMA DE FICHEROS

- Permisos y propiedad de los ficheros y directorios
  - Podemos ver los permisos y la propiedad con `ls -l`

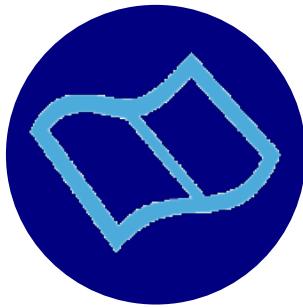
```
salva@ASUS-SALVA:~$ ls -l
total 3868
drwxrwxr-x  4 salva salva  4096 jul  2 12:16 adt-bundle-linux
drwxrwxr-x 13 salva salva  4096 ago 21 12:47 android-sdk-linux
drwxrwxr-x  7 salva salva  4096 ago 13 22:45 android-studio
drwxrwxr-x  3 salva salva  4096 ago 23 12:04 AndroidStudioProjects
drwxr-xr-x  8 salva salva  4096 ago 28 18:28 Descargas
drwxr-xr-x 12 salva salva  4096 ago 23 10:40 Documentos
drwxr-xr-x  2 salva salva  4096 ago 17 11:26 dwhelper
drwxrwxr-x 567 salva salva 20480 ago 29 18:26 ebooks
drwxrwsr-x  9 salva salva  4096 ago 26 19:06 eclipse
drwxr-xr-x  4 salva salva  4096 ago 29 19:00 Escritorio
```

- Primer recuadro: permisos
- Segundo recuadro: usuario propietario
- Tercer recuadro: grupo propietario



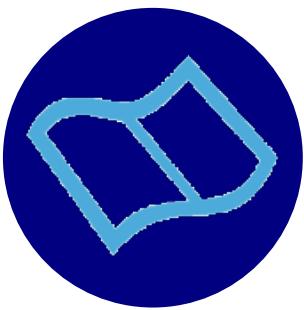
# SISTEMA DE FICHEROS

- Permisos y propiedad de los ficheros y directorios (2)
  - La columna de permisos se divide a su vez en tres columnas más una con información adicional
  - Las columnas incluyen tres permisos cada una: lectura-escritura-ejecución **rwx**
  - La primera columna (izquierda), indica permisos del usuario propietario
  - La segunda columna (centro), indica permisos del grupo propietario
  - La tercera columna (derecha), indica permisos para el resto de usuarios



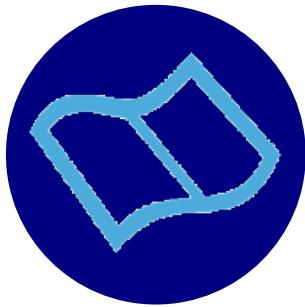
# SISTEMA DE FICHEROS

- Permisos y propiedad de los ficheros y directorios (3)
  - Cambiar usuario propietario: comando **chown**
  - Cambiar grupo propietario: comando **chgrp**
  - Podemos cambiar simultáneamente usuario y grupo con **chown *usuario:grupo fichero***
  - Para aplicar cambios de forma recursiva en directorios, modificador **-R**
  - Las operaciones anteriores requieren de una elevación de privilegios



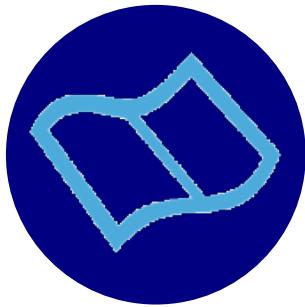
# SISTEMA DE FICHEROS

- Permisos y propiedad de los ficheros y directorios (4)
  - Cambiar permisos: comando **chmod**
  - **chmod** acepta los permisos a través de letras o a través de números en octal
    - Números en octal considerar **r=4**, **w=2**, **x=1** y sumar pesos
    - Ej. **chmod u+rwx ./salva.sh** le pone permisos de lectura, escritura y ejecución para el usuario actual (u) al fichero **salva.sh**
    - Ej. **chmod 765 ./salva.sh** le pone permisos de lectura, escritura y ejecución al usuario propietario (**7=rwx**), lectura y escritura al grupo propietario (**6=rw**) y lectura y ejecución al resto (**5=rx**)



# CONVERSIÓN BINARIO-OCTAL

| BINARIO | OCTAL | PERMISOS |
|---------|-------|----------|
| 000     | 0     | ---      |
| 001     | 1     | --x      |
| 010     | 2     | -w-      |
| 011     | 3     | -wx      |
| 100     | 4     | r--      |
| 101     | 5     | r-x      |
| 110     | 6     | rw-      |
| 111     | 7     | rwx      |

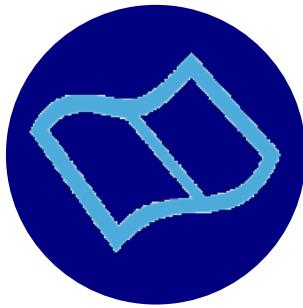


# 1r CARÁCTER DE MÁSCARA

| CARÁCTER | SIGNIFICADO                                          |
|----------|------------------------------------------------------|
| -        | Archivo                                              |
| d        | Directorio                                           |
| b        | Archivo de bloques (dispositivo)                     |
| c        | Archivo de caracteres especiales (tty, impresora...) |
| l        | Enlace simbólico                                     |
| p        | Archivo especial tubería (pipe)                      |

d rwx rw- r--

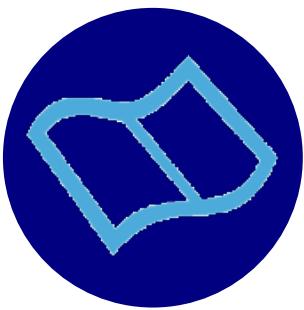
# PERMISOS Y PROPIETARIOS EN LINUX



# SISTEMA DE FICHEROS

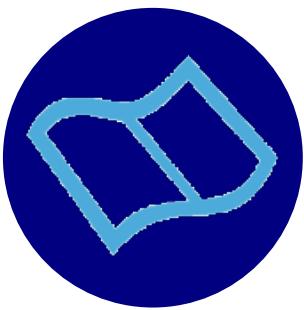
- Permisos especiales
  - **SUID**. Permite a usuarios normales elevar temporalmente el nivel de privilegios para poder escribir sobre ficheros sobre los que no tienen la propiedad
    - Significa que quien ejecute el archivo dispondrá de los mismos permisos que su usuario propietario
    - Ej. Un usuario debe ser capaz de cambiar su propia contraseña, pero ello implica modificar la base de datos de contraseñas, de la cual no tiene propiedad

```
salva@ASUS-SALVA:~$ ls -l /usr/bin/passwd
-rwsr-xr-x 1 root root 47032 feb 17 2014 /usr/bin/passwd
salva@ASUS-SALVA:~$
```



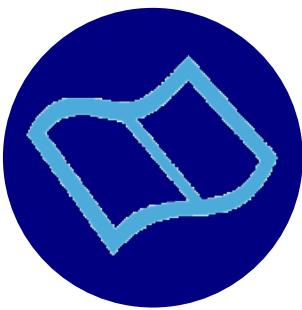
# SISTEMA DE FICHEROS

- Permisos especiales (2)
  - **SGID**. Es lo mismo que SUID, pero a nivel de grupo
    - Todo archivo que tenga activo el SGID tendrá los mismos permisos al ser ejecutado que el grupo al que pertenece
    - Se suele utilizar para crear directorios compartidos a nivel de grupo
  - **Sticky Bit**. Permite que cualquiera pueda escribir o modificar un fichero, pero se mantienen los permisos del usuario propietario (p.ej. solo él podría borrarlo)



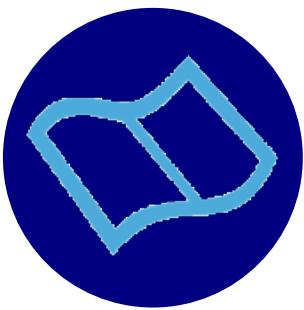
# SISTEMA DE FICHEROS

- Implementación de los permisos especiales
  - **SUID.** `chmod u+s /ruta/al/archivo`
  - **SGID.** `chmod g+s /ruta/al/directorio`
  - **Sticky bit.** `chmod o+t /ruta/al/directorio`
  - Quitar los permisos especiales : `u-s, g-s, o-t`
  - Podemos implementarlos en octal (1r número en grupo de 4 números)
    - SUID = 4, SGID = 2, Sticky = 1
    - Ej. `chmod 4765 fichero`



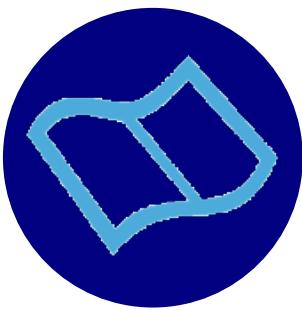
# SISTEMA DE FICHEROS

- Enlaces a ficheros y directorios
  - Los ficheros y directorios están físicamente escritos sobre bloques de disco
  - Puede haber dos o más punteros que apunten al mismo archivo o directorio
  - Se distingue entre enlace físico (duro) y enlace simbólico (blando – acceso directo)
    - Los enlaces duros sólo se pueden hacer dentro de una misma partición del sistema de ficheros
    - Los enlaces simbólicos pueden hacerse entre diferentes particiones o incluso dispositivos



# SISTEMA DE FICHEROS

- Enlaces a ficheros y directorios (2)
  - Tanto ficheros como directorios son ficheros en Linux
  - Un directorio es un fichero especial que contiene nombres de ficheros y directorios, y sus correspondientes inodos
  - Podemos consultar el inodo raíz de los ficheros o directorios mediante `ls -i`
  - Dos ficheros enlazados con un enlace físico, tendrán el mismo inodo raíz
  - Para crear enlaces, comando  
`ln origen_a_enlazar nombre_enlace`

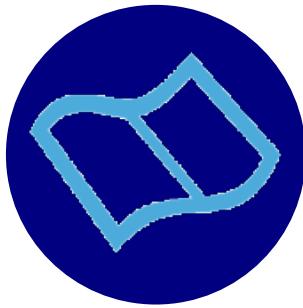


# SISTEMA DE FICHEROS

- Enlaces a ficheros y directorios (3)

```
salva@lliurex-web:~$ sudo ln /etc/samba/smb.conf .
[sudo] password for salva:
salva@lliurex-web:~$ ls -i /etc/samba/smb.conf
2361536 /etc/samba/smb.conf
salva@lliurex-web:~$ ls -i smb.conf
2361536 smb.conf
salva@lliurex-web:~$ █
```

- En la primera instrucción se ha creado un enlace duro al fichero **/etc/samba/smb.conf** en el directorio actual (tomará el mismo nombre de fichero)
- En las dos siguientes instrucciones se comprueba que el inodo raíz es el mismo en los dos sitios
- Al ser un enlace duro, si cambiamos algún atributo, como los permisos en uno de los dos sitios, el cambio se reflejará en el otro



# SISTEMA DE FICHEROS

- Enlaces a ficheros y directorios (4)
  - Al tener enlaces duros, si borramos uno de los enlaces, el otro permanece
    - El fichero no se elimina del disco hasta que no queda sin enlaces
  - Los enlaces simbólicos permiten enlazar recursos que están en diferentes volúmenes (también dentro del mismo)
  - Creación de enlaces simbólicos :  
`ln -s recurso_a_enlazar nombre_del_enlace`
  - El sistema representa un enlace mediante el símbolo `->` apuntando al recurso enlazado

```
salva@ASUS-SALVA:~$ ls -l lampp
lrwxrwxrwx 1 salva salva 11 ago 31 11:01 lampp -> /opt/lampp/
IES ESF salva@ASUS-SALVA:~$ █
```



# SISTEMA DE FICHEROS

- Enlaces a ficheros y directorios (5)
  - Ej. Creación de un enlace simbólico :  
`ln -s /opt/lampp ~/xampp`
  - El ejemplo anterior enlaza de forma simbólica el directorio **/opt/lampp** en nuestro directorio **HOME** (representado por **~**) con el nombre **xampp**
  - Si movemos de sitio el recurso original, se pierde el enlace

```
salva@ASUS-SALVA:~$ ls -l lampp
lrwxrwxrwx 1 salva salva 10 ago 31 11:13 lampp -> /opt/lampp
salva@ASUS-SALVA:~$ sudo mv /opt/lampp /opt/xampp
[sudo] password for salva:
salva@ASUS-SALVA:~$ ls -l lampp
lrwxrwxrwx 1 salva salva 10 ago 31 11:13 lampp -> /opt/lampp
salva@ASUS-SALVA:~$ █
```

# ENLACES...

# EN LINUX

# ACTIVIDAD 1.04

## INSTALACIÓN DE POWERSHELL EN DEBIAN

- Instalar **PowerShell** en **Debian 10.5**. Para ello basarse en las instrucciones que publica Microsoft en la web  
<https://docs.microsoft.com/es-es/powershell/scripting/install/installing-powershell-core-on-linux?view=powershell-7>
- Crear un nuevo usuario con nombre **windows** y la contraseña habitual del curso que utilice de forma predeterminada **PowerShell**
- Hacer uso de algunos comandos básicos de **PS** de ejemplo, tales como: **Get-help**, **Copy-Item**, **Remove-Item**
- Esta actividad debe resolverse **realizando un vídeo** en el que el/la alumno/a describa el proceso realizado