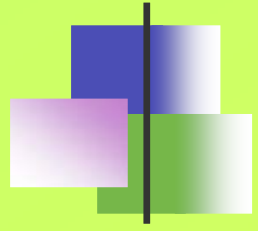


---

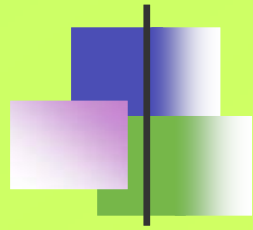
# Entorno Linux/Unix



# ¿Qué es Linux?

---

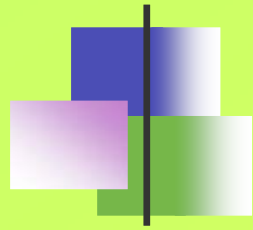
- Una versión gratuita de UNIX para PC



# Vale... ¿y qué es UNIX?

---

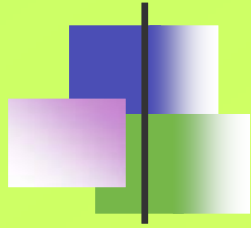
- S.O. diseñado a principios de los años 70 en los laboratorios de AT&T
- Multitarea, multiusuario
- años 70-80: se le añade memoria virtual; redes e internet; interfaces gráficas...
- estándar internacional (normas POSIX)



# Historia de UNIX

---

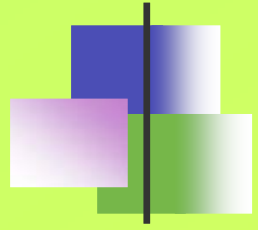
- AT&T Labs crean UNIX, que al principio se llamaba MULTICS
- A partir de ese UNIX, Berkeley Software Distribution crean BSD Unix con mejoras
- USL (Unix System Labs), filial de AT&T crea el Unix System V, antes de que Novell la adquiriera en 1993, vendiéndola más adelante con el nombre de Unix Ware. Luego Novell vende USL a SCO (Santa Cruz Operation)
- Microsoft Desarrolla Xenix, a finales de los 70, cuando comienzan los PCs. Aparece el Intel 80386 y Microsoft y AT&T fusionan Xenix y Unix, apareciendo el Unix System V/386 v.3.2 que funcionaba casi con cualquier hardware. Luego aparece la v.4.0.
- IBM hace un UNIX para sus equipos: el AIX



# GNU/Linux

---

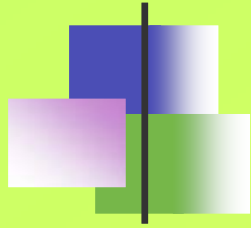
- Proyecto original: Linus Torvalds (23 años) (Universidad de Helsinki, Finlandia)
  - desarrolló un núcleo UNIX para PC
- Al núcleo se le añadieron aplicaciones de sistema desarrolladas en el proyecto GNU (compiladores, editores, GUI, etc.)
- Resultado: versión gratuita de UNIX para PC, de libre distribución y sin propietario



# Software libre

---

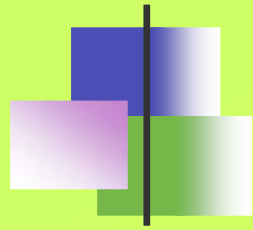
- Linux se enmarca en la corriente de *software libre*:
  - Código fuente accesible sin restricciones
  - Abierto a las modificaciones de cualquiera
  - Se puede copiar sin pagar *royalties*



# Características de Linux

---

- Soporta la mayoría de software creado para UNIX
- Existe en la actualidad mucho software, de todo tipo, libre y no libre, creado para Linux
- Permite configurar y optimizar el PC al máximo (a bajo nivel)
- Soporta diversos sistemas de ficheros : *fat16* (win95), *fat32* (win98), *ntfs* (win NT/XP), y *ext2fs* (linux)
- Soporta muchísimas arquitecturas como ISA, VESA, PCI, ..., muchas controladoras SCSI, tarjetas PCMCIA, lectoras/grabadoras CDROM y DVD, tarjetas de vídeo standards (CGA, EGA, VGA, sVGA, XGA)
- Soporta Redes TCP/IP (familia de protocolos de comunicaciones red) creado para grandes ordenadores
- Linux s Multitarea/Multiusuario
- Existen muchas **DISTRIBUCIONES**, intérpretes de comandos, y gestores de ventanas X-Windows para Linux

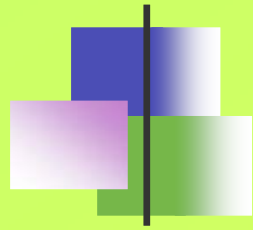


# Distribuciones de linux

---

- Empresas y organizaciones se dedican a preparar versiones instalables de Linux:
  - *Centos*, Red Hat, SuSe, Debian, Caldera, etc...
  - Valor añadido: ahorran el esfuerzo de compilación, configuración, instalación, actualización de módulos, etc...
  - Aunque linux es gratuito, las distribuciones no siempre lo son, debido a su valor añadido (programas, componentes, ...)

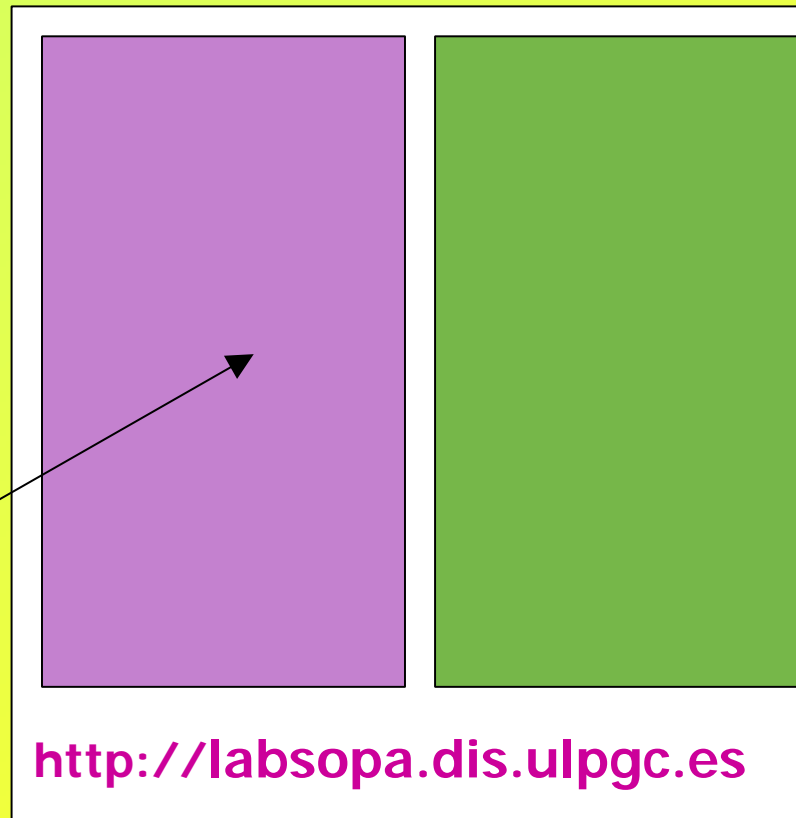


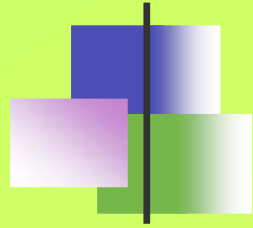


# LabSopa

---

PC's reservados  
para la asignatura

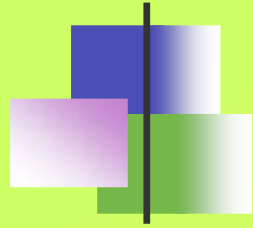




# KDE / GNOME

---

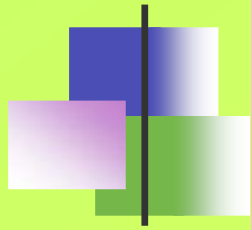
- Entorno gráfico de ventanas parecido a windows
  - Escritorio
  - Arrastrar y soltar
  - Ratón: un solo click
  - Papelera
  - Floppy
  - Cdrom, ...



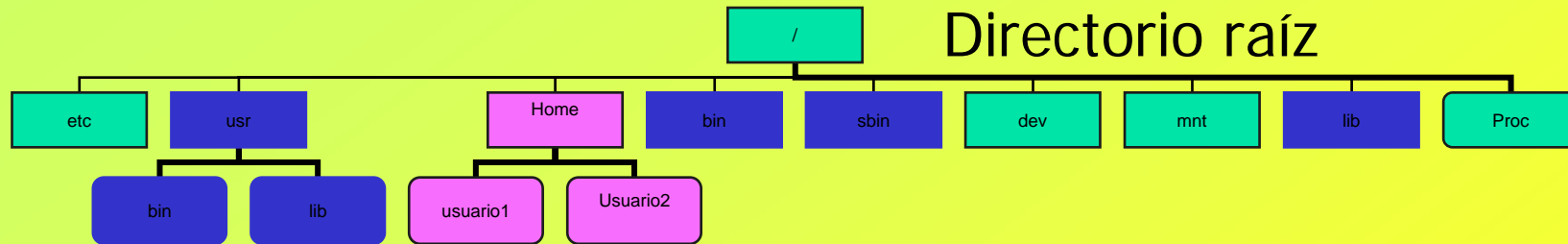
# Escritorio

---

- Panel
  - Menú principal de KDE /GNOME
    - Programas y aplicaciones (editores, etc...)
  - Barra de tareas
  - Escritorios múltiples
  - *Shell*
  - Gestor de ficheros



# Sistema de Ficheros



/etc -> Ficheros de configuración del sistema, p.ej. passwd, fstab, /etc/X11, /etc/skel

.....  
/usr -> Ficheros importantes del sistema, p.ej. /usr/man, /usr/doc, /usr/doc, /usr/lib

/home -> Directorios de usuarios

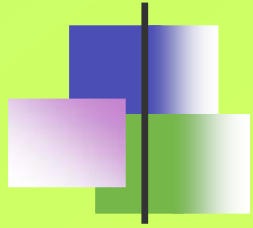
/bin, /sbin, /usr/bin -> Programas ejecutables (binarios), p.ej. comandos básicos del shell

/dev -> Archivos especiales (archivos de dispositivos)

/mnt -> dispositivos montados

/lib -> bibliotecas compartidas que utilizan los programas al ejecutarse

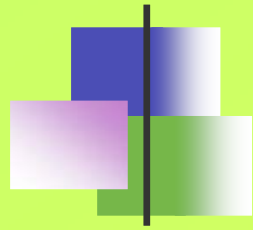
/proc -> archivos referentes a procesos en ejecución



# Árbol de directorios

---

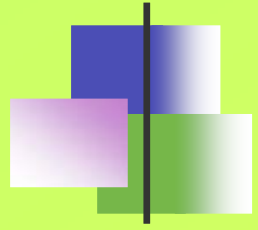
- Desde el punto de vista del usuario, este "árbol" parece una entidad uniforme (sólo se ven archivos y directorios)
- En realidad, muchos de los directorios que se ven en el árbol pueden estar en diferentes particiones, diferentes discos e incluso diferentes computadoras



# Peculiaridades en los nombres de archivos

---

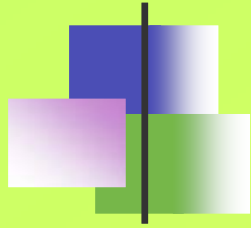
- Distingue mayúsculas
- No existe el concepto de extensión
- El separador es /
- No hay unidades: es un único árbol de directorios
- Los nombres de archivos con espacio van entre "" o bien usando espacio como \sp
- Otros caracteres especiales son: \\*, \?



# Tipos de archivos

---

- Archivos normales
- Directorios
- Enlaces
- Archivos especiales o archivos de dispositivo

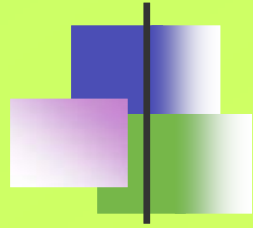


# Archivos de dispositivo

---

- Linux nos permite tratar a todos los dispositivos físicos como ficheros (abstracción)
  - */dev/mouse*            */dev/console*
  - */dev/hd*                */dev/tty*
  - */dev/fd0*               */dev/ttyS*
- Indaguemos en */dev/* y busquemos archivos especiales!

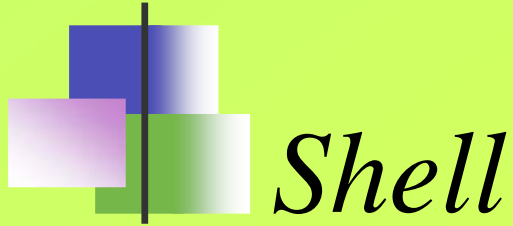




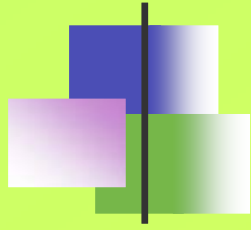
# Primera Práctica...

---

- Conociendo KDE de Linux.
- Ventanas : manejo, cierre, movimiento, ...
- Localizar distintos Árboles de directorios
- Situarnos en `/home/usuario` y crear el directorio *nombre\_alumno*
- Editor de textos **gedit**. Crear un fichero de texto que contenga 7 líneas, cada una de ellas con el nombre de los días de la semana, llamado *semana.txt*
- Crear otro fichero llamado *ordenls.txt* con la orden "ls -al"
- Familiarización con las herramientas del sistema: manejo de discos, configuración de dispositivos, búsqueda de archivos, configuración de Linux, ...
- Escritorios virtuales
- Buscar los tipos de dispositivos especiales
- Buscar el navegador de internet
- Buscar el manejador de e-mail
- Buscar Open Office 1.0



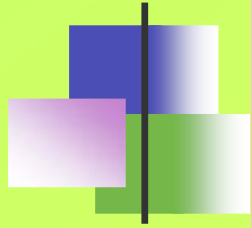
- Es un programa diseñado para aceptar órdenes y ejecutarlas (**intérprete de órdenes**)
- El equivalente en MS-DOS/Windows© sería el *command.com*



## *Shell (2)*

---

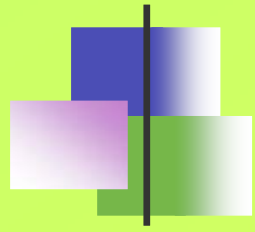
- No forma parte del núcleo (*kernel*) del SO (se ejecuta como un proceso cualquiera)
- Además existen diferentes *shells*, e incluso podemos crearnos nuestro propio *shell*
  - ♦ *sh*                      *ksh*
  - ♦ *csh*                      *bash (Red Hat)*



# Órdenes

---

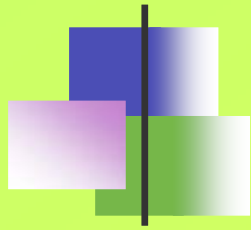
- Son sentencias escritas que el shell es capaz de ejecutar
- El shell *interpreta* órdenes y las ejecuta
- Estructura general de las órdenes:
  - **orden** *-indicadores parámetros*
  - **ls -l /etc**



¡ SOS !

---

· *man orden*



# Órdenes básicas

EJECUTAR

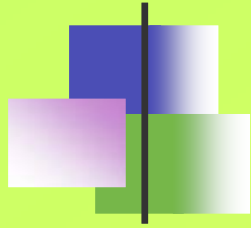
bash

- `cd` CAMBIAR DIRECTORIO
- `pwd` Imprime direc.trabajo actual
- `dir` listar contenido directorios
- `ls` listar contenido archivos
  - ◆ `-a` `-F`
  - ◆ `-l` `-R`
  - ◆ `-r` `-S`
- `clear` limpiar ventana shell
- Qué hace el comando **du**?

## ■ Expresiones regulares

- `*`, `?`, `[]`
- `\*`, `\?`

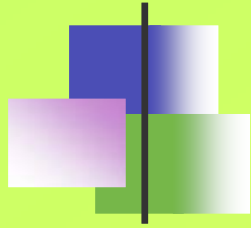
```
ls -a sal*
ls -al ../pep/pp[0..2].[a..d]
cp práctica1.f 'práctica 1.f'
cd ~
```



# Órdenes sobre ficheros

---

- `cp` Copiar archivos
- `mv` Mover o renombrar archivos
- `rm, rm -r`  
borrar archivos
- `mkdir, rmdir`  
crear directorios / borrar directorios (vacíos)
- `cat` listar contenido de un fichero de texto
- `find` Búsqueda de archivos/directorios



# Práctica 2

Realizar un fichero llamado '**práctica 1.txt**' en el que aparezcan las órdenes de comandos **estrictamente necesarios** para llevarla a cabo

- Borrar pantalla
- Situarnos en el directorio por defecto y luego dentro de *nombre\_alumno*
- Crear los directorios *fuentes* y dentro de éste, los directorios *c*, *fortran* y *cobol*
- Crear el directorio `/home/usuario/nombre_alumno/Mis Documentos`
- Hacer 3 ficheros dentro de 'Mis Documentos' que se llamen

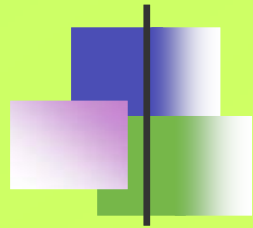
práctica1.c

newton raphson.f

conta\*1.cob

- Copiar los ficheros \*.c a c
- Mover los ficheros \*.f a fortran
- Borrar el contenido de 'Mis Documentos'
- Borrar el directorio 'Mis Documentos'
- Realizar una copia del directorio *fuentes* al directorio *Copia de Fuentes*





# Permisos

- Lectura, escritura, ejecución

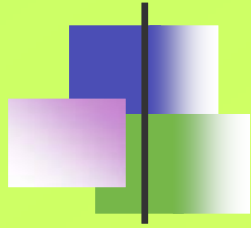
```
$ ls -al
D rwx - - - - - usuario grupo nombre archivo
- rwx rw- r-- fsantana profesor nombre archivo
```

Directorio

Propietario

Grupo

Otros



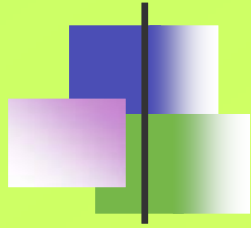
# Cambio de permisos archivos

---

- `chmod ugoa +/-/= rwx fich`

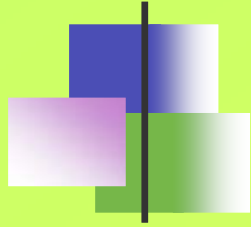
`u = propietario`    `g = grupo`    `o = otros`  
`a = todos`

```
chmod a+rwx semana.txt
chmod go-wx ordenls.txt
chmod a=r semana2.f
chmod 644 ordenls.txt
```



# Creación de *scripts* ejecutables

- Los ficheros han de tener (procesamiento por latex) **permiso de ejecución (+X)** según corresponda
- Para ejecutarlos
  - *./nombre\_script* si no está en la ruta del PATH
  - *bash nombre\_script*
- Para pasar parámetros: *\$1 \$2 ... \$n*
- *\$#* indica el número de parámetros pasados
- *\$@* indica todos los parámetros pasados separados con blancos



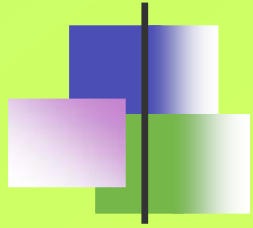
# Creación de *scripts* ejecutables

- **read** VARIABLE : (procesamiento por lotes) II introducir variable teclado  
`read var1 var2`

si introducimos : 1 2 3, al final var1 = "1", y var2 = "2 3"

- **shift** *n*: desplazamiento de *n* posiciones en las variables pasadas como argumentos. El número máximo de variables : \$1 \$2 .. \$9

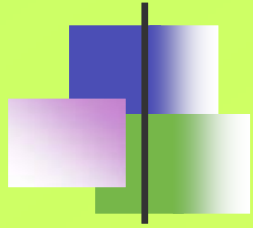
`shift 2`, producirá dos desplazamientos a la izquierda, quedando \$3 \$4 .. \$9 (desaparece \$2 y \$1)



# Práctica

---

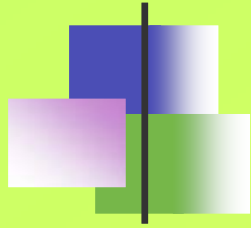
- Realizar un *script ejecutable* que permita borrar pantalla, y copiar el contenido de un fichero pasado como parámetro 1, en otro pasado como parámetro 2. Solicitar confirmación. Deberá presentarse por pantalla : **copiando *fich1* → *fich2***
- La sintaxis del *script* será:  
**copiafich *fich1* *fich2***



# Configuración del entorno *shell*

---

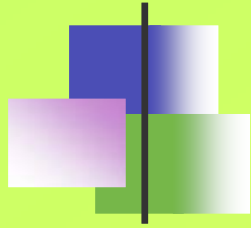
- Consiste en un conjunto de variables y valores que permiten a un programa en ejecución, como un *shell*, determinar el aspecto que tiene el entorno
  - nombre del *shell* (variable SHELL)
  - directorio de usuario (variable HOME)
  - ruta (variable PATH)



# Entorno *shell*

---

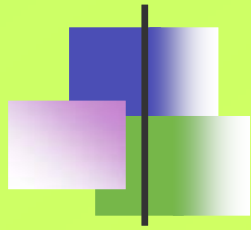
- Muchas de estas variables se definen durante el proceso de entrada al sistema (ficheros de configuración)
  - `.bash_profile`
- Valores de las variables de entorno
  - `env`
- Para variar el valor de alguna variable de forma temporal:
  - `PATH=$PATH:newpath ó`
  - `set PATH=$PATH:newpath`



# Entrada / Salida - Redirección

- Muchos comandos linux toman la entrada de datos como la entrada estándar (*stdin*) y la salida de datos como la salida estándar (*stdout*), que por defecto es el teclado (*stdin*) y la consola (*stdout*)
- La **entrada/salida** puede ser redireccionada mediante :
  - Salida : > y >> (creación/anexo archivos)
  - Entrada: <
  - Pipes : | (la salida de un proceso es la entrada para otro)

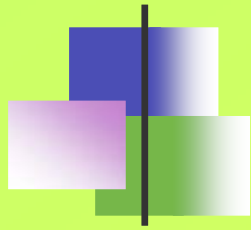




# Visualización de ficheros

---

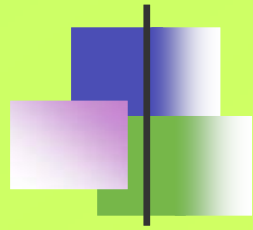
- **Cat** *visualiza contenido archivos*  
`cat nombre_fichero`  
`cat fich1 fich2`  
`cat fich1 fich2 > fich3`  
`cat fich1 >> fich2`  
`cat < fich1`  
`cat semana* | sort | more`
- **Head** *visualiza las -n primeras líneas de un archivo*  
`head -5 semana.txt`
- **Tail** *visualiza las -n últimas líneas de un archivo, ó de la línea +n en adelante*  
`Tail -5 semana.txt`  
`Tail +5 semana.txt`
- **More, Less:** *permiten paginar la ventana en pantallas de 24 líneas*  
*more no permite movernos como un editor, less sí*  
`ls -al /etc | more`



# Búsquedas

---

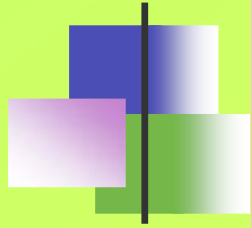
- de archivos: **find**  
**find -name "nombre" / -print**  
**find / -name fich1 -print**  
**find / -name fich1 -name fich2 -print**  
**find / (-name fich1 -o -name fich2) -print**
- de cadenas dentro de archivos: **grep**  
**grep texto-a-buscar nombre-fichero**



# Crear, montar, desmontar, chequear un sistema de ficheros

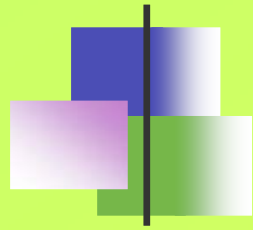
---

- **Creación** : `mkfs -t tipo_sf sf`  
`mkfs -t ext2 /dev/hda1`
- **Montaje** : `mount -t tipo_sf sf punto_montaje`  
`mount -t ext2 /dev/fd0 /mnt/floppy`  
`mount -t msdos /dev/hda1 /mnt/windowsxp`
- **Desmontaje**: `umount punto_montaje/dev`  
`umount /mnt/windowsxp`  
`umount /mnt/floppy`  
`umount /mnt/cdrom`
- **chequeo**: `fsck -t sist. Archivos dispositivo`  
`fsck -t ext2fs /dev/hda5`



# Práctica

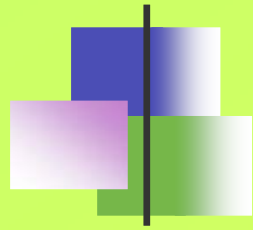
- Crear un sistema de archivos del tipo *ext2fs* en la unidad de floppy. Copiar íntegro el directorio con nuestro nombre de usuario al floppy. Luego, deberá desmontarse. A continuación, volveremos a montarla. Luego, chequearemos el sistema de ficheros del floppy y por último comprobaremos su contenido
- **NOTA: ANTES DE EXTRAER UNA UNIDAD REMOVIBLE, TENEMOS QUE DESMONTAR SU SISTEMA DE FICHEROS PARA EVITAR LA PÉRDIDA DE DATOS**



# Acceso a discos con formato MS-DOS

---

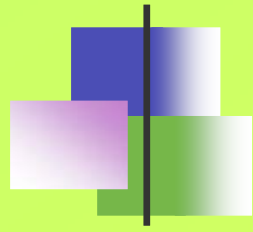
- mtools
  - mcd
  - mcopy
  - mdir
  - mformat
  - mdel
  - mmd
  - mrd



# Procesos

---

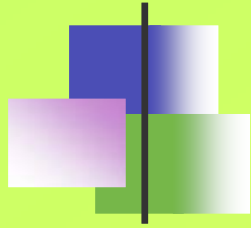
- Un programa en ejecución en linux recibe el nombre de *proceso*
- Para distinguir entre procesos, Linux asigna a cada nuevo proceso un identificador exclusivo llamado *identificador de proceso (PID)*



# Procesos (2)

---

- Ver los procesos en ejecución
  - **ps**
- “Matar” (eliminar de memoria) un proceso
  - **kill** PID
  - **kill -9** PID
- Lanzar un programa en segundo plano
  - *programa &*



# Otras órdenes

## ■ **finger [-s] *usuario***

*permite visualizar información de un usuario*

## ■ **sort *fichero***

*clasifica el contenido de un fichero*

## ■ **passwd**

*cambio de contraseña de usuario*

## ■ **who**

*muestra información sobre los usuarios que están conectado al sistema y proporciona información sobre ellos*

## ■ **logname**

*muestra el nombre de usuario con el que hemos entrado al sistema*

## ■ **date**

*muestra fecha sistema*

## ■ **ln [-s] *origen destino***

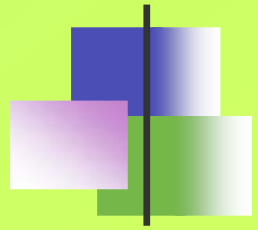
*crea enlaces simbólicos*

## ■ **df**

*visualiza ocupación file system*

■ ...





# Empaquetar, Comprimir/descomprimir

- Podemos empaquetar una estructura de directorios y ficheros en un único fichero para su posterior manejo/distribución.

**tar** [t|c|x] v [f archivo\_tar] *ficheros*

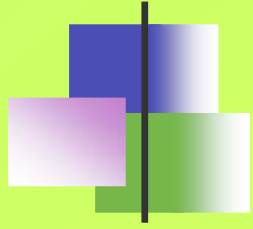
*t : visualiza el contenido del fichero\_tar*

*c : añade archivos al archivo\_tar*

*x : extrae los archivos de archivo\_tar*

## Ejemplos:

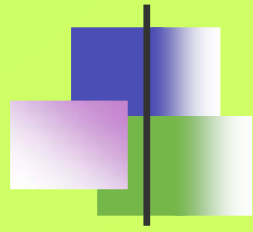
```
➔ tar -cvf /dev/fd0 /home/aquesada  
tar -cf /home/backup.tar /home/aquesada  
tar -xv /home/backup.tar
```



# Compresión/descompresión de archivos

---

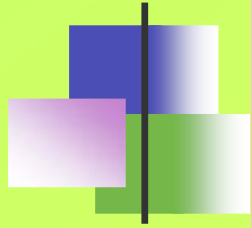
- **Tar** no comprime automáticamente.
- **gzip** : comprime archivos  
`gzip fichero`
- **gunzip** : descomprime archivos  
`gunzip fichero.gz`
- gzip/gunzip suelen trabajar conjuntamente con tar:  
`tar cvf mis_archivos.tar | gzip`



# Imprimir

---

- Ordenes
  - **lpr** -> manda el trabajo a la cola de impresión
  - **lpq** -> muestra la cola de impresión
  - **lprm** -> elimina un trabajo de la cola de impresión



# Práctica

---

- Empaquetar el contenido del directorio *nombre\_alumno* y que se llame *mis\_archivos.tar*  
Luego analizar el contenido del fichero empaquetado.
- Comprimir el fichero *.tar*
- Pasar el fichero comprimido al floppy
- ¿cómo podríamos empaquetar directamente en el floppy sin pasarlo a un fichero tar?