

---

# Unidad 7: Aspectos Avanzados

**V Curso de Introducción al Sistema Operativo  
GNU/Linux  
Extensión Universitaria. Universidad de Sevilla  
Noviembre 2007**

# Contenidos

---

- Actualización de la distribución
- La interfaz de comandos
- Otros aspectos avanzados

# Actualización distribución

---

- Modificar repositorios de paquetes:
  - Con synaptic
  - Editando `/etc/apt/sources.list`
- Recargar listas de paquetes
  - Con synaptic
  - `# apt-get update`
- Actualizar
  - Con synaptic: marcar actualizaciones + Aplicar
  - `# apt-get dist-upgrade`

# Actualización distribución

---

- ¡Leer “release-notes”! con información sobre la nueva versión
  - Instalar nuevos paquetes
  - Borrar paquetes obsoletos
  - etc.

# La interfaz de comandos

---

- Terminales de texto y virtuales
- Formato de la línea de comandos
- Comandos Unix
- Redirección y tuberías
- Aplicaciones de terminal

# Terminales de texto

---

- Linux incluye una interfaz de comandos muy avanzada
  - Facilita muchas tareas
  - Requiere un mayor aprendizaje
- Cada usuario puede interaccionar con el sistema a través de un terminal de texto o consola
- Para cambiar de consola:
  - CTRL-ALT-F1.....CTRL-ALT-F8
- Cada una de las consolas ofrece el prompt de entrada para poder acceder al sistema

# Terminales virtuales

---

- Aplicación que emula un terminal
  - Integran la interfaz gráfica con la interfaz de texto
  - Acceder a un shell de UNIX (programa que interpreta y ejecuta los comandos que el usuario escribe)
  - Ejecutar aplicaciones de terminal
- Tipos de shells:
  - bash: *Bourne again shell*. (/bin/bash)
  - csh: *C shell*
  - Otros: sh, tcsh, ksh, ...

# Formato de la línea de comandos

---

- Comando: fichero ejecutable
- Formato:
  - <comando> [OPCIONES] [ARGUMENTOS]
- [OPCIONES] pueden ser:
  - Letras precedidas del símbolo '-'
  - Palabras precedidas de los símbolos '--'
- [ARGUMENTOS] pueden ser:
  - Nombres de ficheros y directorios, etc.



# Formato de la línea de comandos

---

- Caracteres especiales o comodines: \* y ?
- \* representa cualquier cadena de caracteres
- ? representa cualquier carácter simple

# Formato de la línea de comandos

---

- Ejemplos:

```
$ ls  
cap1 cap2 cap3 suma.c resta.c
```

```
$ ls c*  
cap1 cap2 cap3
```

```
$ ls cap?  
cap1 cap2 cap3
```

```
$ ls *c  
suma.c resta.c
```

# Comandos Unix

---

- Gran cantidad de comandos que permiten realizar entre otras cosas:
  - Operaciones básicas con archivos y directorios: ls, mv, cp, cd, mkdir, rm,...
  - Operaciones avanzadas con archivos y directorios: tar, chmod, gzip,...

# Comandos Unix. Ayuda

---

- man: sistema de ayuda online.

```
$ man <comando>
```

```
$ man passwd
```

```
$ man 5 passwd
```

```
$ whatis passwd
```

- <comando> --help
- apropos <término>: comandos relacionados con <término>

# Comandos básicos

---

- Al comenzar una sesión, nos encontramos en nuestra carpeta home (/home/usuario o ~)
  - pwd (print work directory): devuelve la posición actual absoluta dentro del árbol de directorio

```
$ pwd  
/home/usuario
```

# Comandos básicos

---

- Listar ficheros, ver ficheros
  - ls (list): permite ver el contenido del directorio actual (-a: lista ficheros ocultos, -l: ofrece información adicional)

```
$ ls -la
```

# Comandos básicos

---

- Ver contenido de un ficheros
  - cat (catch): permite ver el contenido de un fichero

```
$ cat fichero.txt
```

- more: vemos el contenido de un fichero página a página

```
$ more fichero.txt
```

- less: variante comando more

```
$ less fichero.txt
```

# Comandos básicos

---

- Moviéndonos entre directorios
  - cd (change directory).

```
$ cd directorio_destino
```

```
$ cd .
```

```
$ cd ..
```

```
$ cd
```



# Comandos básicos

---

- Copiar y mover ficheros y directorios
  - cp (copy): copia ficheros y directorios.

```
$ cp fichero1 fichero2
```

```
$ cp -r directorio1 directorio2
```

- mv (move):

```
$ mv fichero1 fichero2
```

```
$ mv fichero1 directorio1
```

# Comandos básicos

---

- Borrar y crear

- rm (remove): borra ficheros y directorios

```
$ rm fichero
```

```
$ rm -r directorio
```

```
$ rmdir directorio
```

- mkdir (make directory): crea un subdirectorio

```
$ mkdir subdirectorio
```

# Comandos avanzados

---

- Propiedades de los archivos

```
$ ls -l
```

- Este comando ofrece mucha información:
  - Tipo de los archivos
  - Permisos del propietario, del grupo y de otros.
  - Propietario y grupo
  - Fecha de creación
  - ...

# Comandos avanzados

---

- Cambiando las propiedades
  - chown (change owner): cambia el propietario
  - chgrp (change group): cambia el grupo
  - chmod (change mode): cambia el tipo de acceso a un fichero/directorio

```
$ chmod a+rwx fichero
```

```
$ chmod a-rwx fichero
```

```
$ chmod u+r fichero
```

```
$ chmod go-x fichero
```

```
$ chmod 777 fichero
```

# Comandos avanzados

---

- Compresión de archivos
  - gzip: comprime un fichero (fichero resultante con extensión .gz) y borra el fichero original no comprimido

```
$ gzip fichero
```

```
$ gzip -l fichero.gz
```

- gunzip: descomprime el fichero o ficheros

```
$ gunzip fichero.gz
```

# Comandos avanzados

---

- Compresión de archivos
  - tar (tape archive): empaqueta un conjunto de ficheros manteniendo la información de propiedades y permisos de cada uno

```
$ tar cvf fich_dest fichs_orig
```

(c: crea un nuevo archivo)

```
$ tar xvf fich_dest.tar
```

(x: extraer ficheros de un archivo)

# Comandos avanzados

---

- Comprensión de archivos
  - Usando tar con gzip: concatena los dos tipos de compresión

```
$ tar cvzf fich.tar.gz fichs_orig
```

```
$ tar xvzf fichero.tar.gz
```

# Redirección y tuberías

---

- `stdin` (standard input) es la entrada estándar
- `stdout` (standard output) es la salida estándar
  - Muchos comandos toman su entrada de `stdin` y mandan su salida a `stdout`
  - El terminal virtual toma como `stdin` el teclado y como `stdout` la pantalla



# Redirección y tuberías

---

- Ejemplos:

```
$ cat prueba.txt
```

- Este comando lee datos de prueba.txt y los manda a la pantalla (stdout)

```
$ cat
```

- Lee datos del teclado (stdin) y los envía a la pantalla (stdout)

# Redirección y tuberías

---

- Redireccionamiento de la salida
  - Nos permite enviar la salida a un fichero en lugar de a la pantalla

```
$ ls > listado.txt
```

```
$ cat > lista_compra.txt  
galletas  
naranjas  
lentejas  
Ctrl-D
```

# Redirección y tuberías

---

- Redireccionamiento de la entrada
  - Nos permite tomar la entrada de un fichero en lugar de la del teclado

```
$ cat < lista_compra.txt  
galletas  
naranjas  
lentejas
```

# Redirección y tuberías

---

- Redireccionamiento entre comandos (pipes o tuberías)

- Nos permite enviar la salida de un comando como entrada de otro comando

```
$ ls | sort -r
```

```
$ ls /home/usuario | less
```

- Podemos conectar más de dos comandos

```
$ ls /home/usuario | sort -r | less
```

```
$ cat lista.txt | gzip > lista.txt.gz
```

# Redirección y tuberías

---

- Algunas consideraciones:
  - Redireccionar la salida de un comando a un fichero es destructivo: destruye el anterior contenido
  - Podemos usar “>>” para direccionar

```
$ ls >> lista_ficheros
```

- Añade la salida al final de lista\_ficheros

# Redirección y tuberías

---

- Más ejemplos

```
$ ls -R mp3 | mail -s "Mi coleccion" pepe@us.es
```

```
$ ls /*bin /usr/*bin | wc -l
```

# Otros comandos

---

- grep
- diff
- date
- cal
- bc -l

# Aplicaciones de terminal

---

- Editores de texto
  - Editor 'vi':
    - Uso muy extendido en sistema UNIX/Linux
    - Tiene varios modos: modo comando y modo de inserción y modo de línea
    - Si no se conoce es difícil de utilizar
    - Es bastante potente
  - Editor 'nano':
    - Más fácil de utilizar



# Aplicaciones de terminal

---

- Midnight Commander
  - Es un gestor de archivos en modo texto que ofrece muchas posibilidades
    - Edición de archivos
    - Copiar, borrar, crear ficheros y directorios, ect.
    - Acceder a ficheros de un sistema remoto (servidores FTP)
  - Para ejecutarlo:
    - `$ mc`

# Aplicaciones de terminal

---

- Lynx:
  - Es un navegador Web en modo texto
  - Para ejecutarlo:
    - `$ lynx`

# Otros aspectos avanzados

---

- Instalación manual de paquetes
- Mensajes del sistema
- Carga manual de módulos del kernel
- Gestor de arranque
- Recuperación del gestor de arranque
- Recuperación del sistema sin clave
- Crear y usar imágenes de discos
- Archivos interesantes

# Instalación manual de paquetes

---

- Comando dpkg
- Instalación:

```
# dpkg -i paquete.deb
```

- Desinstalación:

```
# dpkg -r nombre_paquete
```

# Mensajes del sistema

---

- Comando dmesg: muestra últimos mensajes del kernel: dispositivos detectados durante el inicio, información de los drivers, etc.
- Ficheros de log:
  - /var/log/syslog: mensajes generales
  - /var/log/kern.log: mensajes kernel
  - etc.

```
$ tail -f /var/log/syslog  
$ tail -f /var/log/kern.log
```

# Carga manual de módulos

---

- Sólo necesario en casos especiales
- Comandos:
  - lsmod: listas módulos cargados
  - rmmod: descargar un módulo
  - modprobe: cargar un módulo
- Configuración
  - /etc/modules: lista de módulos a cargar en el inicio del sistema

# Gestor de arranque

---

- Gestor arranque Ubuntu: GRUB
- Permite cargar distintos sistemas operativos:
  - Linux
  - Windows
- Configuración
  - `/boot/grub/menu.lst`
  - `# update-grub`
- Instalación
  - `# grub-install hd0`

# Recuperación del gestor de arranque

---

- Cuando instalamos Windows necesitamos restaurar el gestor de arranque
  - Introducimos CD instalación Ubuntu
  - boot: *rescue*
  - Seguir los pasos indicados:
    - Montar partición con linux
    - Ejecutar el comando

```
# grub-install hd0
```



# Recuperación del sistema sin clave

---

- Hemos olvidado nuestra clave: ¿Qué podemos hacer?
- Necesitamos abrir un terminal de root:
  - Desde el menú de GRUB seleccionar la opción *recovery mode*
  - A continuación, como root

```
# passwd usuario
```

# Crear y usar imágenes de discos

---

- Ficheros de dispositivos:
  - Disquete: /dev/fd0
  - CD-ROM: /dev/cdrom
- Para crear imágenes de discos (exactamente iguales):

```
$ cat /dev/cdrom > imagen-cd.iso  
$ cat /dev/fd0 > imagen-floppy.img
```

- Copia todos los bloques del disco

# Crear y usar imágenes de discos

---

- Para usar la imagen:

```
# mount imagen-cd.iso /mnt -t iso9660 -o loop
# mount imagen-floppy.img /mnt -t vfat -o loop

# cp nombre_archivo /mnt

# cat imagen-floppy > /dev/fd0
```

- Para dejar de usar la imagen:

```
# umount /mnt
# umount /mnt
```

# Archivos interesantes

---

- En /proc: carpeta virtual, muestra información configuración, de los procesos, dispositivos en forma de archivos.
  - cpuinfo: información cpu
  - meminfo: información memoria
  - ioprots: información puertos de los dispositivos E/S
  - interrupts: número interrupciones de los dispositivos

# Archivos interesantes

---

- Ejemplos:

```
$ less /proc/cpuinfo  
$ less /proc/meminfo  
$ less /proc/ioports  
$ less /proc/interrupts
```

```
$ watch -n 0 cat /proc/interrupts
```