

WUOLAH



pdgiim

www.wuolah.com/student/pdgiim



1478

sesion3-resumen.pdf

Practicas Módulo1



2º Sistemas Operativos



Grado en Ingeniería Informática



Escuela Técnica Superior de Ingenierías Informática y de Telecomunicación
Universidad de Granada



Descarga la APP de Wuolah.

Ya disponible para el móvil y la tablet.





**KEEP
CALM
AND
ESTUDIA
UN POQUITO**

SESIÓN 3

Cuando un administrador de un sistema detecta un problema de rendimiento debe seguir los siguientes pasos:

12. Definir el problema lo más detalladamente posible.
13. Determinar la causa o causas del problema, para ello se utilizará alguna de las herramientas que veremos en esta sesión.
14. Pensar una solución para mejorar el rendimiento del sistema.
15. Llevar a cabo dicha solución, diseñando e implementando las modificaciones al sistema necesarias.
16. Monitorizar el sistema para determinar si los cambios realizados han sido efectivos

CPU

uptime → Muestra una línea con la siguiente información: la hora actual, el tiempo que lleva en marcha el sistema, el número de usuarios conectados, y la carga media del sistema en los últimos 1, 5 y 15 minutos (número de procesos en la cola de ejecución del núcleo. Un sistema con un comportamiento normal debe mostrar una carga igual o inferior a 1, aunque se deben tener en cuenta la configuración y el tipo de programas en ejecución. En sistemas multi-core habría que multiplicar este número por el número de cores)

- p, --pretty
show uptime in pretty format
- h, --help
display this help text
- s, --since
system up since, in yyyy-mm-dd HH:MM:SS format
- V, --version
display version information and exit

w <user> → Misma información pero también muestra los usuarios conectados y lo que están haciendo(sus procesos).

Time → Mide el tiempo de ejecución de un programa y muestra un resumen del uso de los recursos del sistema. Muestra el tiempo total que el programa ha estado ejecutándose (real), el tiempo que se ha ejecutado en modo usuario (user) y el tiempo que se ha ejecutado en modo supervisor (sys), El tiempo de espera de un proceso es por tanto:

$$T_{espera} = real - user - sys$$

Ej: ejecutar time con el programa ps (time ps). Primero se ejecuta ps y después nos muestra los valores de los distintos tiempos: real, user y sys.

Opciones:

Con -o <file> escribe los resultados en el archivo file (lo sobrescribe) en lugar de en la salida estándar de error. Acompañado de -a no sobrescribe el archivo.

Linux realiza una planificación por prioridades. Por defecto, un proceso hereda la prioridad de su

proceso padre.

Orden nice → Permite establecer el valor de prioridad de un proceso (se le pasa el comando como argumento) a un valor distinto del que tendría por defecto. El rango de valores que permite establecer como parámetro es [-20,19]. Asignar un valor negativo aumenta la posibilidad de ejecución de un proceso (la prioridad) y solamente puede hacerlo el usuario root. Los usuarios sin privilegios de administración solo pueden utilizar los valores positivos de este rango, disminuyendo así la prioridad.

-n <valor de prioridad> (no hace falta ponerlo, con poner el valor de prioridad que queremos dar es suficiente)

Renice → permite alterar el valor de prioridad de uno o más procesos en ejecución.

-n <valor de prioridad> (no hace falta ponerlo, con poner el valor de prioridad que queremos dar es suficiente)

-g <ID del grupo de procesos> cambia la prioridad de todos los procesos pertenecientes al grupo de procesos cuyo ID se pasa como argumento

-p <PID> cambia la prioridad de todos los procesos cuyo PID se pasa como argumento (es la opción por defecto)

-u <user(nombre o UID) > cambia la prioridad de todos los procesos pertenecientes al usuario que se pasa como argumento

Pstree → Visualiza un árbol de procesos en ejecución. Opciones más importantes:

-a Muestra los argumentos de la línea de órdenes.

-A Usa caracteres ASCII para dibujar el árbol.

-G Usa los caracteres de VT100.

-h Resaltar el proceso actual y sus antepasados.

-H Igual que -h, pero para el proceso que se especifique.

-l Usa un formato largo, por defecto las líneas se truncan.

-n Ordena los procesos por el PID de su antecesor en vez de por el nombre (ordenación numérica)

-p Desactiva el mostrar los PIDs entre paréntesis después del nombre de cada proceso.

-u Si el uid de un proceso difiere del uid de su padre, el nuevo uid se pone entre paréntesis

Para más opciones ver man (no hay muchas más, las más importantes ya están aquí).

Orden ps → Muestra información sobre los procesos en ejecución.

Esta orden se implementa usando el pseudo-sistema de archivos /proc (contiene una jerarquía de archivos (archivos virtuales) que nos muestran el estado actual del kernel. Esta es la forma en que las aplicaciones y usuarios pueden tener una vista actualizada del estado del sistema. El /proc se puede considerar un pseudo-filesystem, pues no es un sistema de archivos real y no consume almacenamiento, solo memoria. Dentro del /proc de Linux encontraremos información acerca de cada proceso que se está ejecutando en el sistema así también como detalles del hardware del sistema.)

USER : usuario que lanzó el programa

PID : identificador del proceso

PPID: ID del padre del proceso

UID: usuario propietario del proceso

CMD: nombre del programa o comando que inició el proceso

TIME: tiempo de uso de cpu acumulado por el proceso

STIME: hora de inicio del proceso

%CPU : porcentaje de cpu entre el tiempo usado realmente y el que lleva en ejecución

%MEM : fracción de memoria consumida (es una estimación)
 VSZ (SIZE): tamaño virtual del proceso (código+datos+pila) en KB
 RSS : memoria real usada en KB
 TTY : terminal asociado con el proceso
 PRI : Prioridad del proceso.
 NI : Valor nice estándar de Unix.
 STAT : estado del proceso que puede ser:
 R: en ejecución o listo (Running o Ready)
 S: durmiendo (Sleeping)
 T: parado (sTopped)
 Z: proceso Zombie (Proceso que ha completado su ejecución pero aún tiene una entrada en la tabla de procesos, permitiendo al proceso que lo ha creado leer el estado de su salida. La presencia de muchos procesos zombi puede indicar problemas en el sistema, y puede acarrear una alta carga del sistema, lentitud y respuestas lentas)
 D: durmiendo ininterrumpible (normalmente E/S)
 N: prioridad baja (> 0)
 <: prioridad alta (< 0)
 s: líder de sesión
 l: tiene multi-thread
 +: proceso foreground (en primer plano)
 L: páginas bloqueadas en memoria

La orden ps normalmente se ejecuta con las operaciones -ef,
 e/A se seleccionan todos los procesos que estén en el sistema
 f se muestra información completa
 l se imprime en pantalla la información en formato largo
 u <nombre> Muestra el estado de los procesos del usuario <nombre>
 o <formato> Permite usar un formato definido por el usuario para el listado. Se indican los campos que deseamos ver separados por coma.
 a muestra también los procesos de otros usuarios.

N niega el efecto de cualquier opción que se haya especificado.

ps -u root -N visualiza todos los procesos que no sean del usuario root

x muestra procesos que no están controlados por ninguna terminal.

h sin cabecera (header)

r sólo procesos que se están ejecutando

Sin argumentos muestra los procesos lanzados por el usuario que ejecuta esta orden.

Es posible obtener la información sobre el estado de procesos concretos dando sus identificadores como argumento.

```
#ps axo user,stat,pid,pcpu,comm
```

USER	STAT	PID	%CPU	COMMAND
root	Ss	1	0.0	systemd
root	S	2	0.0	kthreadd
root	S	3	0.0	ksoftirqd/0
root	S<	5	0.0	kworker/0:0H
root	S	7	0.0	migration/0
root	S	8	0.0	rcuc/0



Descarga la APP de Wuolah.

Ya disponible para el móvil y la tablet.



```
root S 9 0.0 rcub/0
```

```
#ps axo user,pid,comm
```

```
USER      PID COMMAND
```

```
root      1  systemd
```

```
root      2  kthreadd
```

```
root      3  ksoftirqd/0
```

```
root      5  kworker/0:0H
```

```
l          formato largo
```

```
u          formato usuario: muestra el usuario y la hora de inicio
```

```
j          formato trabajo (jobs): pgid sid
```

```
s          formato señal (signal)
```

```
v          formato vm
```

```
m          muestra información de memoria (combínese con p para obtener el número de páginas).
```

```
f          formato "forest" ("bosque") de familias en forma de árbol
```

```
a          muestra también los procesos de otros usuarios
```

```
x          muestra procesos que no están controlados por ninguna terminal
```

```
S          añade tiempo de CPU y fallos de página de los hijos
```

```
c          nombre del comando obtenido de task_struct
```

```
e          muestra ambiente (environment) después del nombre del comando y '+'
```

```
w          Salida ancha (wide): no se truncan las líneas de comando para que quepan en una línea. Para ser exactos, cada w que se especifica añadirá una posible línea a la salida. Si el espacio no se necesita, no se usa. Puede usar hasta 100 w's.
```

```
h          sin cabecera (header)
```

```
r          sólo procesos que se están ejecutando
```

```
n          salida numérica para USER y WCHAN.
```

```
txx       sólo procesos controlados por la tty xx; para xx debe usar bien el nombre de un dispositivo bajo "/dev" o bien ese nombre sin las letras tty o cu que lo preceden. Esta es la operación inversa que ps utiliza para imprimir el nombre abreviado de tty en el campo TT, por ejemplo, ps -t1.
```

```
O[+|-]k1[, [+|-]k2[, ...]]
```

Ordena la lista de procesos de acuerdo con el ordenamiento multi-nivel especificado por la secuencia de claves ordenación de CLAVES DE ORDENACIÓN, k1, k2, Existen especificaciones de ordenación por defecto para cada uno de los formatos de ps. Éstas pueden ser anuladas por una ordenación especificada por el usuario. El '+' es opcional, meramente para reiterar la dirección por defecto de una clave. '-' invierte la dirección sólo de la clave que precede. Como con t y pids, la opción O debe ser la última opción en un argumento, aunque las especificaciones en argumentos posteriores son concatenadas.

```
pids      Lista sólo los procesos especificados; están delimitados por comas. La lista se debe dar inmediatamente después de la última opción en un argumento simple, sin intervención de espacios, por ejemplo ps -j1,4,5. Las listas especificadas en los argumentos siguientes son concatenadas, por ejemplo ps -l 1,2 3,4 5 6 listará todos los procesos del 1 al 6 en formaTo see every process on the system to largo. Los pids se listan incluso si contradicen a las opciones 'a' y 'x'
```

To see every process on the system using standard syntax:

```
ps -e
```

```
ps -ef
```

ps -eF

ps -ely

To see every process on the system using BSD syntax:

ps ax

ps axu

To print a process tree:

ps -ejH

ps axjf

To get info about threads:

ps -eLf

ps axms

To get security info:

ps -eo euser,ruser,suser,fuser,f,comm,label

ps axZ

ps -eM

To see every process running as root (real & effective ID) in user format:

ps -U root -u root u

To see every process with a user-defined format:

ps -eo pid,tid,class,rtprio,ni,pri,psr,pcpu,stat,wchan:14,comm

ps axo stat,euid,ruid,tt,tpgid,sess,pgrp,ppid,pid,pcpu,comm

ps -eopid,tt,user,fname,tmout,f,wchan

orden top → proporciona una visión continuada de la actividad del procesador en tiempo real, muestra las tareas (procesos) que más uso hacen de la CPU, y tiene una interfaz interactiva para manipular procesos. Ordena los procesos mostrados en orden decreciente en base al uso de la CPU. Permite realizar una serie de acciones sobre los procesos de forma interactiva:

- Cambiar la prioridad de alguno utilizando la opción “r”.
- Matar o enviar una señal con la opción “k”.
- Ordenarlos según diferentes criterios (por PID con “N”, uso de CPU con “P”, tiempo con “A”, etc.).
- Con “n” se cambia el número de procesos que se muestran.
- Para salir se utiliza la letra “q”.

Las cinco primeras líneas muestran información general del sistema:

- las estadísticas de la orden uptime
- estadísticas sobre los procesos del sistema (número de procesos, procesos en ejecución, durmiendo, parados o zombies)
- el estado actual de la CPU (porcentaje en uso por usuarios, por el sistema, por procesos con valor nice positivo, por procesos esperando E/S, CPU desocupada, tratando interrupciones hardware o software, en espera involuntaria por virtualización)
- la memoria (memoria total disponible, usada, libre, cantidad usada en buffers y en memoria caché de página)
 - el espacio de swap (swap total disponible, usada y libre)

VIRT: Nos muestra el tamaño de memoria virtual de dicho proceso.

RES: Muestra el consumo puro de RAM, representando exactamente la cantidad de memoria física que usa el proceso. Este valor casi siempre es menor al VIRT.

SHR: muestra la cantidad de memoria compartida usada por la tarea, cuánto del tamaño VIRT es en realidad compartible (memoria o bibliotecas).

PR: prioridad del proceso. Si pone *RT* es que se está ejecutando en tiempo real.

NI: asigna la prioridad.

S (ESTADO): estado del proceso.

%CPU: porcentaje de CPU utilizado desde la última actualización.

%MEM: porcentaje de memoria física utilizada por el proceso desde la última actualización.

TIME+ (HORA+): tiempo total de CPU que ha usado el proceso desde su inicio.

COMMAND: comando utilizado para iniciar el proceso

`top -u nombre_usuario` muestra sólo los procesos del usuario especificado

pgrep <nombre_proceso> → da el identificador del proceso/s cuyo nombre se ha especificado

orden mpstat → Muestra estadísticas del procesador (o procesadores) del sistema contando desde cero 0 para el primer núcleo. También nos proporciona una media global de las actividades de todos en conjunto (instalado el paquete sysstat que se encuentra en el directorio /fenix/depar/lisi/so/paquetes).

Información de la cabecera:

CPU : número del procesador

%user : porcentaje de uso de la CPU con tareas a nivel de usuario

%nice : porcentaje de uso de la CPU con tareas a nivel de usuario con prioridad “nice” (>0)

%sys : porcentaje de uso de la CPU para tareas del sistema (no incluye el tratamiento de interrupciones) (modo núcleo)

%iowait : porcentaje de tiempo que la CPU estaba “desocupada” mientras que el sistema tenía pendientes peticiones de E/S

%irq : porcentaje de tiempo que la CPU gasta con interrupciones hardware

%soft : porcentaje de tiempo que la CPU gasta con interrupciones software (la mayoría son llamadas al sistema)

%idle : porcentaje de tiempo que la CPU estaba “desocupada” y el sistema no tiene peticiones de disco pendientes

intr/s : número de interrupciones por segundo recibidas por el procesador

Sintaxis:

`mpstat [intervalo] [número]`

Intervalo indica cada cuántos segundos debe mostrar los datos (un valor de 0 indica que las estadísticas de los procesadores serán reportadas desde que arrancó el sistema), y número, cuántos muestreos se solicitan.

Ej:

`mpstat 20 10`

Linux 4.10.0-35-generic (Aspire-F5) 12/10/17 _x86_64_ (4 CPU)

	CPU	%usr	%nice	%sys	%iowait	%irq	%soft	%steal	%guest	%gnice	%idle
01:15:05	all	3,54	0,00	1,13	0,11	0,00	0,05	0,00	0,00	0,00	95,17
01:15:25	all	4,88	0,00	1,65	0,05	0,00	0,13	0,00	0,00	0,00	93,30
01:15:45	all	6,26	1,85	2,33	0,42	0,00	0,28	0,00	0,00	0,00	88,86
01:16:05	all	8,02	0,00	2,56	0,20	0,00	0,32	0,00	0,00	0,00	88,90
01:16:25	all	8,51	0,00	1,85	0,15	0,00	0,23	0,00	0,00	0,00	89,25
01:16:45	all	13,30	0,00	4,10	1,13	0,00	0,42	0,00	0,00	0,00	81,05
01:17:05	all	29,94	0,01	6,40	1,96	0,00	1,12	0,00	0,00	0,00	60,57
01:17:25	all	24,13	0,00	3,11	1,00	0,00	0,42	0,00	0,00	0,00	71,34
01:17:45	all	22,38	0,00	2,70	1,11	0,00	0,34	0,00	0,00	0,00	73,47
01:18:05	all	17,79	0,00	1,94	0,97	0,00	0,18	0,00	0,00	0,00	79,12

Media: all 13,86 0,19 2,77 0,71 0,00 0,35 0,00 0,00 0,00 82,13

MEMORIA

orden free → se utiliza para visualizar el uso actual de memoria.

- Memoria real o principal (RAM) instalada en la computadora.
- Memoria de espacio de intercambio (swap) – esto es, del uso del espacio de intercambio en la partición de almacenamiento en disco correspondiente. Los sistemas operativos utilizan espacio de intercambio cuando necesitan alojar parte del espacio virtual de memoria de un proceso.

Se muestra la memoria total, usada y libre. Pero además se muestra la cantidad de memoria correspondiente a los búferes de disco utilizados por el kernel, y la cantidad de memoria que ha sido llevada a caché de disco. La línea -/+ refleja el total de memoria principal dividido en base a la memoria usada por cada una de estas dos últimas cantidades. Los valores de memoria bajo buffers y cached pueden considerarse como memoria libre.

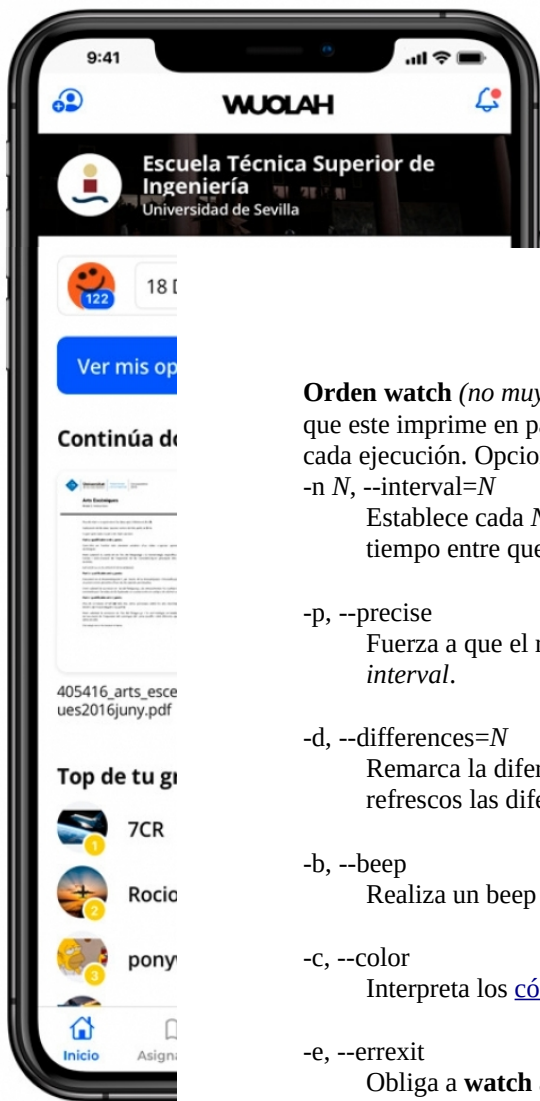
Opciones:

-b, --bytes	Display the amount of memory in bytes.
-k, --kilo	Display the amount of memory in kilobytes. This is the default.
-m, --mega	Display the amount of memory in megabytes.
-g, --giga	Display the amount of memory in gigabytes.
--tera	Display the amount of memory in terabytes.
-h, --human	Show all output fields automatically scaled to shortest three digit unit and display the units of print out. Following units are used.

B = bytes
K = kilos
M = megas
G = gigas
T = teras

If unit is missing, and you have petabyte of RAM or swap, the number is in terabytes and columns might not be aligned with header.

-w, --wide	Switch to the wide mode. The wide mode produces lines longer than 80 characters. In this mode buffers and cache are reported in two separate columns.
-c, --count count	Display the result count times. Requires the -s option.
-l, --lohi	Show detailed low and high memory statistics.
-s, --seconds seconds	Continuously display the result delay seconds apart. You may actually specify any floating point number for delay, usleep(3) is used for microsecond resolution delay times.
-t, --total	Display a line showing the column totals.
--help	Print help.
-V, --version	Display version information.



Descarga la APP de Wuolah.

Ya disponible para el móvil y la tablet.



Orden watch (*no muy importante*) → Sirve para ejecutar periódicamente un comando y mostrar lo que este imprime en pantalla completa, permitiendo ver claramente como cambian los datos tras cada ejecución. Opciones:

-n *N*, --interval=*N*

Establece cada *N* segundos se refresca la pantalla. Este parámetro considera a *N*, como el tiempo entre que finaliza la ejecución del comando y el próximo llamado al mismo.

-p, --precise

Fuerza a que el refresco de pantalla se logre en los segundos indicados en el parámetro -n o --interval.

-d, --differences=*N*

Remarca la diferencia del texto entre cada salida. Es posible indicarle que remarque entre *N* refrescos las diferencias. Vale aclarar que esta función no es perfecta.

-b, --beep

Realiza un beep en el parlante del sistema, si la ejecución del comando termina con errores.

-c, --color

Interpreta los [códigos de escape de color ANSI](#) del texto que se visualiza.

-e, --erexit

Obliga a **watch** a cerrarse si el comando se ejecuta con errores.

-t, --no-title

Indica que no se muestre la frecuencia de refresco, el comando y la hora cuando se ejecuta **watch**.

-x, --exec

Ejecuta el comando vía exec.

-g, --chgex

Exit when the output of command changes.

Orden vmstat → supervisa el sistema mostrando información de memoria pero también acerca de procesos, E/S, estado del sistema y CPU. Normalmente su primera salida proporciona una media desde el último arranque del sistema operativo. Se pueden obtener informes sobre el uso durante el periodo de tiempo actual, indicando el periodo de tiempo en segundos (primer argumento) y número de iteraciones deseadas (segundo argumento).

procs (procesos)

r muestra los procesos ejecutables (ejecutándose o en espera de ejecución)

b muestra cuantos procesos están en sueño ininterrumpible, lo que significa que están esperando un acceso E/S (disco, red, entrada de usuario, otros).

memory (memoria) en KB

swpd muestra la cantidad de memoria virtual utilizada.

free cantidad de memoria libre

buff cantidad de memoria usada en buffers (memoria intermedia)

cache cantidad de memoria usada en caché

swap

Estas columnas muestran la actividad del swap: cuantos bloques por segundo está intercambiando el sistema operativo desde disco (**si** - swapping in) y hacia disco (**so** - swapping out). Interesa que si y so estén en un valor 0 la mayor parte del tiempo.

io (E/S)

Estas columnas muestran cuantos bloques por segundo están siendo leídos desde (**bi**) y escritos hacia (**bo**) los dispositivos de bloques. Esto normalmente refleja la actividad de E/S del disco.

system

Estas columnas muestran el número de interrupciones por segundo (**in**) y el número de cambios de contexto por segundo (**cs**).

cpu

Porcentajes totales de tiempo de CPU consumido en ejecutar código de usuario (**us**), ejecutar código del sistema (**sy**, tareas del sistema como esperando E/S, actualizando estadísticas del sistema, gestionando prioridades, etc.), inactividad (**id**, idle) y espera de E/S (**wa**). Una posible quinta columna (**st**) muestra el porcentaje "robado" por una máquina virtual, si se está utilizando virtualización. Se refiere a el tiempo durante el cual algo pudo ejecutarse en la máquina virtual, pero el hipervisor eligió ejecutar algo más en su lugar. Si la máquina virtual no quiere ejecutar nada más y el hipervisor ejecuta algo más, eso no cuenta como tiempo robado.

OPCIONES

- n muestra la cabecera solo una vez y no periódicamente
- p <device> da estadísticas de la partición (dispositivo)
- d estadísticas del disco
- a muestra la memoria activa e inactiva (active, inact)

Control y gestión de dispositivos de E/S

Metadatos de archivo (también atributos de archivo): estructura de datos por cada archivo del SO que contiene la información que le es necesario mantener para poder trabajar con él. Los metadatos son "datos de los datos", es decir, describen todas las características de los archivos (los datos). Los metadatos se caracterizan por ser datos altamente estructurados que describen características de los datos, como el contenido, calidad, información y otras circunstancias o atributos. Existe un metadato de archivo fundamental que es el nombre de archivo, lo proporciona el usuario para identificar la información que contiene el archivo (**datos de archivo**). Este atributo se debe ubicar obligatoriamente en un sitio distinto al del resto de metadatos, se ubica siempre en una entrada de directorio.

Los archivos de tipo directorio están compuestos por entradas de directorio (una por cada archivo que contiene y cada entrada contiene la información del archivo correspondiente), que deben contener obligatoriamente el nombre de archivo.

inodo (inode): sitio del disco donde se almacena una referencia a los metadatos en la entrada de directorio y los propios metadatos. Es una estructura de datos propia de los sistemas de archivos. Un inodo contiene las características de un archivo regular, directorio, o cualquier otro objeto que

pueda contener el sistema de ficheros. Cada fichero se identifica por un número de inodo. Este número es único dentro de todo el sistema de ficheros.

Los SA tipo UNIX permiten establecer enlaces a archivos.

Tipos de archivos especiales de dispositivo que soportan los SA tipo UNIX: archivos especiales para dispositivos de bloques y para dispositivos de caracteres, los cuales proporcionan al usuario una interfaz abstracta para el trabajo con dispositivos de E/S y almacenamiento secundario.

Los datos de los archivos y los metadatos se almacenan en los dispositivos de almacenamiento persistente. Para la gestión de este almacenamiento, los sistemas de archivos tipo UNIX mantienen, entre sus metadatos de SA, información relativa a bloques de disco libres/ocupados e inodos libres/asignados. La falta de cualquiera de estos dos recursos (bloques de disco e inodos) puede provocar degradación en la utilización del SA por parte de los usuarios, e incluso, en un caso límite, los usuarios podrían dejar de poder utilizar el SA.

Orden ls → muestra los nombres de los archivos incluidos en un directorio.

Si no se especifica el nombre de un directorio, la orden ls actúa sobre el directorio de trabajo. Con la opción -l muestra determinada información relativa a los metadatos de los archivos incluidos en el directorio especificado.

La orden ls también posee capacidades de ordenación por campo. Ayudándonos de la orden **cut** la vamos a utilizar para ordenar la lista de información que nos proporciona (también podríamos utilizar la orden **sort**).

Opciones:

-l muestra en formato largo determinada información relativa a los metadatos de los archivos incluidos en el directorio especificado.

-n igual que -l pero muestra el usuario y grupo en forma numérica

-lh igual que -l pero el tamaño de los archivos se muestra en Kbytes, Mbytes o Gbytes (human readable format).

-X Ordena alfabéticamente (Sort alphabetically by directory entry extension).

-a todos los archivos, incluso los que comienzan con punto (.).

-A Lista todos los ficheros en los directorios, excepto los que comienzan con punto (.) y los que comienzan con doble punto (..).

-F indica tipo de archivo con un símbolo: / (directorio), * (ejecutable), @ (enlace simbólico).

-S clasifica por tamaños, con los ficheros más grandes en primer lugar.

-r invierte el orden de la salida.

-R Lista recursivamente los subdirectorios encontrados.

-t ordenar por fecha de última modificación.

-u ordenar por fecha de último acceso, el más reciente primero. Con -lt ordena y muestra la fecha de último acceso, con -l muestra esta fecha y ordena por nombre.

-U no ordena, muestra los archivos en el orden del directorio.

-x presenta los ficheros por filas.

-i precede la salida con el número de i-nodo de cada archivo

-d Lista la información (metadatos) del directorio en vez de la de los archivos contenidos en él.

-s Muestra el tamaño de cada archivo en bloques de 1024 bytes a la izquierda del nombre.

--color[=cuándo] Especifica si emplear color para distinguir los diferentes tipos de archivos. El argumento *cuándo* puede tener varios valores:

none No usar colores. Esta opción es la predeterminada.

auto Usar colores solamente cuando la salida estándar es una terminal.

always Usar siempre colores. Si **ls** se usa con la opción --color sin especificar la opción de color, el resultado es el mismo que cuando se usa --color=always.

-c muestra ordenado por la fecha de última modificación de la información de estado del archivo(ctime), el más reciente primero. Con -lt ordena y muestra el ctime, con -l muestra el ctime y ordena por nombre.

-L en los enlaces simbólicos, muestra los datos del archivo al que referencia en vez de los del link.

-m muestra los archivos en una línea y separados por comas.

Hay muchas más opciones, ver con man ls.

Campos con la opción -l:

El primer carácter de cada línea indica el tipo de fichero pudiendo ser:

- -, archivo regular.
- d, directorio.
- l, enlace simbólico (hay dos tipos de enlace en UNIX pero solo un tipo de archivo enlace).
- b, archivo especial de dispositivo de bloques.
- c, archivo especial de dispositivo de caracteres.
- p, archivo FIFO para comunicaciones entre procesos.

Los siguientes caracteres, indican los permisos del dueño u, grupo g y de los otros usuarios o.
El segundo campo indica el contador de enlaces duros a archivos.

El tercer y cuarto campo indican el propietario y el grupo propietario del fichero respectivamente.

El quinto campo indica el tamaño del fichero en kbs.

El sexto campo indica la fecha y hora de última modificación del fichero.

El último campo es el nombre del archivo.

Orden **cut** → Cortar un intervalo de caracteres

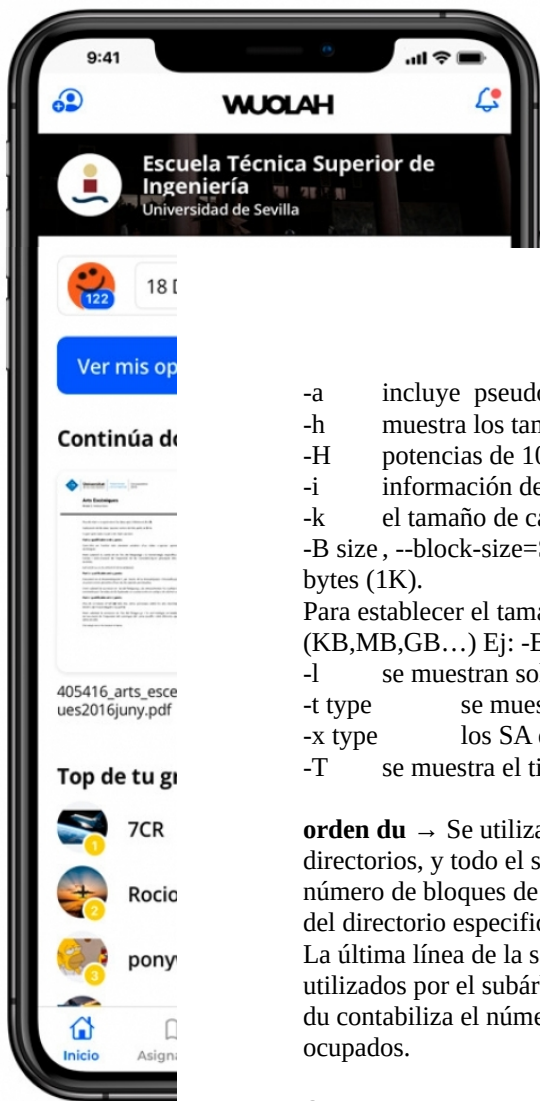
–d delimitador: indica cuál es el delimitador utilizado.

Cada información situada a ambos lados del delimitador representa un campo.

–f n: representa a la secuencia de caracteres del campo n, la información de la columna n separada por el delimitador.

orden df → permite visualizar, para cada SA montado (partición), información sobre su capacidad de almacenamiento total, el espacio usado para almacenamiento y el espacio libre restante, y el punto de montaje en la jerarquía de directorios para cada SA. Con la orden **df -i** podemos visualizar la información sobre los inodos de cada SA montado. Si no se le especifica el SA a df se muestra información de todos los sistemas de archivos.

Opciones:



Descarga la APP de Wuolah.

Ya disponible para el móvil y la tablet.



- a incluye pseudosistemas de archivos, duplicados y sistemas de archivos inaccesibles
 - h muestra los tamaños en potencias de 1024 (gigabytes, megabytes,...)
 - H potencias de 1000
 - i información de los inodos
 - k el tamaño de cada bloque es 1K
 - B size, --block-size=SIZE los bloques se establecen del tamaño size. Por defecto son de 1024 bytes (1K).
- Para establecer el tamaño podemos utilizar potencias de 1024 (K,M,G,T,P...) o potencias de 1000 (KB,MB,GB...) Ej: -B 10K
- l se muestran solo sistemas de archivos locales
 - t type se muestran solo los SA del tipo type
 - x type los SA de tipo type no se muestran
 - T se muestra el tipo de sistema de archivos

orden du → Se utiliza para ver el espacio en disco que gasta un directorio de la jerarquía de directorios, y todo el subárbol de la jerarquía que comienza en él. Proporciona el número de bloques de disco asignados a todos los archivos (incluidos directorios) que “cuelgan” del directorio especificado como argumento.

La última línea de la salida por pantalla muestra la cantidad total de bloques de disco utilizados por el subárbol.

du contabiliza el número de bloques de disco asignados estén o no completamente ocupados.

Opciones:

- h muestra los tamaños en potencias de 1024 (gigabytes, megabytes,...)
 - si potencias de 1000
 - k el tamaño de cada bloque es 1K
 - m el tamaño de bloque es 1M
 - b tamaño de bloque es 1byte
 - B size, --block-size=SIZE los bloques se establecen del tamaño size. Por defecto son de 1024 bytes (1K).
- Para establecer el tamaño podemos utilizar potencias de 1024 (K,M,G,T,P...) o potencias de 1000 (KB,MB,GB...) Ej: -B 10K
- s muestra solo el total de espacio ocupado por el argumento
 - time muestra el tiempo de última modificación de los archivos y subdirectorios.
 - inodes muestra información de uso de los inodos en vez de información de bloques
 - a muestra información de todos los archivos, no solo directorios
 - X File excluye directorios que se ajusten al patrón en File
 - x excluye directorios que estén en diferentes sistemas de archivos

Para encontrar los 5 directorios más pesados en nuestro home:

```
du -hs * | sort -nr | head -5
```

ENLACES A ARCHIVOS

El objetivo de los enlaces a archivos es disponer de más de un nombre para los archivos en nuestro espacio de nombres de archivo soportado por la estructura jerárquica de directorios.

Enlaces simbólicos → referencia al nombre de otro archivo

Enlaces duros → referencia a los metadatos de otro archivo, es un nombre distinto para los metadatos (inodo) de un archivo ya existente.

Todos los nombres de archivo forman un enlace duro sobre el inodo asociado a ese archivo simplemente por existir en el espacio de nombres, es decir, al crear un archivo se establece el primer enlace duro sobre su inodo asociado, cuando el SO crea la entrada en el directorio en donde se va a ubicar.

Todos los directorios tienen dos enlaces duros que crea el SO automáticamente para mantener la jerarquía de directorios mediante el establecimiento de la relación jerárquica (directorio_padre, directorio_hijo). Estos enlaces duros se identifican con las entradas cuyos nombres de archivo son: “.” para el enlace duro al directorio en el que se encuentra la entrada asociada y, “..” para el enlace duro al directorio padre del directorio en el que se encuentra la entrada correspondiente.

El contador de enlaces duros (que podemos ver con la orden `ls -l`) mantiene el número de enlaces duros a archivos con el objetivo de poder liberar el inodo cuando todos los nombres de archivo que utilizan dicho inodo hayan sido eliminados de la estructura de directorios y nunca antes, pues si se liberase el inodo con un contador de enlaces mayor que 0, se podría intentar acceder al archivo y se produciría una inconsistencia entre el espacio de nombres y los metadatos de archivo.

orden ln → Para crear enlaces duros o enlaces simbólicos sobre un archivo creado previamente . Como argumentos básico debemos proporcionarle el nombre del archivo a enlazar (link target) y el nuevo nombre de archivo (link name). Por defecto crea enlaces duros.

```
ln [OPTION]... [-T] TARGET LINK_NAME (1st form)
ln [OPTION]... TARGET (2nd form)
ln [OPTION]... TARGET... DIRECTORY (3rd form)
ln [OPTION]... -t DIRECTORY TARGET... (4th form)
```

Con la primera forma se crea un enlace a TARGET con el nombre LINK_NAME. Con la segunda forma se crea un enlace a TARGET en el directorio actual. Con la tercera y cuarta forma se crea un enlace a cada TARGET en DIRECTORY. No debería existir el nombre del nuevo enlace y cada TARGET debe existir obligatoriamente cuando creamos enlaces duros.

Opciones:

- v Imprime el nombre del archivo antes de enlazarlo.
- i Pide confirmación si fichero_nuevo existe en destino. Si existe fichero_nuevo y no ponemos esta opción no permite hacer el enlace.
- s Hace enlaces simbólicos en lugar de enlaces duros
- f Elimina el fichero enlace si existía con anterioridad.
- n No sobrescribe archivos existentes.
- b hace una copia de seguridad de cada fichero de destino existente (en caso de que exista)
- t DIRECTORY especifica el directorio donde crear los enlaces
- T trata a LINK_NAME como un archivo normal.

ARCHIVOS ESPECIALES DE DISPOSITIVO

Los dispositivos de nuestro sistema se representan en un SO tipo UNIX mediante archivos especiales de dispositivo. Existen dos tipos principales de archivos especiales de dispositivo: de bloques (representan a dispositivos de bloques, que normalmente coinciden con los dispositivos de almacenamiento persistente, los ramdisks y los dispositivos loop) y de caracteres (representan a dispositivos de caracteres del tipo puertos serie, paralelo y USB, consola virtual (console), audio, los dispositivos de terminal (tty),etc).

El directorio `/dev` contiene los archivos de dispositivos especiales para todos los dispositivos hardware.

Un fichero especial es una tripleta (booleano, entero, entero) almacenado en el sistema de ficheros. El valor lógico escoge entre fichero especial de caracteres y fichero especial de bloque. Los dos enteros son los números de dispositivo mayor y menor.

Así, un fichero especial casi no ocupa sitio en el disco, y se emplea sólo para la comunicación con el sistema operativo, no para almacenamiento de datos. A menudo los ficheros especiales se refieren a dispositivos físicos (disco, cinta, terminal, impresora) o a servicios del sistema operativo (`/dev/null`, `/dev/random`).

Los ficheros especiales de bloque son dispositivos similares a discos (donde se puede acceder a los datos dado un número de bloque, y p.ej. tiene sentido tener un caché de bloques). Todos los otros dispositivos son ficheros especiales de caracteres

orden mknod → crear archivos especiales de dispositivo, tanto de bloques (buffered) como de caracteres (unbuffered). Permite especificar el nombre del archivo y los números principal (major) y secundario (minor). Estos números permiten identificar a los dispositivos en el kernel, concretamente en la Tabla de Dispositivos. El número principal determina el controlador (es un programa informático que permite al sistema operativo interactuar con un periférico, haciendo una abstracción del hardware y proporcionando una interfaz -posiblemente estandarizada- para usarlo (drivers)) al que está conectado el dispositivo y el número secundario al dispositivo en sí. Es necesario consultar el convenio de nombres dado que la asignación de estos números va a depender de la distribución.

`mknod [OPTION]... NAME TYPE [MAJOR MINOR]`

NAME es el nombre del archivo a crear

TYPE es el tipo de fichero: b(bloques), c/u (caracteres), p(FIFO)

MAJOR MINOR es el número principal y secundario. Se dan en valor hexadecimal (0x), octal (empieza por 0) o decimal (en otro caso).

-m modo, --mode=modo

Establece los permisos de los ficheros creados a *modo*, que es simbólico como en `chmod` y emplea el modo predeterminado (los permisos de los ficheros creados son 0666 ('a+rw') menos los bits puestos a 1 en la `umask`) como punto de partida.