

**UD3: OPERATING SYSTEMS
PRIMEROS PASOS CON UBUNTU**

**CFGM DAW
DPT INF**

PRIMEROS PASOS UBUNTU

En Ubuntu existen tres versiones: **Desktop** (para equipos de escritorio), **Netbook** (para portátiles) y **Server** (para servidores). Además, para cada tipo puedes utilizar la versión de **32 bits** y de **64 bits**.

1. x-Windows

Uno de los elementos que ha propiciado la gran expansión de los sistemas GNU/Linux en empresas y hogares es la utilización de entornos gráficos sencillos y amigables. Los sistemas GNU/Linux cuentan con diversos entornos gráficos, muy potentes, que permiten utilizar el sistema fácilmente.

x-Windows (o sistema de ventanas X) es el nombre por el que se conoce al **entorno gráfico usado por los sistemas Unix**. Desarrollado desde mediados de la década de los 80 en el MIT (Instituto Tecnológico de Massachussets) se encuentra actualmente en su versión 11, por lo que normalmente suele ser referenciado como X11.

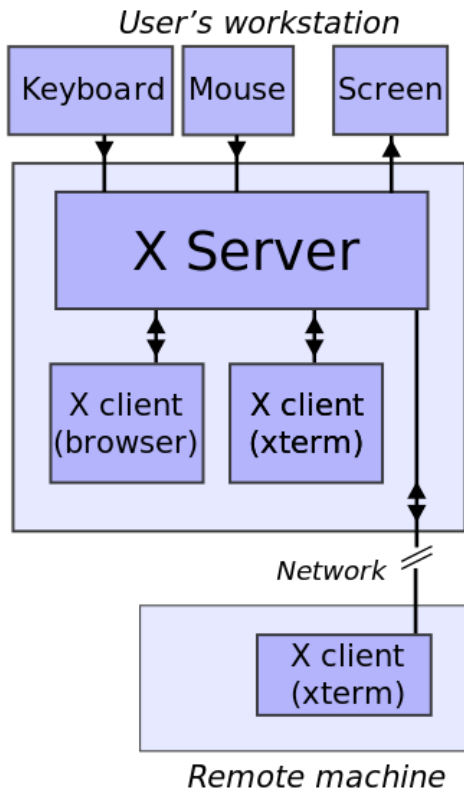
x-Windows proporciona una interfaz gráfica (GUI) al mundo de Linux. x-Windows, al igual que el sistema operativo Windows de Microsoft, ofrece una forma de manejo de algunos de los elementos de interacción más comunes como ventanas, cuadros de diálogo, botones y menús.

X Windows está construido con una arquitectura cliente-servidor y es una capa totalmente independiente del Sistema Operativo. El **servidor de X-Windows**, ejecutado sobre la máquina servidora, se encarga de generar las instrucciones gráficas. El **cliente de X-Windows**, ejecutado sobre los clientes conectados a la máquina servidora, es el encargado de convertir las instrucciones gráficas que recibe del servidor en las instrucciones que precisa el sistema operativo para mostrar las ventanas y su contenido.

Gracias a esta arquitectura se consigue aislar el servidor X-Windows de los diferentes sistemas operativos que se conecten como clientes.

UD3: OPERATING SYSTEMS

CFGM DAW
DPT INF



En este ejemplo, el servidor X toma datos de entrada desde el teclado y el ratón y los muestra en una pantalla. Un [navegador web](#) y un [emulador de terminal](#) se ejecutan en la workstation del usuario, y un emulador de terminal se ejecuta en un servidor remoto pero es controlado desde el ordenador del usuario. Notar que las aplicaciones remotas se ejecutan de la misma manera que lo harían en forma local.

Estructura del sistema gráfico X Window

El sistema gráfico X Window consta de varias capas de software que constituyen la interfaz gráfica de usuario:

- **Sistema de ventanas X (Es el propio servidor X)**
Es el responsable de las operaciones gráficas básicas como dibujar los iconos, los fondos, y las ventanas en las que se ejecutan las aplicaciones.
- **Gestor de ventanas (Puede formar parte del entorno de escritorio o ser independiente.)**
Añade elementos alrededor de las ventanas, de forma que el usuario pueda cambiar el tamaño, cerrar, ocultar, mover...
- **Gestor de pantalla (Forma parte del cliente X)**
Gestiona la sesión de usuario. También llamado Display Manager y es lanzado como un servicio del proceso init1.
- **Entorno de escritorio**
El entorno de escritorio puede ser, desde un simple gestor de ventanas, hasta un conjunto muy completo de aplicaciones de escritorio. En realidad es lo que 've' el usuario después de abrir la sesión.

Esta forma de diseño origina que existan diferentes implementaciones de gestores de ventanas, con diferentes características funcionales y visuales. El servidor X, es

UD3: OPERATING SYSTEMS

CFGM DAW
DPT INF

altamente portable y en el caso de Ubuntu permite utilizar los tres principales entornos de escritorio o GUI (Graphical User Interfaces o Interfaces Gráficas de Usuario): GNOME está orientado a la simplicidad, KDE ofrece un mayor conjunto de aplicaciones así como posibilidades de caracterización por defecto, o Xfce está optimizado para su uso con requisitos hardware bajos. Aparte de los citados, es posible utilizar otros entornos de escritorio como Fluxbox, Sugar o LXDE.

Por defecto Ubuntu Desktop utiliza el entorno gráfico GNOME.

2. Primeros pasos con Linux

2.1. Intérprete de comandos

El intérprete de comandos o shell del sistema es la interfaz entre el usuario y el sistema operativo. La función del shell es recibir las órdenes del usuario a través de la línea de comandos, interpretarlas, ejecutarlas y mostrar su resultado.

Resulta muy útil aprender a utilizar el shell del sistema ya que aunque al principio puede parecer un poco difícil, resulta fundamental para obtener el máximo rendimiento del sistema. El shell permite interactuar directamente con el sistema y con sus ficheros de configuración.

En Ubuntu, entramos en Terminal.

Una vez que accedes al sistema se muestra un prompt con el siguiente aspecto o parecido:

usuario@ubuntu-virtual-machine:~\$

donde **usuario** es el nombre del usuario que está utilizando el sistema, **@ubuntu-virtual-machine** indica el nombre del equipo. A continuación, se muestra el directorio en el que se encuentra. En el caso de que se encuentre el carácter **~** es porque está en el directorio **home**. Por último, el símbolo **\$** o **#** indica si es un usuario normal (**\$**) o es el administrador del sistema (**#**).

El usuario **root** es el administrador del sistema y puede realizar cualquier tarea de administración. En algunas distribuciones puede acceder directamente al sistema como usuario root, pero otras distribuciones, como Ubuntu, te obligan a acceder al sistema con un usuario sin privilegios de administrador y luego cambiar de usuario.

Si deseas ejecutar una tarea de forma puntual como root puedes utilizar el comando **sudo** de la siguiente forma:

UD3: OPERATING SYSTEMS

CFGM DAW
DPT INF

\$ sudo <comando>

Si necesita ejecutar múltiples tareas puedes obtener un shell de root ejecutando sudo bash o su (substitute user):

\$ sudo bash
#

su [Nombre de Usuario]
sudo -s

Además, si lo deseas, puedes activar la cuenta de *root* al establecer su contraseña:

\$ sudo passwd root

\$sudo [nombre usuario] para cambiar la contraseña a cualquier usuario

A continuación, puedes ver los comandos más utilizados en los sistemas GNU/Linux.

2.2. Instalar y quitar componentes.

En GNU/Linux se puede realizar la instalación de una aplicación directamente a **partir del código fuente** o a **través de la aplicación compilada** (paquete). A las aplicaciones preempaquetadas se le denomina **paquete** y contienen los binarios, los archivos complementarios y archivos de configuración para poder ejecutarse.

Para facilitar el proceso de instalación se utilizan **gestores de paquetes** que facilitan la administración de los paquetes. A continuación, vas a aprender a instalar aplicaciones de todas las formas posibles: mediante *x-Windows*, mediante apt-get, directamente instalando paquetes deb o a partir del código fuente. Las dos primeras formas son las más fáciles de utilizar y por lo tanto las recomendadas.

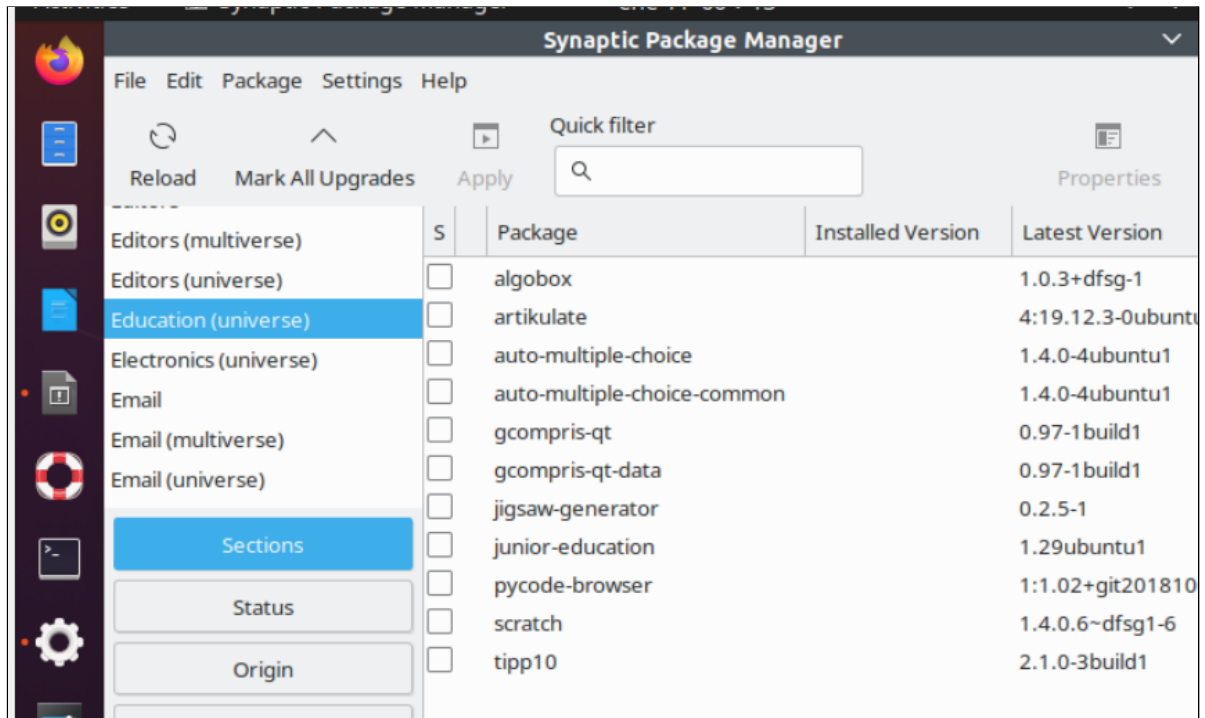
Synaptic.

Synaptic es una herramienta de x-Windows que facilita las tareas de instalación y eliminación de software. Si tenemos la última versión de Ubuntu para tener Synaptic tendremos que dirigirnos al Centro de Software de Ubuntu y buscar Synaptic e instalarlo.

Una vez iniciada la herramienta (véase la figura adjunta) el menú de la izquierda muestra las diferentes categorías de aplicaciones. Si pulsas en una categoría aparecen sus diferentes aplicaciones. Si una aplicación ya se encuentra instalada en el sistema su campo de selección se encuentra activo. Si deseas instalar o desinstalar una aplicación sólo debes seleccionar o deseleccionar la aplicación y pulsar el botón Aplicar.

UD3: OPERATING SYSTEMS

CFGM DAW
DPT INF



Se pueden definir repositorios adicionales para aumentar la disponibilidad de software del sistema. Un **repositorio** es una ubicación de red que almacena paquetes de software junto a los metadatos que los describe. Los repositorios también se utilizarán con el comando **apt-get** que veremos más adelante.

http://cefire.edu.gva.es/file.php/1/LLiurex_pera_la_tasca_docent/Unidad_5/gestor_de_paquetes_synaptic.html

apt-get.

apt-get permite instalar o desinstalar por línea de comandos cualquier paquete. Para empezar, apt-get utiliza una serie de repositorios que se encuentran en el fichero `/etc/apt/sources.list`. Si lo deseas, puede modificar los repositorios del sistema y actualizar el sistema ejecutando:

apt-get update

A continuación, se van a ver los procedimientos más utilizados:

- 🌐 **Actualizar el sistema.** Permite actualizar el sistema con todas las dependencias. Se utiliza:

apt-get upgrade

UD3: OPERATING SYSTEMS

CFGM DAW
DPT INF

- 🌐 **Búsquedas.** Permite localizar un paquete o término en alguno de los repositorios.
Se ejecuta:

apt-cache search <nombre>

,donde *nombre* indica el nombre del paquete que desea buscar.

- 🌐 **Consulta de información.** Permite consultar información de un paquete.

apt-cache show <paquete>

Por ejemplo si quiere información sobre el servidor web ejecute:

```
#apt-cache show apache2
```

- 🌐 **Instalación de paquetes.** Permite realizar la instalación de paquetes con la resolución automática de dependencias (es decir, paquetes que dependen de otros paquetes y si éstos últimos no están instalados no podremos instalar el primero).

apt-get install <paquete>

Por ejemplo si desea instalar el servidor web ejecute:

```
# apt-get install apache2
```

- 🌐 **Desinstalar un paquete.** Para desinstalar un paquete hay que ejecutar:

apt-get remove <paquete>

Por ejemplo si desea desinstalar el servidor web ejecute:

```
# apt-get remove apache2
```

*(A la hora de buscar o instalar un programa puede utilizar el carácter * para indicar cualquier carácter. Por ejemplo si desea instalar cualquier aplicación que empiece por php entonces ejecuta apt-get install php-*.)*

Aptitude

aptitude es un gestor de paquetes por línea de comandos muy cómodo y sencillo

UD3: OPERATING SYSTEMS

CFGM DAW
DPT INF

de utilizar. Para poder utilizarlo antes necesitas instalarlo.

root@ubuntu ~ #apt-get install aptitude

Ejecuta en el terminal como *root* el comando:

root@ubuntu ~ #aptitude

Una vez ejecutada la herramienta puedes acceder a las diferentes categorías e instalar o desinstalar el software.

Instalación manual

Si lo deseas puedes realizar la instalación o desinstalación directa de un paquete. Para realizar la instalación debes descargar previamente el paquete y ejecutar:

dpkg -i nombre_paquete

Si por el contrario deseas eliminar un paquete, primero debes conocer su nombre exacto. Para ello debes ejecutar:

dpkg-query -s nombre

Una vez que conoce el nombre exacto se realiza la instalación ejecutando:

dpkg -r nombre_completo

A veces se encuentran aplicaciones que no proporcionan paquetes de instalación, y hay que compilar a partir del código fuente. Para ello, lo primero que debe realizar es instalar las herramientas de compilación ejecutando:

apt-get install build-essential

Además, puedes realizar la instalación directamente desde el código fuente, es posible que surja algún problema de dependencias. Si sucede esto, entonces debes resolver la dependencia y continuar con el proceso de instalación.

Práctica (realiza capturas donde aparezcan las instrucciones ejecutadas):

Verificamos que el compilador de GCC se ha instalado correctamente mostrando la versión del mismo.

Ejecuta desde el terminal: **\$ gcc --version**

```
ubuntusin@ubuntusin-VirtualBox:~$ gcc --version
gcc (Ubuntu 9.3.0-17ubuntu1~20.04) 9.3.0
Copyright (C) 2019 Free Software Foundation, Inc.
This is free software; see the source for copying conditions. There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
```

Creamos ahora nuestro código con el editor de texto nano.

Ejecuta desde el terminal: **\$ nano hello.c**

UD3: OPERATING SYSTEMS

CFGM DAW
DPT INF

Escribimos el siguiente código en C:



Con [Ctrl + X] guardamos el fichero.

Compilamos el código fuente ejecutando desde el terminal:

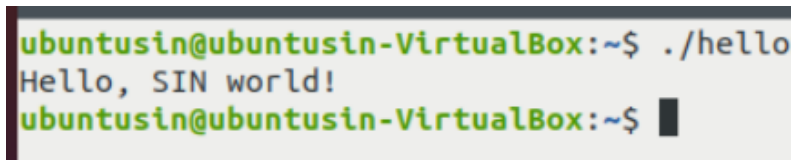
```
$ gcc hello.c -o hello
```

Con esto creamos el fichero binario hello en el mismo directorio donde estamos
(Compruébalo ejecutando el comando ls)

Ahora ejecutamos el programa hello:

Ejecuta desde el terminal: `$./hello`

El resultado debe ser similar a lo mostrado en la imagen siguiente:



Nota: En general, los pasos a seguir para compilar una aplicación son los siguientes:

- Descarga el código fuente.
- Descomprime el código, generalmente está empaquetado con tar y comprimido con gzip (*.tar.gz o *.tgz) o bzip2 (*.tar.bz2).
- Accede a la carpeta creada al descomprimir el código.
- Ejecuta el script ./configure que permite comprobar las características del sistema que afectan a la compilación y crear el archivo makefile.
- Compila el código ejecutando el comando make.
- Instala la aplicación en el sistema ejecutando make install. Si desea desinstalar la aplicación entonces ejecuta make clean.

UD3: OPERATING SYSTEMS

CFGM DAW
DPT INF

3. Distribuciones basadas en Ubuntu

En el siguiente link podrás ver la apariencia de las distintas distribuciones.

<https://www.softzone.es/programas/linux/mejores-distribuciones-ubuntu/>

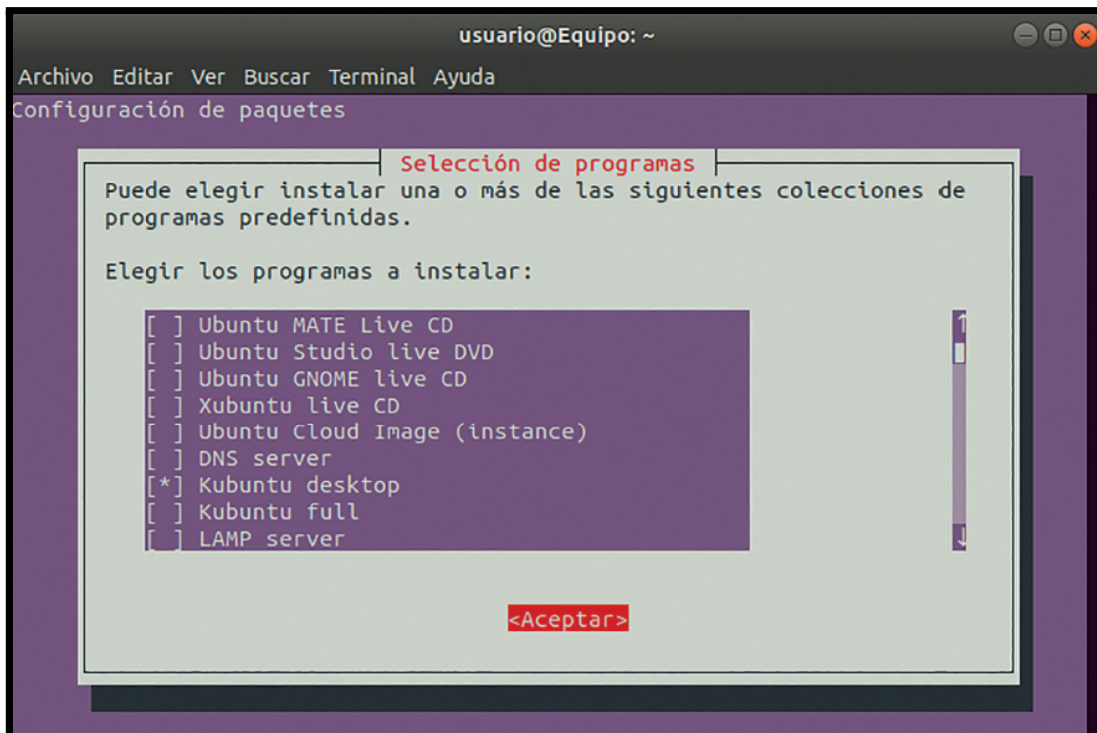
Elige uno de los escritorios e instálalo sobre tu máquina virtual clonadaa partir de tu MV original. Para ello dispones de las siguientes 2 opciones:

Si quisiéramos el escritorio de Kubuntu ejecutaríamos:

```
sudo apt install kubuntu-desktop
```

O instalando la utilidad tasksel:

```
sudo apt install tasksel  
sudo tasksel
```



Una vez instalado, cierra sesión y al acceder de nuevo tendrás posibilidad de elegir el escritorio original de Ubuntu o el que acabas de instalar.

UD3: OPERATING SYSTEMS

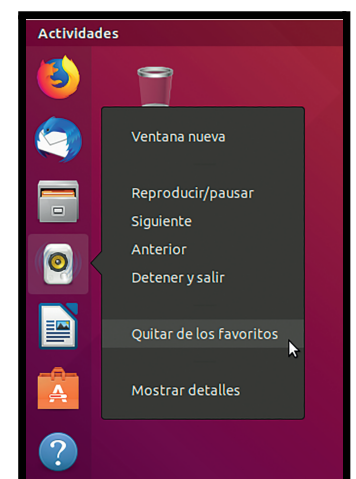
CFGM DAW
DPT INF

4. Escritorio Ubuntu

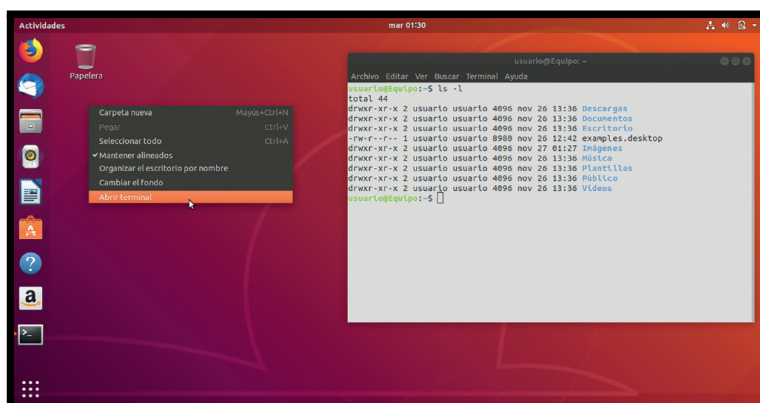
Configuración: Acerca de veremos: memoria, procesador, memoria gráfica, si el procesador trabaja a 32 o 64 bits y el espacio en disco duro. Vemos también la configuración de las **“Actualizaciones de Software”**

Añade al Dock tus aplicaciones preferidas.

Dock: Para desanclar una aplicación, con el botón derecho del ratón sobre su icono elegimos: -> Quitar de los favoritos. De igual forma sobre una aplicación que vayamos a utilizar mucho la podemos anclar al Dock



Shell



- Ctrl + Alt + F1 ... Ctrl + Alt + F6 -> entramos en las terminales o consolas virtuales
- Carácter escape: Si queremos escribir un nombre de directorio o fichero que tenga espacios en blanco, podemos escribirlo entre comillas, o bien utilizar el carácter \ delante del espacio en blanco. (Se le llama

UD3: OPERATING SYSTEMS

CFGM DAW
DPT INF

- carácter de escape porque protege al carácter que viene detrás de él.)
- Histórico de comandos: Con las flechas del cursor podemos buscar un comando escrito previamente.
- Función Autocompletar: Si se va a escribir el nombre de un fichero o directorio como argumento de un comando se puede escribir la primera o las primeras letras y pulsar la Tecla tabulación. Entonces se termina de escribir el nombre.
- Cambia la pantalla de bloqueo y actívala de manera que a los 3 minutos se active (Configuración).
- Cambia el fondo de pantalla y añádele una imagen (Configuración/Sistema)
- Añade atajos de teclado para aumentar y reducir el tipo de letra (Configuración/Dispositivos/Teclado).

Práctica (realiza capturas donde aparezcan las instrucciones):

Partiendo de los comandos mostrados al final de este documento, ejecuta las instrucciones desde el terminal que devuelvan lo siguiente:

- Muestra en el terminal el nombre del sistema operativo y el nombre del equipo.
- Conéctate como usuario en la tercera terminal virtual. Regresa al modo gráfico y abre la terminal. Mira tu nombre de usuario y los usuarios que están conectados al sistema y dónde.
- Entra en modo texto y crea un alias llamado **quien**, que te borre la pantalla, te diga quién eres y muestre la fecha actual.
- Busca dónde están los comandos **date** y **uname** dentro del sistema de ficheros. Muestra en una sola línea el día, el mes y el día de la semana. A continuación, muestra el nombre del sistema operativo.
- Muestra el calendario del mes y del año en que naciste.
- Muestra la fecha actual
- Borra la pantalla
- Muestra información sobre el núcleo del sistema operativo que está instalado.
- Mira la ayuda sobre el comando **whereis**
- Crea un alias llamado **meses** que borre la pantalla, muestre el mes actual, el anterior y el próximo, y después escriba el nombre de usuario. Comprueba su funcionamiento.
- Mira los alias definidos en el sistema.
- Elimina el alias **meses** y comprueba que lo has borrado
- Borra el historial de los comandos
- Muestra el mes actual, el anterior y el posterior
- Sal de la aplicación Terminal

UD3: OPERATING SYSTEMS

CFGM DAW
DPT INF

date

Muestra el día y la hora del sistema. También se puede utilizar para cambiarlas.

Sintaxis:

date

cal

Muestra el calendario del mes y año que se le indique. Si no se le especifica nada, muestra el mes actual.

Sintaxis:

cal [-3my] [[mes] año]

Opciones:

-3

Muestra el mes actual, el anterior y el próximo.

-m

Muestra el lunes como primer día.

-y

Muestra el año actual.

Argumentos:

mes: indica el mes que queremos mostrar, si no se especifica mostrará el año entero.

año: indica el año del que queremos ver su calendario.

who

Muestra quién está conectado al sistema.

Sintaxis:

who [opciones]

Opciones:

-a | --all

Muestra, además de quién está conectado, información adicional.

UD3: OPERATING SYSTEMS

CFGM DAW
DPT INF

whoami

Muestra el nombre de usuario.

Sintaxis:

whoami

man

info

Estos dos comandos nos muestran una ayuda detallada del comando que se le pase como parámetro.

Sintaxis:

man [comando]

info [comando]

Si queremos salir de la ayuda hay que pulsar la letra «q» (*quit*).

Para la mayoría de los comandos, existe una opción que muestra una ayuda más resumida.

comando [--help | -h]

clear

Borra la pantalla.

Sintaxis:

clear

whereis

Busca dónde está dentro del sistema de ficheros el comando que se le pase como parámetro.

Sintaxis:

whereis comando

UD3: OPERATING SYSTEMS

CFGM DAW
DPT INF

alias

Asigna un nombre a la ejecución de uno o varios comandos con sus respectivas opciones y argumentos.

Sintaxis:

alias [nombre='comando1;comando2;...']

El carácter «;» se utiliza en Linux como separador de comandos.

El comando solo muestra todos los alias que hay creados.

unalias

Elimina el alias especificado.

Sintaxis:

unalias nombre

El argumento nombre debe ser un alias que ya exista o hayamos creado en el sistema.

uname (*unix name*)

Muestra diferente información sobre el sistema operativo y el equipo.

Sintaxis:

uname [opciones]

Si no se especifica ninguna opción equivaldría a la opción -s.

Opciones:

-a | --all

Muestra toda la información, excepto la que se muestra con -p y -i.

-s | --kernel-name

Muestra el nombre del núcleo.

-n | --nodename

Muestra el nombre del equipo dentro de la red de ordenadores en la que se encuentre.

UD3: OPERATING SYSTEMS

CFGM DAW
DPT INF

-r | --kernel-release

Muestra la versión del núcleo.

-v | --kernel-version

Muestra la fecha de la compilación del núcleo.

-m | --machine

Muestra el nombre hardware del equipo.

-p | --processor

Muestra el tipo de procesador o *unknown* (desconocido).

-i | --hardware-platform

Muestra el tipo de hardware o *unknown* (desconocido).

-o | --operating-system

Muestra el sistema operativo.

passwd

Cambia la contraseña del usuario.

Sintaxis:

`passwd [opciones] [usuario]`

Sin opciones ni argumentos cambia la contraseña del usuario actual. Para cambiar la contraseña de otro usuario, o para utilizar ciertas opciones, es necesario tener permisos de administrador del sistema.

history

Muestra los comandos utilizados. Tiene un número de comandos almacenados que se puede modificar.

Sintaxis:

`history [opciones]`

Sin opciones ni argumentos muestra los últimos comandos que se hayan ejecutado en la terminal.

Opciones:

-c

Borra el historial de comandos, que se encuentra en el archivo **.bash_history**, en el subdirectorio personal del usuario.

UD3: OPERATING SYSTEMS

CFGM DAW
DPT INF

exit

Termina la sesión de un usuario o sale de la terminal en modo texto.

Sintaxis:

exit

5. SUDO (SuperUser DO o Substitute User DO)

En las versiones antiguas de Linux, teníamos 2 tipos de usuarios, el usuario root y el resto de usuarios sin privilegios.

Ahora tenemos **sudo**, para que usuarios **pseudo-administradores** puedan ejecutar comandos como si fuesen **root**, pero sin acceder a él, de forma que no es necesario conocer el password de root.

“Sudo es un programa diseñado para permitir a los administradores de sistemas proporcionar los privilegios limitados de root a ciertos usuarios y registra las actividades de root. La filosofía básica es dar tan pocos privilegios como sea posible, pero permitiendo que la gente pueda desarrollar su trabajo”

No todos los usuarios pueden usar sudo, solo aquellos que se definen como administradores, cuando se crean.

El fichero de configuración de sudo es /etc/sudoers. En él se indica qué usuarios pueden modificar qué comandos y en qué modo. El comando que se utiliza para modificar el fichero sudoer es **visudo**.

Actividad: Accede al fichero /etc/sudoers utilizando visudo. Averigua las opciones de visudo para modificar este fichero, las zonas que forman sudoers y cómo se editan sus líneas para facilitar su uso a determinados usuarios.

UD3: OPERATING SYSTEMS

CFGM DAW
DPT INF

5.1. Opciones de sudo

- **u (sudo -u usuario comando).** Permite ejecutar comandos de otros usuarios
- **e (sudo -e fichero).** Permite modificar ficheros de configuración que solo el root puede modificar, como, sudo -e /etc/passwd.
- Algunas de las diferencias entre estas dos utilidades se enumeran a continuación:

sudo	su
Los comandos son ejecutados por un usuario regular que debe ser parte de un grupo de usuarios con la posibilidad de utilizar sudo (sudoers).	Sirve para cambiar de un usuario a otro, generalmente se pasa de un usuario regular al usuario <i>root</i> .
Solo es necesaria la contraseña del usuario actual.	Se requiere conocer la contraseña del usuario <i>root</i> , la cual no debería ser revelada a ningún usuario regular.
Se pueden registrar las acciones ejecutadas bajo la figura de sudo .	Las posibilidades para registrar eventos son más limitadas.
Ofrece varias características que proveen un mayor control de lo que hacen —y pueden hacer— los usuarios.	Una vez que accede como <i>root</i> , no hay control de lo que puede hacer en el sistema.

6. El usuario Root

La administración real del sistema y el acceso a todos sus lugares solo es permitido por root.

UD3: OPERATING SYSTEMS

CFGM DAW
DPT INF

6.1. Comando su

su nos permite cambiar de usuario. Su sintaxis es: **su usuario** (si no se especifica nada y escribimos **su** indicamos al sistema que queremos acceder como root)

El password del root puede ser modificado por un administrador: **sudo passwd root**

Si escribimos: **passwd -e root** eliminaremos el password de root y el usuario no podrá acceder al sistema.

7. Uso del terminal

7.1. El prompt

Caracteres que se muestran en el terminal precediendo a aquello que vamos a ejecutar.

El prompt viene definido por la variable de entorno **PS1**. (usuario@usuarioPC~\$)

7.2. Variables de entorno

Son elementos del sistema operativo que almacenan un valor que puede variar durante la ejecución del mismo. El sistema lo usará en determinados momentos según los procesos que se estén ejecutando.

Las variables de entorno más usuales son:

PATH: Contiene información de todos los directorios que almacenan ficheros ejecutables, de modo que podemos ejecutar los mismos sin necesidad de acceder a la carpeta en la que se almacenan. El S.O., cuando decidamos ejecutar un comando lo buscará en los directorios indicados en el PATH, si no lo encuentra mostrará un mensaje de error.

HOME: Almacena la ruta al directorio personal del usuario con el que hemos iniciado sesión.

USER : Almacena el nombre del usuario que ha iniciado sesión.

SHELL: Almacena la ruta del intérprete de comandos que se está ejecutando por defecto.

HOSTNAME: Guarda el nombre del PC

Actividad: Podemos modificar el prompt. Localiza si existe, el comando que permite el cambio o

UD3: OPERATING SYSTEMS

CFGM DAW
DPT INF

cómo podríamos cambiarlo usando la variable de entorno PS1.

7.3. Comandos más usuales en SO Linux (Ubuntu)

Comandos para obtener ayuda sobre los distintos comandos

man comando (ejecuta man date y comprueba el resultado obtenido)

info comando

comando –help

Sintaxis básica: **nombre_del_comando [opciones] [valor]**

Lo incluido entre corchetes indica opcionalidad

[opciones]: modificadores que al agregarse al comando modifican su comportamiento base. Ej: **ls** muestra los ficheros y directorios de la carpeta donde nos encontramos, si escribimos **ls -l**, vemos el mismo contenido pero con mucha más información.

[valor]: Puede ser un fichero, un directorio, un número o conjunto de caracteres, etc., dependerá del tipo de comando.

Comandos para la administración de procesos

Un proceso es un programa en ejecución. Los procesos se identifican con un número entero denominado **PID**. Además, los procesos son generados por otros procesos denominados procesos padre. Este proceso padre, se identifica igualmente con un número denominado **PPID**. Todo proceso es propiedad de un usuario.

Ejecución en primer y segundo plano

Cuando lanzas un comando en el terminal, por ejemplo **ls** tienes que esperar hasta que termine. Por el contrario, un proceso que se ejecuta en segundo plano no interfiere en otras labores. Si lanzas un proceso en segundo plano desde un terminal, tu podrás continuar utilizando el terminal, mientras que este proceso se ejecuta por detrás, sin que tu seas consciente de lo que está sucediendo.

UD3: OPERATING SYSTEMS

CFGM DAW
DPT INF

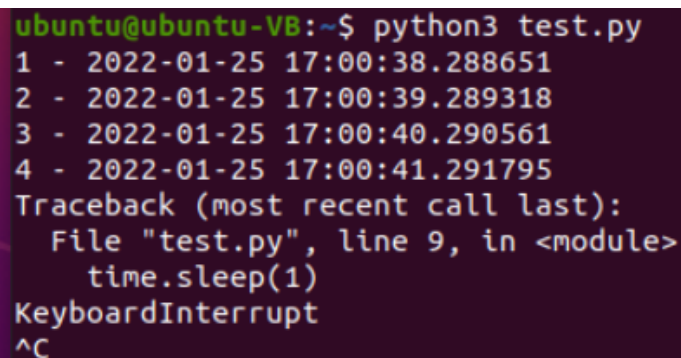
Vamos a probar con un ejemplo, creadon el fichero **test.py** (un sencillo script en Python) con el contenido siguiente:

```
#!/usr/bin/env python3
import time
import datetime

counter = 0
while True:
    counter += 1
    print('{0} - {1}'.format(counter, datetime.datetime.now()))
    time.sleep(1)
```

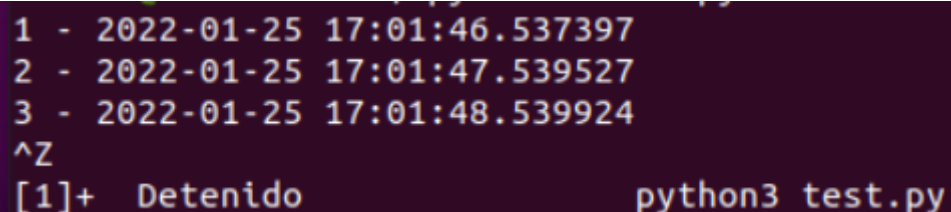
Para ejecutar este script solo tienes que ejecutar esta instrucción en el terminal **python3 test.py**. Lo único que hace este script es imprimir un número consecutivo cada segundo seguido de la fecha y hora a la que lo ha hecho. Esto de forma ininterrumpida hasta que lo detengas.

Si lanzas este proceso de la forma habitual observarás que no puedes continuar utilizando el terminal. No lo puedes continuar utilizando porque este *script* en Python se ha apoderado del terminal. Hasta que no utilices el atajo de teclado **Ctrl+C** no se detendrá.



```
ubuntu@ubuntu-VB:~$ python3 test.py
1 - 2022-01-25 17:00:38.288651
2 - 2022-01-25 17:00:39.289318
3 - 2022-01-25 17:00:40.290561
4 - 2022-01-25 17:00:41.291795
Traceback (most recent call last):
  File "test.py", line 9, in <module>
    time.sleep(1)
KeyboardInterrupt
^C
```

Sin embargo, en lugar de utilizar **Ctrl+C** prueba a utilizar **Ctrl+Z**. Lo que hace este atajo de teclado es enviar el proceso a segundo plano. Sin embargo ha detenido el proceso, y aunque está en segundo plano, no está trabajando para ti, realmente está detenido. Fíjate lo que se muestra en el terminal.



```
1 - 2022-01-25 17:01:46.537397
2 - 2022-01-25 17:01:47.539527
3 - 2022-01-25 17:01:48.539924
^Z
[1]+  Detenido                  python3 test.py
```


UD3: OPERATING SYSTEMS

CFGM DAW
DPT INF

Para saber los procesos que tienes en segundo plano tienes que utilizar el comando **jobs**

Pero lo tienes en segundo plano, ¿cómo lo recuperamos? Primero utiliza **jobs** para saber el número de proceso, y posteriormente **fg %** seguido del número de proceso. Fíjate,

```
ubuntu@ubuntu-VB:~$ fg %1
python3 test.py
4 - 2022-01-25 17:02:42.380047
5 - 2022-01-25 17:02:43.381774
6 - 2022-01-25 17:02:44.383594
7 - 2022-01-25 17:02:45.383976
^Z
[1]+  Detenido                  python3 test.py
```

Probablemente te estés preguntando como puedes lanzar el proceso para que se continúe ejecutando. Sencillo, para iniciar un proceso directamente en segundo plano solo tienes que añadir **&** al final de la instrucción. Por ejemplo, para el caso anterior:

```
python3 test.py &
```

<https://atareao.es/como/procesos-en-segundo-plano-en-linux/#:~:text=traer%20el%20proceso%20a%20primer.segundo%20plano%20con%20bg%20%252%20.>

El **aspcars** es el carácter que indica la ejecución en segundo plano. Si no lo ponemos, estaremos ejecutando en primer plano.

COMANDOS:

ps/pstree. Ambos se encargan de visualizar información de todos los procesos que se están ejecutando en el sistema. **Pstree** muestra información en forma de árbol de procesos. Ej: **ps**; **ps -all**
top. Muestra un listado de procesos en ejecución, identificando memoria que se está ocupando, CPU, etc.

kill. Comando usado para “matar” procesos, es decir, acaba con la ejecución de un proceso. El comando **kill** debe saber qué proceso terminar, con lo que irá acompañado del PID del mismo.

Ej: **kill -9 1234**

En ocasiones podemos usar **killX**, que permite que una vez ejecutado, se haga clic sobre alguna de las aplicaciones abiertas, provocando el cierre o fin de ejecución del proceso.

Jobs. Visualiza aquellos procesos que se estén ejecutando en segundo plano.

DEMONIOS:

UD3: OPERATING SYSTEMS

CFGM DAW
DPT INF

Son procesos que están cargados en memoria, ejecutándose en segundo plano y continuamente a la espera para ofrecer un determinado servicio. Por ejemplo: servidores http, de correo electrónico, etc.

Comandos de control del sistema

halt. Apaga el equipo.

reboot. Resetea el sistema

shutdown -h now. Indica que el sistema debe ser apagado en ese momento.

shutdown -r now. Ordena que el sistema debe ser reiniciado en ese mismo instante.

8. Herramienta Hardinfo (System Profiler and Benchmark)

Instala la herramienta y ejecutala comprobando la información que proporciona.

Accede a la zona de benchmark en hardinfo y haz clic en algunos de los algoritmos de rendimiento que se proponen.

Sigue los pasos indicados en la página siguiente:

<https://www.solvetic.com/tutoriales/article/8836-ver-informacion-hardware-linux-hardinfo-herramienta-informacion-sistema/>