

Introducción MS-DOS y PowerShell

Interprete de comandos

- MS-DOS.
- PowerShell:
 - Se usa a nivel WorkStation y Server, tambien en la nube.
 - Mas potencia que las herramientas gráficas.
 - Potente lenguaje de Scripts orientados a objetos basado en el Framework **.NET**
 - Los comandos se llaman “**cmdlets**”
 - Es **OpenSource**, es codigo abierto,

Lo primero que encontramos es el **prompt** que nos indica la ruta donde nos ubicamos.

Observaciones importantes

Windows es un sistema **case insensitive** (no diferencia entre mayúsculas y minúsculas), en cambio Linux es un sistema **sensitive** (diferencia entre mayúsculas y minúsculas).

Antiguamente los nombres de archivos no podían tener mas de 8 caracteres , 1 punto y 3 caracteres para extensión. Actualmente estas restricciones se han rebajado, pero siguen teniendo unas normas:

- No se permiten caracteres especiales.
- La extensión no es obligatoria pero es recomendable para identificar el tipo de fichero.

Ayuda del sistema

MS-DOS

Comando	Utilidad
Help [comando]	Muestra funcionalidad de todos los comandos, pero si introducimos un comando nos muestra información sobre él.
exit	Cierra la ventana

PowerShell

Cmdlet	Utilidad
Get-Help * o [comando]	Muestra funcionalidad de todos los comandos, pero si introducimos un comando nos muestra información sobre él. Si añadimos el parámetro -Full mostrará documentación completa
Update-Help	Actualiza la ayuda de los comandos.
Exit	Cierra la ventana

Los arboles de directorios

Windows estructura su sistema de ficheros mediante arboles. Cada árbol comienza en la raíz de una unidad. Una unidad es la ubicación de un almacenamiento de datos.

- **A:** Normalmente se usa para disquetes.
- **B:** Normalmente se usa para disco duros
- **C:** Normalmente se usa para unidades ópticas.

Dentro de cada directorio hay dos ficheros especiales:

- `.` : es un enlace al propio directorio
- `..` : es un enlace al directorio **superior** / **padre**

MS-DOS

Comando	Utilidad
<code>dir [directorio]</code>	Muestra el contenido del directorio
<code>cd directorio</code>	Cambia de directorio
<code><UNIDAD>:</code>	Cambia de unidad.

PowerShell

Cmdlet	Utilidad
<code>GET-PSDrive</code>	Muestra las unidades.
<code>Get-ChildItem [directorio] o gci</code>	Muestra contenido de un directorio
<code>Set-location directorio o sl</code>	Cambia de directorio
<code><UNIDAD>:</code>	Cambia de unidad

Con el comando `Get-ChildItem` podemos observar una columna llamada **Mode**. La columna contiene caracteres y representa lo siguiente:

Mode	Descripción
d	Es un directorio
a	Es un archivo
r	Acceso unicamente de lectura
h	Es un archivo oculto
s	Es un objeto del sistema
l	Es un acceso directo

La consola tiene mecanismo para filtrar los archivos que muestra:

- `?` representa ningún o 1 carácter.
- `*` representa 0 o n caracteres.

El comando **tree** muestra el contenido de los directorios y subdirectorios.

Administración de ficheros y directorios

MS-DOS

Comando	Utilidad
del fichero	Elimina un fichero
mkdir directorio	Crea un directorio
rmdir directorio	Elimina un directorio si está vacío. Con el parámetro /s permite eliminarlo.
copy origen destino	Duplica un fichero o directorio
rename origen destino	Cambia el nombre de un fichero o directorio
move origen destino	Mueve un fichero o directorio
type fichero	Muestra el contenido de un fichero
cls	Limpia la pantalla

PowerShell

Cmdlet	Utilidad
Remove-Item fichero	Elimina un fichero
New-item fichero -ItemType File	Crea un fichero
New-item fichero -ItemType Directory / mkdir	Crea un nuevo directorio
Remove-Item fichero / rmdir	Elimina un directorio
Copy-Item -Path -Destination	Duplica un fichero o directorio
Rename-Item -Path -NewName	Cambia el nombre de un fichero o directorio
Move-Item -Path -Destination	Mueve un fichero o directorio
Get-Content -Path / cat	Muestra el contenido de un fichero
Set-Content -Path -Value	Escribe el contenido en un fichero de texto
cls	Limpia la pantalla

Introducción a Bash

Entrada y salida del sistema

En la terminal aparecerá el **prompt**, que según el tipo de usuario será:

- **Administrador (root) :** #
- **Usuario :** \$

Archivos y directorios

Hay reglas para los nombres de archivo:

- El único carácter que no se puede usar es / aunque hay otros que no se recomienda su uso.
- El sistema es case sensitive, diferencia entre mayúsculas y minúsculas.

El sistema de ficheros está formado por un único árbol, y todas las unidades se integran a él.

El comando **whoami** muestra el usuario actual.

El comando **hostname** muestra el nombre de la maquina.

El comando **uname** imprime información del sistema.

El comando **uptime** muestra el tiempo que el sistema lleva arrancado.

Complementos del interprete de Unix

Completar ordenes al pulsar el tabulador.

Acceder al historial de comandos mediante las flechas o con el comando **history**.

- El comando **!!** ejecuta el ultimo comando del historial
- El comando **!N** ejecuta el comando N del historial.
- El comando **!!Texto** ejecuta el ultimo comando que contenga el texto indicado.

Esta información se almacena en el archivo oculto **.bash_history**.

Comandos relacionados con ficheros y directorios

Comando	Utilidad
pwd	Muestra la ruta del directorio donde te encuentras
ls [directorio]	Muestra el contenido de un directorio. -l muestra más información. -S ordena por tamaño. -a para mostrar archivos ocultos
cd [directorio]	Cambia de directorio
rm fichero	Elimina un fichero
mkdir fichero	Crea un directorio
rmdir fichero	Elimina un directorio si está vacío. -r para eliminarlo lleno.
cp origen destino	Duplica un fichero o directorio. -R para copiar una estructura completa
mv origen destino	Mueve un fichero o directorio. También para cambiar el nombre
cat fichero	Muestra el contenido de un fichero. -n enumera todas las líneas incluso si están vacías y -b enumera las líneas con texto. -s comprime las líneas en blanco
tac fichero	Muestra el contenido de un fichero invertido
clear	Limpia la pantalla

Metacaracteres y filtros

Comando	Utilidad
*	Representa una serie de caracteres
?	Representa un carácter
[]	Representa una serie de caracteres dentro de un rango
~	Representa el directorio del usuario.

Configuración de Red

Al usar el comando **ip** los cambios no se guardarían. Para ello tenemos que modificar un fichero que se encuentra en **/etc/netplan** llamado **00-installer-config.yaml**.

Una posible configuración del archivo es:

```
network:
  ethernets:
    enp0s1:
      dhcp4: no
      addresses: [192.168.0.24/24]
      gateway4: 192.168.0.1
      nameservers:
        addresses: [212.166.132.96]
  version: 2
  renderer: NetworkManager
```

Para guardar los cambios tenemos que usar el comando **sudo netplan apply**

Programas y procesos

Un **programa** es una secuencia de instrucciones.

Un **proceso** es un programa en ejecución.

Para gestionar los procesos, el SO:

- Tiene que maximizar el uso del procesador.
- Tiene que tener una política de asignación de recursos.
- Tiene que permitir la comunicación de procesos.

El procesador ejecuta las instrucciones que reside en la memoria principal.

El **planificador de procesos** gestiona la intercalación de procesos:

- Salta un **temporizador** cuando un proceso lleva mucho tiempo ejecutándose (1 quantum)
- El **cambio de contexto** es una acción para pasar de un proceso a otro.
- El **idle** se ejecuta cuando no hay nada en ejecución.

Los procesos tienen prioridad. La prioridad puede ser asignada por el SO o por el propietario. La prioridad no es fija, se puede cambiar.

Los procesos se identifican mediante el PID (Process Identifier). El SO almacena una tabla de procesos.

Una aplicación para crear un proceso necesita que el SO lo cree y lo gestione. Disponemos de 2 llamadas:

- **fork** : Permite que un proceso cree otro proceso.
- **exec** : Permite ejecutar una aplicación externa. Sustituye el proceso en ejecución por otro.

Los procesos son ficheros que podemos encontrar en la carpeta /proc.

Tipo de procesos:

- **Proceso hijo** : Creado por fork.
- **Proceso huérfano** : El padre acaba la ejecución antes que el hijo.
- **Proceso zombie** : Proceso acabado pero espera una respuesta del padre.
- **Proceso parado** : Proceso en pausa.
- **Proceso demonio** : Se ejecuta permanentemente y en segundo plano.

El núcleo (kernel) es el primer proceso que se ejecuta. El gestor de arranque lo ejecuta. **Scheduler** (planificador de procesos) es el proceso del núcleo que gestiona el resto de procesos.

El **proceso init** es el proceso padre de toda la resta de procesos usuario. PID 1. Si un proceso queda huérfano el init lo adopta.

El **proceso kthreadd** es el proceso padre de los procesos que solo ejecutan código del kernel. PID 2.

El comando pstree permite visualizar la jerarquía de procesos.

Creación de procesos

Linux

- **Primer plano** (fg) es el modo default de ejecución. La orden bloquea el uso del interprete.
- **Segundo plano**: se puede jacer mediante **&** o con la orden **bg**

Windows

- Primer plano es el modo default de ejecución. Se realiza mediante el comando **Start-Process**. **No** bloquean el uso del interprete
- Segundo plano usa la orden **Start-Job**. Ejem: Start-Job {programa}.

Control y Monitorización de tareas

Linux

- ps : Muestra la tabla de procesos del terminal y usuario actual.
- jobs : Muestra procesos que se están ejecutando en el interprete de comandos.
- top: Muestra los procesos ordenados por el uso de CPU.
- fg: Cambia un proceso de segundo plano a primer plano.
- bg : Cambia un proceso de primer plano a segundo.

Windows

- Get-Process: Muestra la tabla de procesos de la terminal y usuario actual.
- Get-Job: Muestra los procesos que se están ejecutando en segundo plano.
- Get-Process | Sort-Object cpu -Descending: Muestra los procesos ordenados por el uso de CPU

Matar procesos

Linux

Para acabar un proceso en primer plano:

- Ctrl+c.
- Ctrl+\.
- Comando **kill**.
- Ctrl+Z. Para el proceso. No lo mata.

Para acabar un proceso en segundo plano tenemos que usar el comando **kill**.8

- kill -9 PID (Mata un proceso mediante el PID)

- `kill -9 -1` (Mata todos los procesos)
- `killall nombre` (Mata un proceso mediante el nombre)

Windows

Usamos el comando **Stop-Process**.

- `Stop-Process -id id` (Mata un proceso mediante el PID)
- `Stop-Process -Name "nombre"` (Mata un proceso mediante el nombre)

Prioridades de procesos

Linux

La prioridad se indica mediante un rango: -20 a 19

El valor mas alto de prioridad es el -20 y el más bajo es el 19. Por defecto la prioridad es 0.

Los procesos con prioridad alta usan más recursos y los de prioridad baja cuando no se ejecute nada

El propietario del proceso solo puede modificar la prioridad en un intervalo de 0 a 19, pero el root puede cambiarlo por cualquiera.

Para ejecutar un programa con una prioridad: **nice -n [-20-19] proceso**

Para modificar la prioridad de un proceso en ejecución:

- **renice [-20-19] PID** (Modificar prioridad mediante el PID)
- **renice [-20-19] -u user** (Modificar prioridad de procesos de un usuario)

Windows

Para consultar la prioridad usaremos: **Get-Process | Format-Table -View priority**. La prioridad puede ser alta, normal o baja.

Concatenación de ordenes

Tenemos diferentes operadores para ejecutar múltiples ordenes:

- **Operador AND(&&):** Se ejecuta el segundo comando si el primero se ejecuta.
- **Operador OR(||):** Se ejecuta el segundo comando si el primero no se ejecuta.
- **Lista de ordenes(;):** Para ejecutar varias ordenes de forma consecutiva.

Flujo de datos

El flujo de datos (stream) es un conjunto de información que se desplaza de un origen a un destino. Permite realizar tareas complejas combinando programas sencillos. El SO los trata como si fueran ficheros.

3 flujos de datos estándar:

- **Entrada estándar (stdin):** Es el flujo de entrada de ordenes y aplicaciones.
- **Salida estándar (stdout):** Es el flujo de salida de ordenes y aplicaciones.
- **Error estándar (stderr):** Es el flujo de salida para mensajes de error.

Filosofía de Linux

- Programas pequeños con una sola utilidad

- Las operaciones complejas se realizan combinando programas.
- Se trabaja con flujos de datos de texto.

Redirecciones

Permiten modificar los flujos de datos estándar. Podemos modificar la entrada o salida estándar que usa una orden o aplicación. Redireccionar la salida de una orden hacia un fichero.

Operadores de redirección

Operador	Descripción
>	Crea un fichero conteniendo la salida estándar del comando ejecutado. Sobrescribe si contiene cosas.
>>	Crea un fichero conteniendo la salida estándar del comando ejecutado. Añade si contiene cosas.
2>	Crea un fichero conteniendo la salida estándar de error del comando ejecutado. Sobrescribe si contiene cosas.
2>>	Crea un fichero conteniendo la salida estándar de error del comando ejecutado. Añade si contiene cosas.
&>	Crea un fichero conteniendo la salida estándar y la salida estándar de error del comando ejecutado. Sobrescribe si contiene cosas.
<	Envía los datos del fichero de la derecha como entrada estándar al comando de la izquierda
<<	Acepta un texto de varias líneas como entrada estándar.
<>	El fichero especificado a la derecha es la entrada estándar y la salida estándar del comando de la izquierda.

Hay un fichero para silenciar la salida de un comando : **/dev/null** . En Windows usamos el comando **Out-Null**.

Conductos

Los conductos son un mecanismo que permite enlazar la salida estándar de una orden como entrada estándar de otra orden.

Comandos

-tee : Copia su entrada estándar sobre su salida estándar, y sobrescribe los ficheros.

-echo : Permite escribir por la salida estándar.

-sleep : Parar la ejecución durante un tiempo. En Windows: **Start-Sleep**

-cat : Concatena archivos.

-tac : Concatena pero de forma inversa.

-Paginadores:

- more.
- less.
- pg.

-head : Muestra las primeras 10 líneas de un fichero. Con -n podemos indicar el número de líneas. En Windows: **select-object -first numero** o **get-content texto -Head numero**.

-tail : Muestra las ultimas 10 líneas de un fichero. Con **-n** podemos indicar el número de líneas. En Windows: **select-object -last numero o get-content texto -Tail numero**.

-sort : Ordena las líneas de fichero. Con **-n** lo ordenará de forma numérica, con **-r** de forma inversa, con **-k** permite elegir una columna , con **-t"delimitador"** permite seleccionar el delimitador para el **-k**.

En Windows: **Get-ChildItem | sort-object [-Property propiedad] [-descending]**

-wc : Permite ver por defecto el numero de líneas(-l) , palabras(-w) y bytes (-c) de un fichero. En Windows: **Measure-Object**

-nl : Enumera las líneas de un fichero. En windows: **Select Linenumber, Line**

-pr : Prepara un fichero para su impresión.

-fmt : Formatea líneas largas, según la mida que se indique. Ejem : **ls -l | fmt -w 10**.

-uniq : Compacta en una única línea todas las líneas repetidas consecutivas. En windows : **Get-Content texto | get-unique**

-split : Permite dividir un fichero. Ls -l | split -l "Por cada n líneas realiza el split " - nombre

-cut : Selecciona una porción de las líneas de un fichero de texto. Con **-d** permite elegir el delimitador por el que se corta junto a **-f** para seleccionar la zona, podemos cambiar el delimitador con **-output-delimiter="delimitador"**, con **-c** podemos elegir los caracteres (**-c11-19**).

-tr : Permite sustituir caracteres, eliminar caracteres, comprimir caracteres repetidos.

-grep : Muestra las líneas que contengan x cadena de caracteres. Con **^** los que comiencen, con **-c** cuenta el numero, con **.** es cualquier carácter, con **\$** los que acaben, con **-v** los que no contengan, con **[]** definimos un rango de valores. En Windows : **Select-String -Pattern "texto"** . Con **-n** no contienen.

Usuarios y Grupos

Los sistemas actuales son multiusuario y multitarea. El SO tiene que:

- Gestionar la entrada y identificación de los usuarios.
- Controla que programa pueda ejecutar cada usuario o grupo de usuarios.
- Protege los ficheros de un usuario determinado.

El sistema de contraseñas de Unix

La contraseña nunca se almacena como texto: se guarda cifrada. Cuando colocas la contraseña se cifra y se compara con la cifrada. No se puede obtener la clave original a partir de la clave cifrada.

Consejos para elegir contraseña segura:

- No dejarla vacía o usar el nombre de usuario.
- No elegir palabras, ni nombres propios o de localidades.
- Evitar secuencias, frases echas,etc
- No usar combinaciones o permutaciones de palabras
- Combinar minúsculas, mayúsculas, símbolos, etc.
- Usar 8 caracteres o más.

Grupos de Usuarios

Conjunto de usuarios con unas características similares. Permite dar permisos a varios usuarios al mismo tiempo.

Superusuario

El superusuario (root) es una cuenta que tiene privilegios máximos que le permiten realizar cualquier acción. Se encarga de la administración del sistema. Esta cuenta no se tiene que usar para trabajar de forma habitual en el sistema.

Otras cuentas de usuarios especiales serían:

- **nobody**: Esta cuenta se encarga de correr los servicios del sistema.
- **bin**: Esta cuenta se usa para proteger los binarios esenciales del sistema.

Ficheros de usuarios y grupos

Toda la información de usuarios y grupos se almacenan en ficheros:

- **/etc/passwd** : información (nombre, directorio home, etc) de usuario.
- **/etc/group** : información sobre los grupos de usuario.
- **/etc/shadow** : contraseñas cifradas de los usuarios y configuración de la misma.

Estos ficheros están organizados por líneas, cada una de las cuales se identifica un usuario o grupo.

Fichero /etc/passwd

login:contraseña:UserID:GroupID:comentarios:Directorio:Interprete

1. **Login**: El nombre de usuario. No pueden existir dos iguales.
2. **Contraseña cifrada**: Si se usa el fichero shadow, se muestra una x. En otro caso se almacena el segundo campo.
3. **User ID**: Número de identificación del usuario. El 0 es el root y los usuarios convencionales comienzan con 1000.
4. **Group ID**: Número de identificación del grupo principal.
5. **Comentarios**: Campo opcional que almacena información adicional: nombre completo de usuario, dirección, teléfono, ...
6. **Directorio del usuario**: ruta completa del directorio home del usuario.
7. **Interprete de comandos utilizado**: Es el interprete por defecto del usuario. Si indica /bin/false quiere decir que el usuario no puede ejecutar ningún comando.

Fichero /etc/group

nombre:contraseña:GroupID:Usuarios

1. **Nombre del grupo**: El nombre de grupo. No pueden existir dos iguales, pero puede coincidir con el nombre de usuario.
2. **Contraseña cifrada**.
3. **Group ID**: Número que identifica unívocamente los grupos. El 0 se reserva para el grupo root.
4. **Usuarios** : Campo opcional donde pueden aparecer solo nombres de usuarios que pertenecen al grupo de forma secundaria.

Fichero /etc/shadow

Login:contraseña:díasUltimoCambio:díasPoderCambiar:díasTenerCambiar:díasAvisoCaduca:díasAntesDesh:díasDesh:reservado

- **Login** : El nombre de usuario. Tiene que ser el mismo que a passwd.
- **Contraseña cifrada.**
- **Días que han pasado**, desde el 1 de enero de 1970, hasta que la contraseña ha sido cambiada por ultima vez.
- **Días que han pasado hasta que la contraseña se pueda cambiar.**
- **Días que han pasado** hasta que la contraseña se tenga que cambiar.
- **Días antes** de que caduque la contraseña para avisar al usuario que la tenga que cambiar.
- Opcionales:
 - Días que pueden pasar después que la contraseña caduque, antes de deshabilitar
 - Días, desde el 1 de enero de 1970, desde que la cuenta se deshabilita.
 - Campo reservado para posibles usos futuros

Operaciones básicas

-who / w : Permite ver que usuarios hay conectados.

-whoami : Muestra nuestro nombre de usuario

-groups : Muestra los nombres de los grupos que pertenecemos.

-id : Muestra información de los dos comandos anteriores, junto con el UID y GID.

-su login: Permite acceder al sistema con otro usuario, sin tener que cerrar sesión.

-newgrp : Permite acceder al sistema con los privilegios de un grupo.

-sudo : Permite ejecutar una orden con los privilegios de otro usuario. No se indica será el root.

-sg : Permite ejecutar una orden con los privilegios de un grupo.

Añadir usuarios

-adduser : Permite añadir un nuevo usuario al sistema de forma interactiva.

-useradd : Permite crear un usuario o cambiar la configuración por defecto.

Ejem: **useradd -m -g nom-grupo -G grup1,grup2,... -d directorioHome -p contraseña nombreUser**

Editar información de usuario

-usermod : Permite modificar el directorio, el shell, la expiración de contraseña, ...

-chfn : Cambia la información personal del usuario, el campo de comentarios del fichero passwd.

-chsh : Cambia el shell del usuario.

-passwd : Para cambiar la contraseña de un usuario, información de expiración o para bloquear/desblo cuentas.

-chage : Permite cambiar la información de expiración de una cuenta.

Eliminar usuarios

deluser (se borra o guarda los ficheros según los parámetros que le pasemos,etc) o **userdel**

Operaciones relacionadas con grupos

- **addgroup** o **groupadd**: Permite añadir un grupo al sistema.
- **groupmod** : Nos permite modificar la información (nombre y GID) de un grupo determinado.
- **delgroup** o **groupdel** : Elimina un grupo determinado. Si es primario de alguien no se puede eliminar.
- **gpasswd** : Para cambiar la contraseña del grupo.

Gestión de usuarios de Windows

- **net localgroup** : Consulta la lista de grupos locales.
- **net localgroup nombre /add** : Añade un grupo.
- **net localgroup nombre** : Muestra los usuarios asociados a un grupo.
- **net localgroup nombreGrupo nombreUsuario /add** : Añade un usuario al grupo.
- **net localgroup nombreGrupo nombreUsuario /delete** : elimina un usuario de un grupo.
- **net localgroup nombreGrupo /delete** : Elimina un grupo.

Organización y acceso a ficheros

El sistema de ficheros es la parte del SO encargada de la administración de los datos a los dispositivos de almacenamiento secundario.

Características del sistema de ficheros de Linux

- Ficheros organizados por directorios.
- **Sistema Virtual de ficheros**. Estructura jerárquica con un único directorio raíz /.
- Cada directorio tiene 2 ficheros especiales: . y .. (referencia al propio directorio y al directorio padre)
- El nombre puede tener 255 caracteres.
 - Case sensitive.
 - Los que comienzan por punto son ocultos.
 - Todos los caracteres son validas excepto /. Aunque no se recomienda usar caracteres especiales.
 - La extensión no es obligatoria y no determina el tipo de fichero.
- En Linux todo son ficheros: Los ficheros, directorios, dispositivos, enlaces, etc.
- Los ficheros se representan por el camino que lleva a ellos, puede ser absoluto o relativo.
 - Camino absoluto: Ruta desde la raíz. Comienza siempre con /.
 - Camino relativo: Ruta desde el lugar donde nos encontramos. No comienza con /.
- Internamente trabaja con el concepto de i-nodo.

File Hierarchy Standard (FHS)

Es un estándar que sigue el sistema de ficheros de Linux, que define los directorios principales y sus contenidos.

Objetivo del FHS:

- Permite al programa predecir la localización de ficheros y directorios a instalar.
- Permite a los usuarios predecir donde el programa instalará los ficheros y los directorios.
- Las aplicaciones, distribuciones y sistemas operativos que siguen el estándar son llamados “FHS compliant”.

Jerarquía de ficheros

- **Ejecutables** (binario):
 - **/bin** : Ejecutables básicos para el funcionamiento del sistema.
 - **/sbin** : Ejecutables del superusuario.
 - **/usr/bin** y **/usr/sbin** : Ejecutables una vez se conecta el usuario al sistema, la mayor parte del tiempo es una copia de **/bin** y **/sbin**.
- **Librerías**:
 - **/lib** : Librerías principales de los ejecutables principales.
 - **/usr/lib** : Librerías secundarias del resto de ejecutables.
- **Homes** (carpetas de usuario):
 - **/home** : Contiene los directorios personales de los usuarios de la máquina.
- **Ficheros de configuración**:
 - **/etc** : Contiene los ficheros de configuración y scripts de arranque del sistema.
 - **/etc/skel** : Contiene los archivos que se copiarán en el directorio del usuario al crear una cuenta.
- **/usr**
 - Jerarquía secundaria donde se almacena las aplicaciones y utilidades multiusuario, compartidos con todos los usuarios.
- **/var**
 - Se encuentran todos los archivos de datos variables con el tiempo.
- **/mnt**
 - Puntos de montaje temporales.
- **/media**
 - Puntos de montaje para dispositivos re movibles CD, DVD, USB.
- **/dev**
 - Contiene archivos como **/dev/cdrom** para representar la unidad de CD/DVD.
 - **/dev/null** es como una papelera.
- **/root**
 - Home del superusuario
- **/tmp**
 - Se usa para almacenar los archivos temporales.

- **/lost+found**
 - Se crea automáticamente al arranque del sistema. Usa la herramienta de verificación fsck para guardar los archivos recuperados después de un accidente.
- **/proc**
 - Sistema de archivos virtuales que representa el estado del sistema. Solo existe en la memoria RAM (Contiene procesos)
- **/boot**
 - Contiene el núcleo de Linux y otros archivos que se ejecutan durante el arranque del sistema.
- **/opt**
 - Contiene paquetes de programas opcionales.

Estructura de Directorios Windows

- **\PerfLogs**
 - Puede contener ficheros de rendimiento (log) asociados al sistema operativo.
- **\Program Files**
 - Las aplicaciones instaladas se alojan en este directorio.
- **\Program Files (x86)**
 - Aparece en las ediciones de Windows de 64 bits.
- **\Program Data (Hidden)**
 - Contiene datos necesarios para ejecutar aplicaciones.
- **\Users**
 - Esta carpeta contiene una subcarpeta para cada usuario que ha iniciado sesión al sistema.
 - Contiene dos carpetas más:
 - Public: Para compartir ficheros.
 - Default: Contiene los perfiles de los usuarios.
 - [username]\AppData : Almacena los datos y la configuración de las aplicaciones para cada usuario.
- **\Windows**
 - Es donde se aloja la instalación del SO.
 - \System
 - \System2
 - SysWOW64

Sistemas de ficheros

Distribución de un disco duro (MBR)

- Primer pista del disco duro:
 - Master Boot Record (MBR): Sector 512 bytes.

- 446 bytes para bootloader.
- 64 bytes para la tabla de particiones.
- 2 bytes para un código único.
- Resto de pistas:
 - Particiones: Divisiones físicas del disco duro para separar tipos de sistemas de ficheros o organizar el espacio. Cada partición tiene su propio sistema de fichero.

Ventajas del uso de particiones

- Soporte de múltiples sistemas operativos.
- Usar múltiples sistemas de ficheros.
- Gestión del espacio del disco.
- Protección contra errores.
- Seguridad.
- Copia de seguridad.

Sistema de particiones(MBR)

- Particiones primarias:
 - Se pueden formatear lógicamente y marcar como arrancables.
 - En Linux van enumeradas del 1 al 4.
- Particiones ampliadas:
 - No se pueden formatear ni marcar como arrancables.
 - Son contenedores de particiones lógicas.
 - Solo hay 1
- Particiones lógicas
 - Se pueden formatear y marcar como arrancables
 - En Linux comienza por 5.

Sistema de Particiones (GPT)

Mejoras respecto MBR:

- GPT solo funciona en SO de 64 bits.
- GPT maneja discos 256TB. MBR de 2 TB.
- GPT admite 128 particiones primarias, MBR solo 4.

Espacio de intercambio (swap)

Partición usada para guardar procesos que no podemos colocar en la RAM. (Memoria Virtual)

Formatación de dispositivos

Formatación a nivel físico:

- Formateado a bajo nivel.
- Formateado a alto nivel.

Formatación a nivel lógico:

- Asigna el sistema de ficheros.

Manipulación de la tabla de particiones

Linux

Fdisk : Permite modificar la tabla de particiones.

Windows

- **Get-Disk**: Lista los discos activos.
- **Get-Partition -DiskNumber n** : Lista las particiones del disco número n.
- **Clear-Disk n -RemoveData** : Elimina los datos del disco n.
- **Initialize-disk -number n -PartitionStyle tipo** : Inicializa el disco n.
- **New-partition -DiskNumber n -Size tamaño -AssignDriveLetter**: Crea una nueva partición al disco n, del tamaño y se le asigna una letra.
- **Get-Volume -DriveLetter Letra** : Lista la partición con la letra.
- **Format-Volume -DriveLetter Letra -FileSystem tipo**: Formatea la partición asignando un sistema de ficheros.
- **Remove-Partition -DriveLetter Letra** : Elimina la partición con letra.

Tipos de sistema de ficheros

- Ext2/Ext3 /Ext4
- FAT
- NTFS
- ReFS

Crear sistema de ficheros

- Mkfs. Ejem: mkfs.ext3 /dev/sda1
- mkswap permite crear particiones de intercambio (no se montan: swapon) Ejem: mkswap /dev/sda5

Comando mount

Consultar información sobre dispositivos montados:

- mount.
- /etc/mtab.
- /proc/mounts.

Montaje de dispositivos:

mount /dev/sda5 /media/disc5

umount es la orden contraria. Ejem: umount /dev/sda5

Fichero /etc/fstab

Configuración de puntos de montaje.

“Dispositivo” “Punto de montaje” “Tipo de Sistema” “Opciones” “0 o 1 copias” “0 o 1”

UUID

Para calcular el UUID usamos el **blkid**.

Comprobación del uso del disco

- du : Espacio que ocupa en disco un fichero.

- df : Estudio del espacio usado.

Comprobación y reparación de errores

Linux

- **fsck** : Permite comprobar y reparar sistemas de ficheros Linux.

Windows

Repair-Volume