**PROCESOS:**

**Los procesos que no podemos controlar**, son los programas que permiten el funcionamiento del sistema operativo. Se

llaman demonios. Dependen de otros procesos.

Todo inicia en la memoria donde se guarda la entrada y salida de datos de los puertos en la computadora.

**BIOS:** Entrada y salida estandar que tiene que tener cualquier computadora para que todos los sistemas operativos puedan utilizarlo. Genérico, para poder compartir. Es la primera pared de abstracción. Permite abstraerse del hardware.

El primer proceso que se va a cargar en la memoria **ram** es la **BIOS**. Se tiene en memoria **RAM**.

El segundo proceso es el **bootloader**. Permite seleccionar con qué sistema operativo quiero iniciar. De acá se puede

manipular las direcciones donde voy a iniciar mi máquina.

**Magicnumber:** Son 64 bites.

El tercero es el **Kernel** (escrito en C): Cuando levanta, deja todos los procesos abiertos. Administra como los programas

acceden al hardware. Que informacion se va a cargar, y como se va a manipular la momoria ram, el procesador, el hardware

y el software.

En windows -> Mas funcionalidad grafica. //// En linux -> Mas funcionalidad con comandos.

El cuarto es **INIT**: Main en C. Lo que hace es crear procesos. Estos procesos pueden tener una funcionalidad determinada por un tiempo. Ademas hay procesos que son interactivos, se realiza una accion y se genera una respuesta.

Cada uno de estos procesos, tiene una direccion de memoria. (Una memoria reservada). **Proceso zombie:** Es un proceso el cual ya no tiene direccion de memoria. No está ni el proceso padre, ni el proceso hijo. Solamente hay una posicion de memoria reservada.

**COMANDOS:**

**TOP:** Muestra los procesos.

**PID:** **ID** DEL PROCESO. **USUARIO**: Quien creo el proceso **PR**: Prioridad del proceso **RT**: 1-99 prioridad dada por otro proceso.

**Salir:** CRTL + Z

**ls -lth | more**

**BG:** Llevar o traer procesos del background. Poner procesos en pausa BAAR PROCESO

**FG:** Toma el control segun el proceso

**JOBS:** Los que estan corriendo.

**KILL PID**

**KILL ALL**

**PS -EF** Muestra los procesos en consola.

**PROCESOS:**

**R**->Run

**S**->Sleep

**Z**->Zombie

**Swap** -> No esta en memoria.

**Stoped** -> No lo veo.

**Matar zombie. Encontrar, y KILL PDI**

Si se le pone un & adelante se ejecuta en segundo plano.

**USUARIOS - PERMISOS - MEMORIAS - S. CONTROL DE VERSIONES**

**PERMISOS:**

**ls -l** Muestra grupo de permisos.

**Los permisos pueden ser: R / W / X**

**RWX** --- ----

**Propietario** **Grupo** **Resto del mundo**.

111 000 000

Dependiendo el permiso, en binario, es el número que se le va a dar

**R W X** 111 = 7

**R W -** 110 = 6

**R - X** 101 = 5

**R- -** 100 = 4

**CHMOD XXX** -> Propietario - Grupo - Resto del mundo.

**CHMOD** -> Es un proceso que va al puntero, toma la primer cadena de bytes, y los modifica para

dar o quitar permisos.

**useradd pepito** Agregar usuario. Tambien se le puede poner una fecha para identificar cuando el usuario dejará de tener permisos. Ademas se le puede agregar el interprete de comandos. **-s** pasándole donde está la nueva consola.

A partir de ahora cada vez que el usuario abre la consola, tendrá la que se le asignó

**sudo passwrd pepito** Pone la contraseña de pepito. Hay que insertarla luego 2 veces más.

**login** Con esto entro al usuario. Hay que poner nombre y contraseña

**userdel pepito** RIP PEPITO LO ACABAS DE MATAR. F

**useradd -m -d /home/Maria Maria** Creas a maria en la carpeta /home/Maria

**usermod** Modificar.

**cat /etc/passwd** Muestra los usuarios y demas cosas.

**sudo useradd -G** **Agrega un grupo** **-G** -> Cambia grupo principal **-g** Agrega grupos secundarios.

**cat /etc/group** Muestra los grupos

**sudo groupadd empleado** Crea un grupo de empleados.

**chgroup** // **chgrp** **Maria** **compartidaa** Comparte a Maria la carpeta compartidaa.

**sudo usermod -g alumno Maria**

**cd**

**cd ./config**

**user.dirs.dirs**

<https://expertosdelinux.com/anadir-usuario-grupo-linux/>

**¿COMO HABILITO Y DESHABILITO?** **sudo passwd -l Maria** // HABILITA **sudo passwd -u Maria** // DESHABILITA

**COMANDOS EN LINUX**

**LS SD\*** Muestra las particiones 🡪(Aca nos damos cuenta por ejemplo que **sda** -> Tiene 3 particiones, y **sdb** -> No tiene ninguna)

**lsblk** Lista los bloques -> Muestra las particiones en forma de árbol.

**sudo fdisk sda** Es un comando para meterse dentro de la partición **sda**. Habrá más opciones dentro.

**sudo fdisk sdb** Es un comando para meterse dentro de la partición **sdb**. Habrá más opciones dentro.

Si apretamos la letra **m -> Nos va a mostrar ayuda**. Si apretamos la letra **n -> Nos da la opción de crear la partición**. Luego de eso, hay que apretar “**enter” (la tecla)** 4 veces. (**n -> Partition Type -> Primary**).

Para guardar todo esto, hay que apretar la letra **W**.

**sudo mkfs sdb1** Guarda la partición en el escritorio. Hay que hacerlo con cada partición.

El siguiente comando tiene su explicación compleja. Su sintaxis es **mount** “” “”. El primer parámetro es la partición, el segundo parámetro es donde se guarda esa partición, o en donde hace referencia.

**Mount /dev/sdb1 /home/alumno/discoUno.** Esto va a hacer que la partición creada **sdb1**, tenga su acceso en el directorio <**discoUno**> que se guardó en “**alumno**”.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**sudo fdisk /dev/sdb Crear la partición sdb.**

Después de esto hay que apretar **(( P ))** – **((N))** (CREAR PARTICION) – **((P))** (PRIMARIA) +2GB

Ahora hay que apretar **(( t ))** – Seleccionar la partición – Poner **(( 8e ))** Y se va a cambiar a **LINUX LVM**.

**AHORA A TRABAJAR CON LVM:**

**Crear volúmenes físicos:**

**pvcreate /dev/sdb1** Crea volumen físico de la partición SDB1

**pvs** Muestra todos los volúmenes.

**Crear un grupo de volúmenes:**

**vgcreate vgapp1 /dev/sdb1** Crea un grupo de volúmenes. A cada **PV** lo asignamos a un **VG**. En caso que se quiera hacer un grupo, con 2 volumenes de distinto disco, el comando sería:

**vgcreate “Nombre” /dev/sbd1 /dev/sdc1**

**Aignamos los volúmenes lógicos:**

**lvcreate -l +100%Free -n lvapp1 vgapp1**  El comando sería: **lvcreate -l +100%Free -n “Nombre” “Nombre del grupo”**

**Ahora damos formato:**

**sudo mkfs.ext4 /dev/mapper/vgapp1-lvapp1**

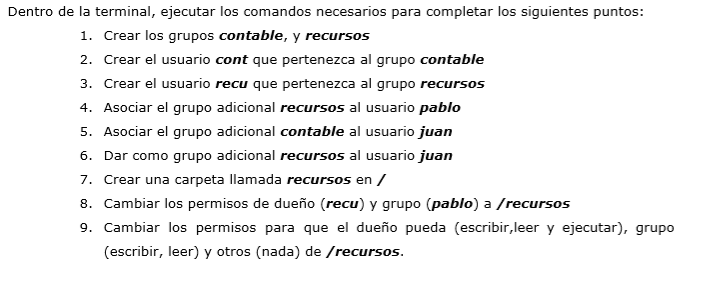
**Ahora realizamos los puntos de montaje:**

**mkdir -p /alumno/home/app.**  Crea el directorio app.

**mount /dev/mapper/vgapp1-lvapp1 /app** Monta el volumen lógico en el directorio app.

**EXTENDER: lvextend-L+1GB/dev/mapper/vgdb-lvdb1**

**RESIZE DEL DISCO: resize2fs/dev/mapper/vgdb-lvdb1**



**¿COMO HACERLO HACERLO?**

**1) Creamos los grupos con:**

1) Crear los grupos contable y recursos.

**sudo groupadd “Nombre del grupo”**

sudo groupadd contable

sudo groupadd recursos

2 – 3 ) Crear el usuario cont, grupo contable. Y el usuario recu, grupo recursos.

**sudo usermod -a -G “Nombre del grupo” “Nombre del usuario”**

sudo usermod -a -G contable cont

sudo usermod -a -G recursos recu

4) Asociar el grupo adicional recursos al usuario pablo. Asociar el grupo adicional contable al usuario juan.

**sudo useradd -g** **“Nombre del grupo” “Nombre del usuario”**

sudo useradd -g recursos pablo

sudo useradd -g contable juan

6) Dar como grupo adicional recursos al usuario juan.

**useradd -m -g default\_group -G other\_group user\_name**

useradd -m -g contable -G recursos juan

(O puede ser este) sudo usermod -G nuevogrupo usuario

7) Crear una carpeta llamada recursos en /

**mkdir -p /Carpeta.**

8) No se

9) sudo chmod 760 /recursos.