**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**

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