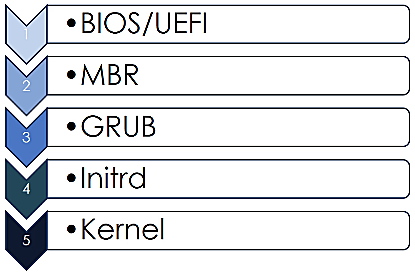
**Sobre el proceso de Arranque de Linux**

Cómo se produce, qué secuencia tiene, etc,...

**A continuación la secuencia de Arranque de un Sistema Operativo Linux:**



**1. Se inicia la BIOS o UEFI (depende de la que tenga seleccionada para trabajar).**

**2. Cargar el MBR (Master Boot Record).**

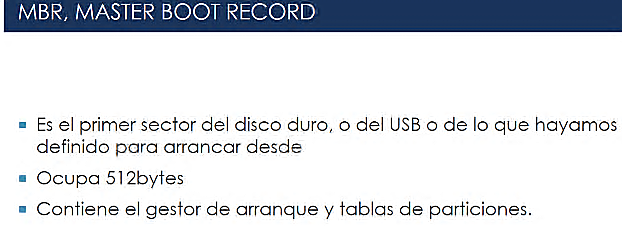
**3. Cargar el GRUB, un gestor de arranque.**

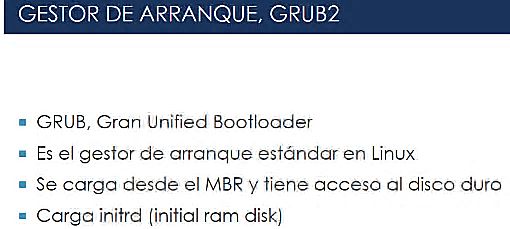
**4. Arranca el Initrd (Initial RAM Disk).**

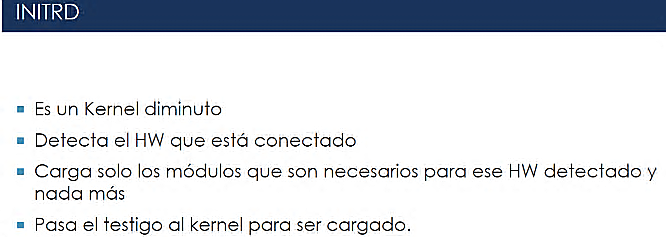
**5. Arranca el Kernel con sus módulos, iniciados por Init; y así, en definitiva, se carga nuestro Linux.**

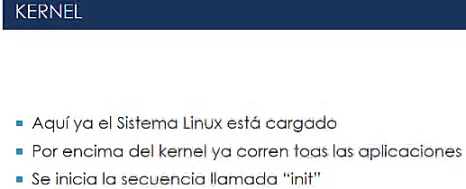
**A modo de resumen:**

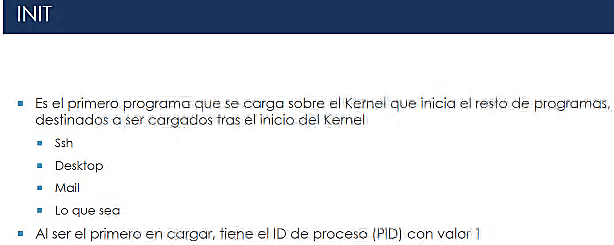






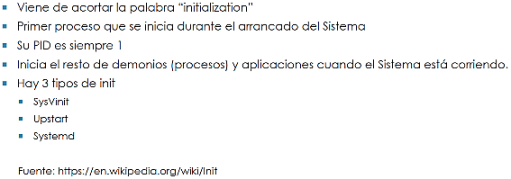






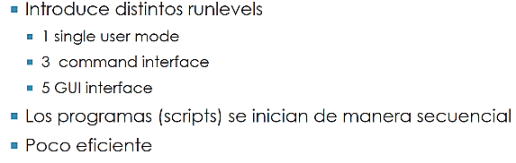
**INIT a profundidad**

**Qué es INIT?**



***Init*** es el primer programa ejecutado; por lo cual, su PID siempre será 1.

**Qué es SYSVINIT?**



Nos dará la facultad de elegir e inicializar nuestros programas en diferentes niveles de ejecución ***(runlevels).***

***Nivel 1:*** Single user mode.

***Nivel 2:*** Command interface (Interfaz solo de comandos).

***Nivel 3:*** GUI interface.

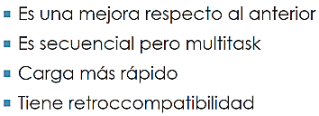
***Hay más niveles, mas éstos son los más usados.***

Adicionalmente, hay que decir que los programas *o scripts* (archivos ejecutables) tienen una enumeración a la hora de ser cargados; es decir, *se inician de manera secuencial* y se ejecutan en orden según de menor a mayor enumeración.En otras palabras, cuánto más baja es la enumaración del programa en cuestión, mayor es la prioridad a la hora de ser inicializado; y, hasta que no finalice la ejecución de un programa, no comienza el programa inmediatamente ulterior.

Lo lamentable es que **SysVinit** es muy poco eficiente por su propia naturaleza y puede dilatar mucho el arranque. Por ejemplo, pongamos el siguiente caso: Si suponemos que tenemos un problema en el programa número 14 y queremos solucionarlo; para solucionarlo, tendríamos que esperar entonces a que todos los programas *o procesos* que van desde el número 1 hasta el número 13 se ejecuten y se cierren, ¿no es eficiente, cierto?

Pues bien, para solucionar ese tipo de problema de eficiencia, la familia Ubuntu ha optado por incorporar otro tipo de init: ***Upstart.***

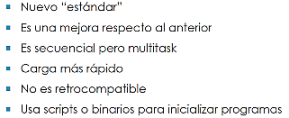
**Qué es Upstart?**



La gran diferencia de ***Upstart*** con relación a ***SysVinit*** es que puede inicializar más de una aplicación a la vez; esto si, entre ellas, no hay interferencias (multitask); en todo caso, su carga es más rápida.

Por último, tenemos el último tipo de init: ***Systemd***

**Qué es Systemd?**



La diferencia de ***Systemd*** con relación a los otros dos tipos de INIT es que ***Systemd*** también puede ejecutar o inicializar su programas a base de binarios (y no sólo con ***Scripts***). Aparte de que acá hay una mayor gestión a la hora de inicializar los procesos.

**Run Levels (Niveles de ejecución)**

**Distintos niveles de ejecución SYSVINIT (7 en concreto).**

**0. Halt:** *Cuando lo ejecutamos le notificamos a nuestro Hardware que detenga todas las funciones de la CPU; por lo que, va a mantener a nuestro servidor o PC en una especie de “estado de coma”. Él seguirá encendido, pero sólamente podrá ejecutar tareas de mantenimiento muy básicas.*

**1. Single user mode:** *Siendo un chico “malo”; con él, por ejemplo, podríamos resetear la contraseña de root; ahora, siendo un chico“bueno” (todo depende de cómo lo vea y bajo qué circunstancias), podríamos resetear la contraseña de root en caso que nosotros mismos seamos el administrador de sistemas y realmente se nos haya olvidado, con genuinidad, nuestra contraseña.*

**2. Multiusuario sin red:** *Esto quiere decir que vamos a poder cargar nuestro sistema operativo sin necesidad de una red. Y de paso, con Multiusuario, aún te puedes logear con muchos usuarios más (por eso se llama multiusuario)*

**3. Multiusuario con red:** *Es exactamente lo mismo que Multiusuario con red. Esto quiere decir que vamos a poder cargar nuestro sistema operativo; pero, ahora, con una red.*

**4. No tiene uso**

**5. Multiuser con GUI:** *Graphic User Interface (GUI), carga lo que sería nuestro entorno de escritorio: escritorios, ventanas, iconos, etc.*

**6. Reiniciar:** *El PC o servidor se va a reiniciar.*

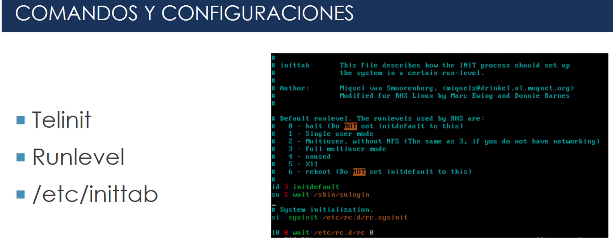
**Cada familia de Linux tiene unos niveles de ejecución diferentes.**

**Es, por ejemplo, diferente los niveles de ejecución de la familia Debian/Ubuntu si los comparamos con la familia Centos/Suse.**





**Ahora, necesitamos saber sobre cómo interactur con los niveles de ejecución (Run Levels) en Consola**



Al ejecutar **Telinit,** por ejemplo, vamos a poder cambiar entre los distintos niveles de ejecución; sin embargo, para poder correr este comando, necesitamos permisos de *root.*

Al ejecutar el 2do comando, **Runlevel,** se nos dice en qué nivel de ejecución estamos. Al igual, el mismo comando, nos refleja cuál era el anterior nivel de ejecución; es decir, cuál era el nivel de ejecución en el que estabamos antes de haber hecho el cambio.

Adicionalmente, algo para agregar: Muchas veces el resultado que arroja este comando, al principio, refleja una N y luego un número (#); pues bien, la ***N*** significa ***None*** (Ninguno) y el significado de ***None*** nos está diciendo que no habíamos corrido ningún nivel de ejecución antes de hacer el cambio; es decir, estaríamos corriendo realmente nuestro primer cambio de nivel de ejecución, y por eso el resultado arroja *N* en ocasiones. Y si por ejemplo aparece en los números un: ***5*** y luego ***3,*** quiere decir entonces que, anteriormente estabamos en el nivel de ejecución ***5) Multiuser con GUI*** y que ahora estamos en el nivel de ejecución ***3) Multiusuario con red*** (para saber de dónde vinimos y hacía dónde vamos).