

Proceso de arranque de un sistema LINUX

```

chip eprom -----> lista de disp.almacenamiento
bios(legacy)          permanentes en busca de un sist.op
UEFI(efi system)      arrancable, si no lo encuentra ERROR
    ||                  |
testea disp.configurados carga en memoria RAM el gestor de arranque
mensajes POST al usuario del s.o q se encuentra en primer sector
                        reservado del disp.almacenamiento

```

ej: disco 5 particiones

```

-----
|X| | win-server | | win-11 | |ubuntu | |ubuntu-server | datos
-|-----
| secund.gest          |secund.gestor
cuerpo ppal            GRUB-stage-2
gestor arranque
GRUB-stage-1

```

una vez cargado el stage-2 del GRUB (gestor de arranque) carga en memoria un fichero INITIAL RAM DISK (initrd), se encarga:

- cargar en memoria los drivers principales de los dispositivos que hay en el equipo para que el kernel los tenga a su disposicion
- carga en memoria el modulo del sistema de gestion de ficheros del sist.operativo (ext4)

```

kernel <----- modulo sistema gestion de ficheros
                    (indica como se debe almacenar/recup
                    informacion del disp.almacenamiento)

```

- carga el kernel (fichero vmlinuz...)
- el kernel una vez cargado en RAM, el modulo de gestion de procesos se ejecuta y genera el PROCESO PRINCIPAL o PROCESO PADRE del sist.op llamado INIT o SYSTEMD, q carga en memoria el RUNLEVEL o modo de ejecucion del sist.op por defecto, q es el 5

existen otros runlevel

```

0 = apagado <----- para cambiar a este runlevel:  init 0
6 = reinicio
1 = sin ent.grafico,monousuario sin red
2,3,4 = sin uso, reservados
5 = con ent.grafico, multiusuario y con red

```

en cada runlevel, el proceso INIT arranca varios servicios definidos en un directorio /etc/rcNUM-RUNLEVEL.d

p.e: para el RUNLEVEL 5 <--- esta el directorio /etc/rc5.d
donde se encuentran los servicios a
arrancar/parar de ese runlevel

```

hay 2 tipos de ficheros:
S_xx_nombre_servicio <--- fich.arranque
                        servicio "start"
K_xx_nombre_servicio <--- fich.parada
                        servicio "kill"

```

la combinacion xx_nombre indica el orden en que se arranca/para el servicio por INIT, pq se levantan en cascada (secuencialmente)

el inconveniente de INIT es esa espera por el levantamiento secuencial de los servicios; asi nacio SYSTEMD (system-daemon) o el nuevo INIT, sigue siendo el proceso inicial pero arranca los servicios en paralelo (a la vez)

para SYSTEMD no existen runlevels, sino TARGETS:

```

poweroff.target <--- simula el runlevel 0

```

```
rescue.target <----- simula el runlevel 1
graphical.target <--- simula el runlevel 5
reboot.target <----- simula el runlevel 6
```

una vez arrancados los servicios del target/runlevel, se
lanza el servicio del entorno grafico para q el usuario
meta sus credenciales (se autentifique)