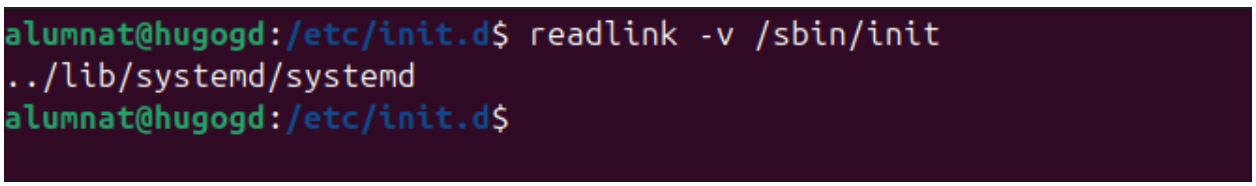
Unidad 1

Comandos Generales

Estos comandos sirven para saber los servicios que tenemos

Man Init

readlink -v /sbin/init



El comando **runlevel** dice en que nivel estas al ser SystemV es el N5

Con esta comanda se mostraran las correspondencias entre RUNLEVEL y targets

```
alumnat@hugogd:~$ ls -l /lib/systemd/system/runlevel*.target
lrwxrwxrwx 1 root root 15 d'abr.   19  2024 /lib/systemd/system/runlevel0.target -> poweroff.target
lrwxrwxrwx 1 root root 13 d'abr.   19  2024 /lib/systemd/system/runlevel1.target -> rescue.target
lrwxrwxrwx 1 root root 17 d'abr.   19  2024 /lib/systemd/system/runlevel2.target -> multi-user.target
lrwxrwxrwx 1 root root 17 d'abr.   19  2024 /lib/systemd/system/runlevel3.target -> multi-user.target
lrwxrwxrwx 1 root root 17 d'abr.   19  2024 /lib/systemd/system/runlevel4.target -> multi-user.target
lrwxrwxrwx 1 root root 16 d'abr.   19  2024 /lib/systemd/system/runlevel5.target -> graphical.target
lrwxrwxrwx 1 root root 13 d'abr.   19  2024 /lib/systemd/system/runlevel6.target -> reboot.target
alumnat@hugogd:~$
```

Dentro de la carpeta **/etc/init.d** estaran todos los scripts modificables dentro del sistema

```
alumnat@hugogd:/etc/init.d$ ls -l
total 132
-rwxr-xr-x 1 root root 5623 de nov.  30  2022 alsa-utils
-rwxr-xr-x 1 root root 2055 de set.  25  2023 anacron
-rwxr-xr-x 1 root root 3740 de gen.  10  2024 apparmor
-rwxr-xr-x 1 root root 2275 d'abr.   19  2024 apport
-rwxr-xr-x 1 root root 2968 de gen.   3  2024 bluetooth
-rwxr-xr-x 1 root root 1235 de febr. 26  2024 console-setup.sh
-rwxr-xr-x 1 root root 3103 de febr. 27  2024 cron
-rwxr-xr-x 1 root root 2804 de febr. 28  2024 cups
-rwxr-xr-x 1 root root 3152 de des.   5  2023 dbus
-rwxr-xr-x 1 root root 3029 d'abr.    4  2024 gdm3
-rwxr-xr-x 1 root root  985 de febr. 23  2024 grub-common
-rwxr-xr-x 1 root root 3131 de maig  19  2017 kerneloops
-rwxr-xr-x 1 root root 1482 de febr. 26  2024 keyboard-setup.sh
-rwxr-xr-x 1 root root 2043 d'abr.   18  2024 kmod
-rwxr-xr-x 1 root root 9138 de febr. 28  2024 openvpn
-rwxr-xr-x 1 root root 1386 de març  21  2024 plymouth
-rwxr-xr-x 1 root root  760 de març  21  2024 plymouth-log
-rwxr-xr-x 1 root root  959 de març  24  2024 procps
-rwxr-xr-x 1 root root 4417 d'abr.   12  2024 rsync
-rwxr-xr-x 1 root root 2224 d'abr.   15  2018 saned
-rwxr-xr-x 1 root root 2040 d'oct.    5  2023 speech-dispatcher
-rwxr-xr-x 1 root root 2484 de nov.  21  2023 spice-vdagent
-rwxr-xr-x 1 root root 2555 d'abr.    5  2024 sssd
-rwxr-xr-x 1 root root 1581 de gen.   9  2024 sysstat
-rwxr-xr-x 1 root root 2091 de febr. 11  2024 ufw
-rwxr-xr-x 1 root root 1391 de febr. 12  2024 unattended-upgrades
-rwxr-xr-x 1 root root 1306 d'abr.    9  2024 uuidd
-rwxr-xr-x 1 root root  485 d'ag.    11  2018 whoopsie
-rwxr-xr-x 1 root root 2762 d'oct.   19  2021 x11-common
alumnat@hugogd:/etc/init.d$
```

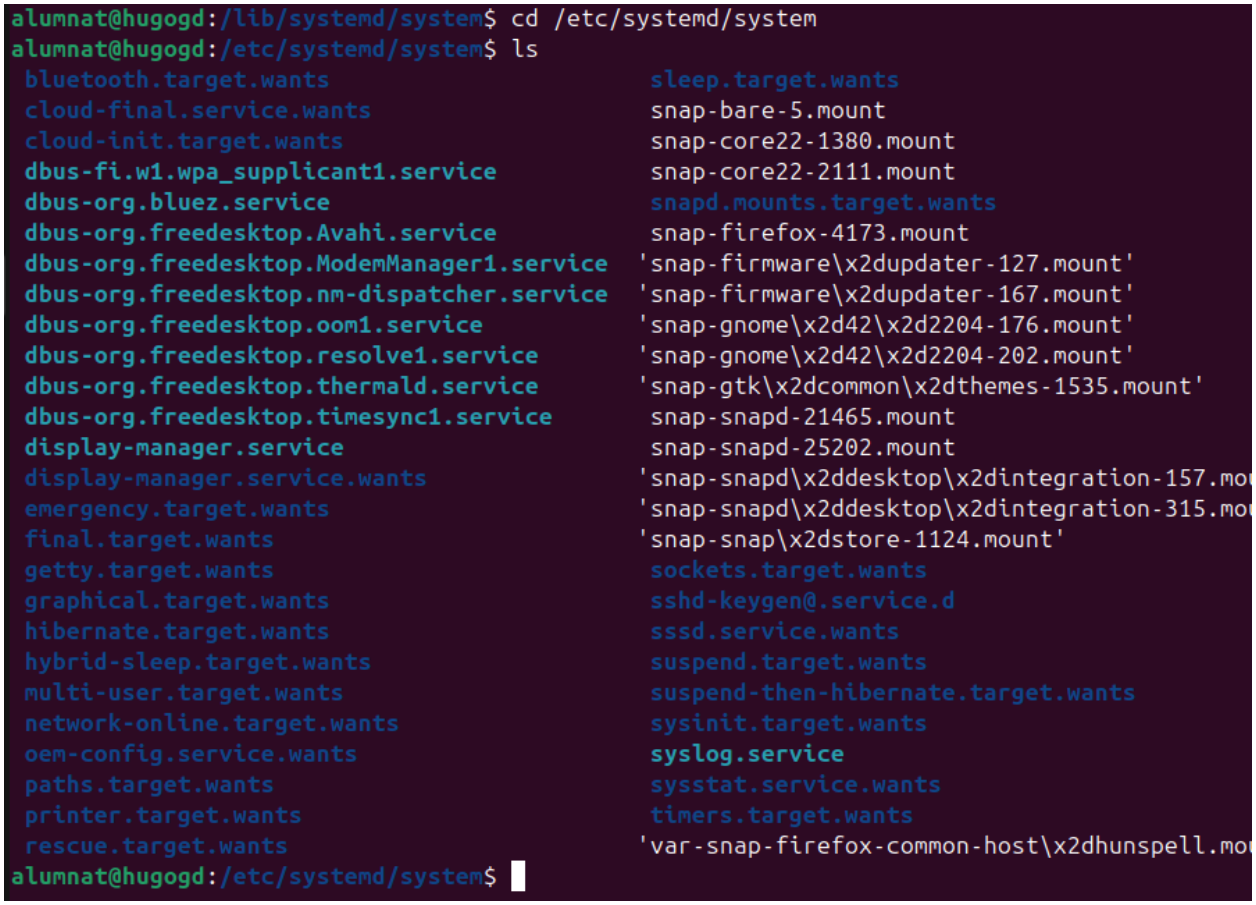
Si entramos por ejemplo al **runlevel nivel 0** veremos las procesos asociadas al los scripts del poweroff.

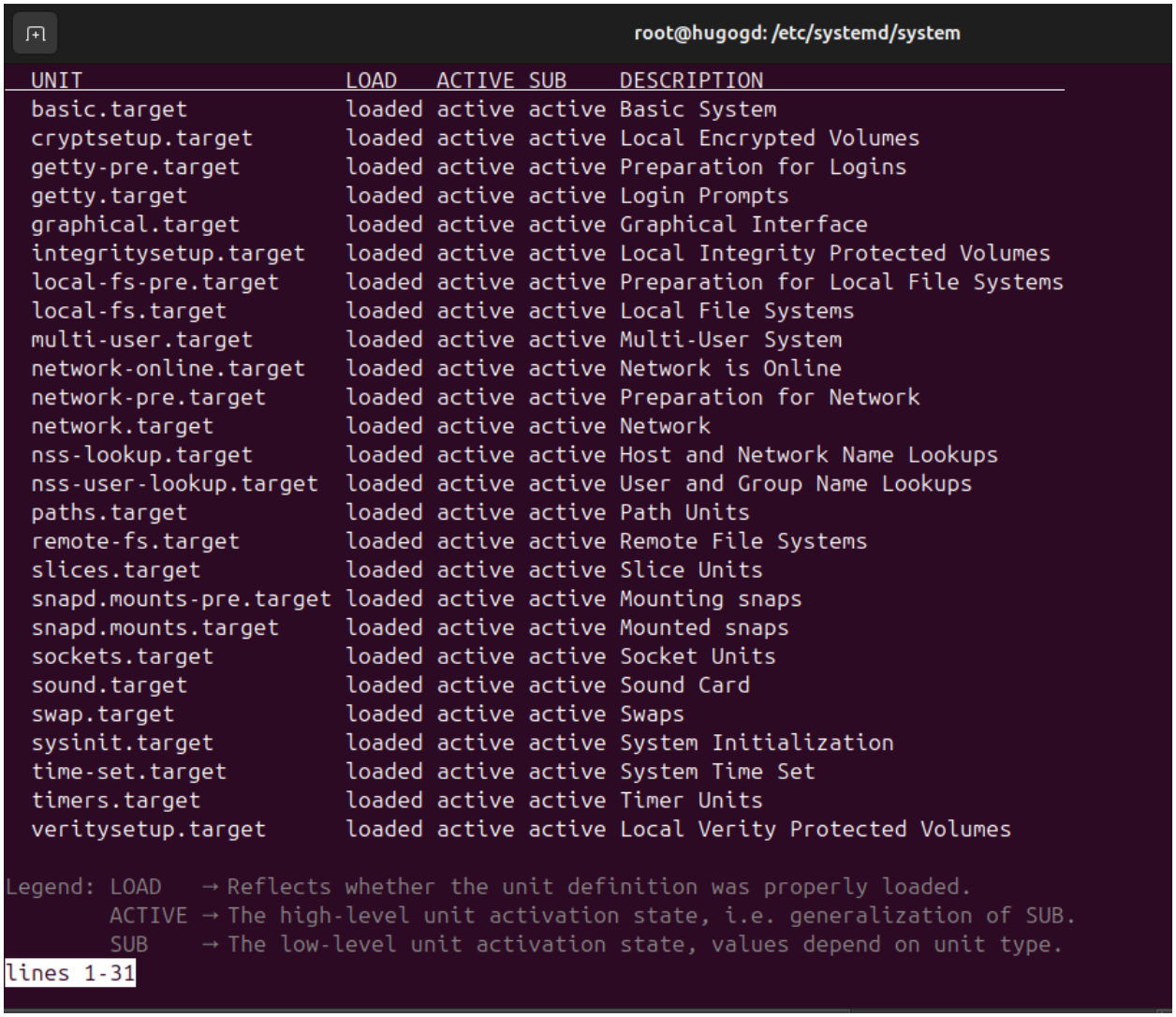
```
alumnat@hugogd:/etc/rc0.d$ ls -l
total 0
lrwxrwxrwx 1 root root 20 d'abr.   24  2024 K01alsa-utils -> ../init.d/alsa-utils
lrwxrwxrwx 1 root root 19 d'abr.   24  2024 K01bluetooth -> ../init.d/bluetooth
lrwxrwxrwx 1 root root 14 d'abr.   24  2024 K01gdm3 -> ../init.d/gdm3
lrwxrwxrwx 1 root root 20 d'abr.   24  2024 K01kerneloops -> ../init.d/kerneloops
lrwxrwxrwx 1 root root 17 d'abr.   24  2024 K01openvpn -> ../init.d/openvpn
lrwxrwxrwx 1 root root 18 d'abr.   24  2024 K01plymouth -> ../init.d/plymouth
lrwxrwxrwx 1 root root 15 d'abr.   24  2024 K01saned -> ../init.d/saned
lrwxrwxrwx 1 root root 27 d'abr.   24  2024 K01speech-dispatcher -> ../init.d/speech-disp
lrwxrwxrwx 1 root root 23 d'abr.   24  2024 K01spice-vdagent -> ../init.d/spice-vdagent
lrwxrwxrwx 1 root root 14 d'abr.   24  2024 K01sssd -> ../init.d/sssd
lrwxrwxrwx 1 root root 29 d'abr.   24  2024 K01unattended-upgrades -> ../init.d/unattende
lrwxrwxrwx 1 root root 15 d'abr.   24  2024 K01uuidd -> ../init.d/uuidd
alumnat@hugogd:/etc/rc0.d$
```

En la carpeta /lib/systemd/system tenemos todos los targets de los diferentes niveles de runlevel que hay, en cada uno de estos tendremos la equivalencia del target como .wants

```
alumnat@hugogd:/lib/systemd/system$ ls
accounts-daemon.service                 smartcard.target
alsa-restore.service                    snapd.apparmor.service
alsa-state.service                      snapd.autoimport.service
alsa-utils.service                      snapd.core-fixup.service
anacron.service                         snapd.failure.service
anacron.timer                           snapd.mounts-pre.target
apparmor.service                        snapd.mounts.target
apport-autoreport.path                  snapd.recovery-chooser-trigger.service
apport-autoreport.service               snapd.seeded.service
apport-autoreport.timer                 snapd.service
apport-coredump-hook@.service           snapd.snap-repair.service
apport-forward@.service                 snapd.snap-repair.timer
apport-forward.socket                   snapd.socket
apport.service                          snapd.system-shutdown.service
apt-daily.service                       sockets.target
apt-daily.timer                         sockets.target.wants
apt-daily-upgrade.service               soft-reboot.target
apt-daily-upgrade.timer                 sound.target
apt-news.service                        sound.target.wants
autovt@.service                         speech-dispatcherd.service
avahi-daemon.service                    spice-vdagentd.service
avahi-daemon.socket                     spice-vdagentd.socket
basic.target                            spice-vdagent.service
blockdev@.target                        ssl-cert.service
bluetooth.service                       sssd-autofs.service
bluetooth.target                        sssd-autofs.socket
bolt.service                            sssd-nss.service
boot-complete.target                    sssd-nss.socket
brltty.service                          sssd-pac.service
brltty-udev.service                     sssd-pac.socket
cloud-config.service                    sssd-pam-priv.socket
```

Siempre que tengamos que modificar algo es mejor modificarlo en la carpeta /etc/systemd/system y no tocar los de la carpeta /lib para tenerlos como copia de seguridad





Por ejemplo con el comando systemctl get-default tendremos el target default

```
root@hugogd:/etc/systemd/system# systemctl get-default
graphical.target
root@hugogd:/etc/systemd/system#
```

Para saber sus dependencias tendremos que usar el comando **systemctl list-dependencies graphical.target**

```
root@hugogd:/etc/systemd/system# systemctl list-dependencies graphical.target
graphical.target
● ├─accounts-daemon.service
● ├─gdm.service
● ├─gnome-remote-desktop.service
● ├─power-profiles-daemon.service
● ├─switcheroo-control.service
○ ├─systemd-update-utmp-runlevel.service
● ├─udisks2.service
● └─multi-user.target
○   ├─anacron.service
●   ├─apport.service
●   ├─avahi-daemon.service
●   ├─console-setup.service
●   ├─cron.service
●   ├─cups-browsed.service
●   ├─cups.path
●   ├─cups.service
●   ├─dbus.service
○   ├─dmesg.service
○   ├─e2scrub_reap.service
○   ├─grub-common.service
○   ├─grub-initrd-fallback.service
●   ├─kerneloops.service
●   ├─ModemManager.service
○   ├─networkd-dispatcher.service
●   ├─NetworkManager.service
●   ├─openvpn.service
●   ├─plymouth-quit-wait.service
○   ├─plymouth-quit.service
```

Por ejemplo podemos probarlo con otros **targets** como por ejemplo **poweroff.target**

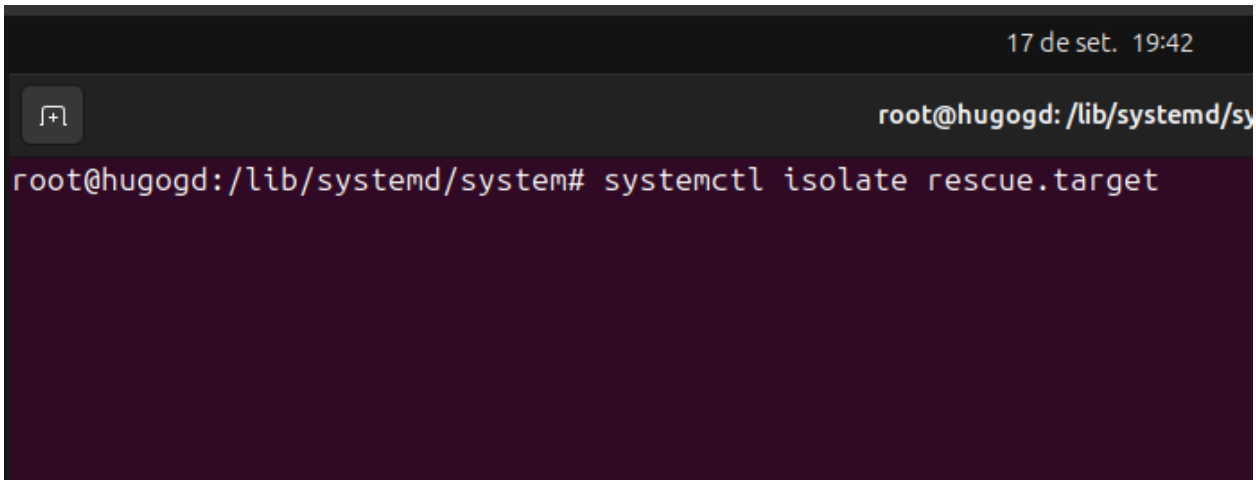
```
root@hugogd:/etc/systemd/system# systemctl list-dependencies poweroff.target
poweroff.target
○ ├─plymouth-poweroff.service
○ ├─plymouth-switch-root-initramfs.service
○ └─systemd-poweroff.service
root@hugogd:/etc/systemd/system#
```

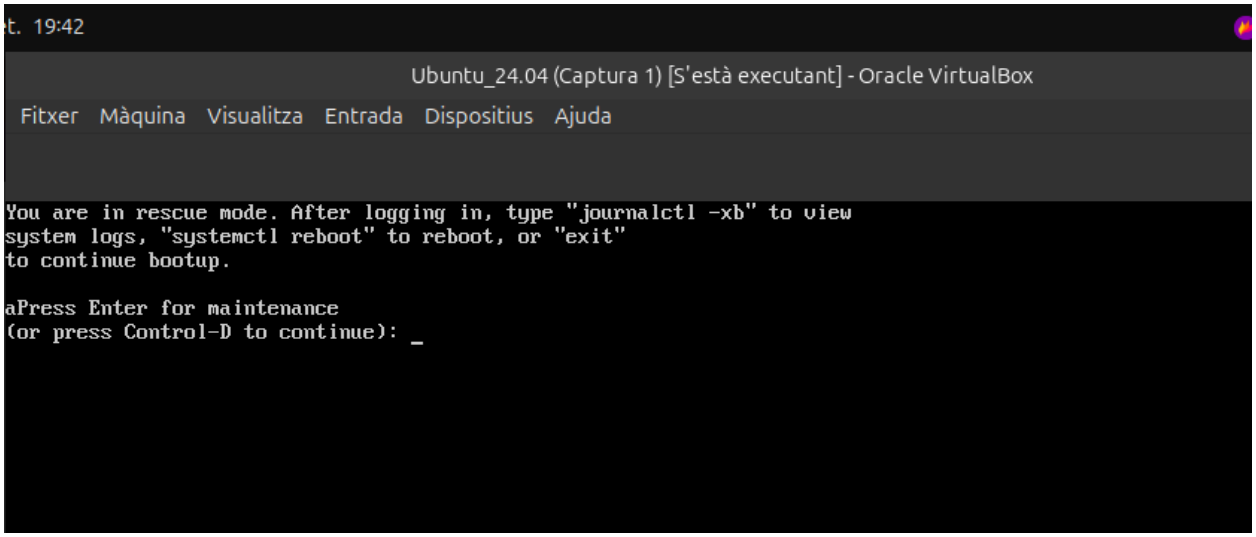
Para saber donde se encuentra un servicio podemos usar el comando ls con la ruta de la carpeta usando ***.wants** para buscar en todos los archivos **.wants**

```
root@hugogd:/lib/systemd/system# ls /etc/systemd/system/*.wants/cron.service
/etc/systemd/system/multi-user.target.wants/cron.service
root@hugogd:/lib/systemd/system#
```

Para cambiar el target por defecto lo podemos hacer de forma definitiva o temporal para cambiarlo usaremos:

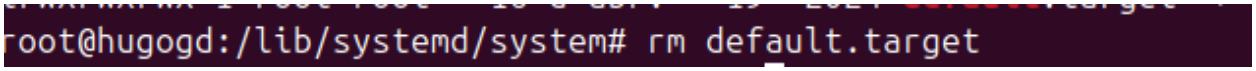
TEMPORAL:





POR DEFECTO:

Si hacemos un ls vemos que tiene una l delante eso es que es un enlace lo borraremos usando



Y crearemos uno nuevo con el comando

```
root@hugogd:/lib/systemd/system# ln -s  rescue.target default.target
root@hugogd:/lib/systemd/system# ls -l | grep default
lrwxrwxrwx 1 root root   13 de set.  17 19:45 default.target -> rescue.target
root@hugogd:/lib/systemd/system#
```

ahora haciendo un reboot podremos comprobar que se inicia usando rescue.target

Para volver a cambiar el target con el que iniciara tendremos que hacer lo contrario

- rm default.target
- ln -s graphical.target default.target

Para saber como esta un servicio podemos hacer un systemctl status en este caso **SSH** que no lo tenemos

```
root@hugogd:/home/alumnat# systemctl status ssh
Unit ssh.service could not be found.
root@hugogd:/home/alumnat#
```

Una vez instalado con apt install ssh podremos volver a hacer un status para comprobar que lo tenemos.

```
root@hugogd:/home/alumnat# systemctl status ssh
○ ssh.service - OpenBSD Secure Shell server
     Loaded: loaded (/usr/lib/systemd/system/ssh.service; disabled; preset: ena>
     Active: inactive (dead)
TriggeredBy: ● ssh.socket
       Docs: man:sshd(8)
             man:sshd_config(5)
lines 1-6/6 (END)
```

El comando **systemctl is-enabled “servicio”** sirve para saber si el servicio esta en un target

```
root@hugogd:/# systemctl is-enabled cron
enabled
root@hugogd:/#
```

El comando **systemctl enabled “servicio”** servira para crear un enlace para que se encienda default el servicio con la arrancada del ordenador y **system disabled “servicio”** para quitar el enlace.

```
root@alumnat-VirtualBox:/etc/rc2.d# systemctl enable ssh
Synchronizing state of ssh.service with SysV service script with /usr/lib/systemd/system
d-sysv-install.
Executing: /usr/lib/systemd/systemd-sysv-install enable ssh
Created symlink /etc/systemd/system/sshd.service → /usr/lib/systemd/system/ssh.service.
Created symlink /etc/systemd/system/multi-user.target.wants/ssh.service → /usr/lib/syste
md/system/ssh.service.
root@alumnat-VirtualBox:/etc/rc2.d#
```

```
root@alumnat-VirtualBox:/etc/rc2.d# systemctl disable ssh
Synchronizing state of ssh.service with SysV service script with /usr/lib/systemd/system
d-sysv-install.
Executing: /usr/lib/systemd/systemd-sysv-install disable ssh
Removed "/etc/systemd/system/multi-user.target.wants/ssh.service".
Removed "/etc/systemd/system/sshd.service".
Disabling 'ssh.service', but its triggering units are still active:
ssh.socket
root@alumnat-VirtualBox:/etc/rc2.d#
```

Para hacer que el ssh se habilite con un servicio en este caso **multi-user.target.wants.**

Primero deshabilitaremos el **cron** con el domando

```
root@hugogd:/etc/systemd/system/multi-user.target.wants# systemctl disable cron
Synchronizing state of cron.service with SysV service script with /usr/lib/syste
md/systemd-sysv-install.
Executing: /usr/lib/systemd/systemd-sysv-install disable cron
Removed "/etc/systemd/system/multi-user.target.wants/cron.service".
root@hugogd:/etc/systemd/system/multi-user.target.wants#
```

Y habilitaremos **SSH** con el comando

```
root@hugogd:/etc/systemd/system/multi-user.target.wants# systemctl enable ssh
Synchronizing state of ssh.service with SysV service script with /usr/lib/system
d/systemd-sysv-install.
Executing: /usr/lib/systemd/systemd-sysv-install enable ssh
Created symlink /etc/systemd/system/sshd.service → /usr/lib/systemd/system/ssh.s
ervice.
Created symlink /etc/systemd/system/multi-user.target.wants/ssh.service → /usr/l
ib/systemd/system/ssh.service.
root@hugogd:/etc/systemd/system/multi-user.target.wants# reboot
```

Y reiniciaremos para comprobarlo

```
root@hugogd: /home/alumnat
root@hugogd:/home/alumnat# systemctl status cron
○ cron.service - Regular background program processing daemon
     Loaded: loaded (/usr/lib/systemd/system/cron.service; disabled; preset: en>
     Active: inactive (dead)
       Docs: man:cron(8)
lines 1-4/4 (END)
```

```
root@hugogd:/home/alumnat# systemctl status ssh
● ssh.service - OpenBSD Secure Shell server
     Loaded: loaded (/usr/lib/systemd/system/ssh.service; enabled; preset: enab>
     Active: active (running) since Wed 2025-09-17 20:08:46 CEST; 38s ago
TriggeredBy: ● ssh.socket
       Docs: man:sshd(8)
             man:sshd_config(5)
    Process: 1007 ExecStartPre=/usr/sbin/sshd -t (code=exited, status=0/SUCCESS)
   Main PID: 1020 (sshd)
      Tasks: 1 (limit: 4615)
     Memory: 2.1M (peak: 2.3M)
        CPU: 37ms
     CGroup: /system.slice/ssh.service
             └─1020 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"

de set. 17 20:08:46 hugogd systemd[1]: Starting ssh.service - OpenBSD Secure Sh>
de set. 17 20:08:46 hugogd sshd[1020]: Server listening on :: port 22.
de set. 17 20:08:46 hugogd systemd[1]: Started ssh.service - OpenBSD Secure She>
lines 1-17/17 (END)
```

Para comprobar que estan sincronizado haremos un ls -l en /etc/rc5.d podremos ver que esta como K01 (kill)

```
root@hugogd:/etc/rc5.d# ls -l
total 0
lrwxrwxrwx 1 root root 14 d'abr.   24  2024 K01cron -> ../init.d/cron
lrwxrwxrwx 1 root root 27 d'abr.   24  2024 K01speech-dispatcher -> ../init.d/speech-disp
lrwxrwxrwx 1 root root 17 d'abr.   24  2024 S01anacron -> ../init.d/anacron
lrwxrwxrwx 1 root root 16 d'abr.   24  2024 S01apport -> ../init.d/apport
lrwxrwxrwx 1 root root 19 d'abr.   24  2024 S01bluetooth -> ../init.d/bluetooth
lrwxrwxrwx 1 root root 26 d'abr.   24  2024 S01console-setup.sh -> ../init.d/console-setu
lrwxrwxrwx 1 root root 14 d'abr.   24  2024 S01cups -> ../init.d/cups
lrwxrwxrwx 1 root root 14 d'abr.   24  2024 S01dbus -> ../init.d/dbus
lrwxrwxrwx 1 root root 14 d'abr.   24  2024 S01gdm3 -> ../init.d/gdm3
lrwxrwxrwx 1 root root 21 d'abr.   24  2024 S01grub-common -> ../init.d/grub-common
lrwxrwxrwx 1 root root 20 d'abr.   24  2024 S01kerneloops -> ../init.d/kerneloops
lrwxrwxrwx 1 root root 17 d'abr.   24  2024 S01openvpn -> ../init.d/openvpn
lrwxrwxrwx 1 root root 18 d'abr.   24  2024 S01plymouth -> ../init.d/plymouth
lrwxrwxrwx 1 root root 15 d'abr.   24  2024 S01rsync -> ../init.d/rsync
lrwxrwxrwx 1 root root 15 d'abr.   24  2024 S01saned -> ../init.d/saned
lrwxrwxrwx 1 root root 23 d'abr.   24  2024 S01spice-vdagent -> ../init.d/spice-vdagent
lrwxrwxrwx 1 root root 13 de set.  17 19:56 S01ssh -> ../init.d/ssh
lrwxrwxrwx 1 root root 14 d'abr.   24  2024 S01sssd -> ../init.d/sssd
lrwxrwxrwx 1 root root 17 d'abr.   24  2024 S01sysstat -> ../init.d/sysstat
lrwxrwxrwx 1 root root 29 d'abr.   24  2024 S01unattended-upgrades -> ../init.d/unattende
lrwxrwxrwx 1 root root 15 d'abr.   24  2024 S01uuidd -> ../init.d/uuidd
lrwxrwxrwx 1 root root 18 d'abr.   24  2024 S01whoopsie -> ../init.d/whoopsie
```

Si lo habilitamos y volvemos a hacer el ls -l podremos ver que sale como S01cron

```
root@hugogd:/etc/rc5.d# systemctl enable cron
Synchronizing state of cron.service with SysV service script with /usr/lib/systemd/systemd
Executing: /usr/lib/systemd/systemd-sysv-install enable cron
Created symlink /etc/systemd/system/multi-user.target.wants/cron.service → /usr/lib/syste
root@hugogd:/etc/rc5.d# ls -l
total 0
lrwxrwxrwx 1 root root 27 d'abr.   24  2024 K01speech-dispatcher -> ../init.d/speech-disp
lrwxrwxrwx 1 root root 17 d'abr.   24  2024 S01anacron -> ../init.d/anacron
lrwxrwxrwx 1 root root 16 d'abr.   24  2024 S01apport -> ../init.d/apport
lrwxrwxrwx 1 root root 19 d'abr.   24  2024 S01bluetooth -> ../init.d/bluetooth
lrwxrwxrwx 1 root root 26 d'abr.   24  2024 S01console-setup.sh -> ../init.d/console-setu
lrwxrwxrwx 1 root root 14 d'abr.   24  2024 S01cron -> ../init.d/cron
lrwxrwxrwx 1 root root 14 d'abr.   24  2024 S01cups -> ../init.d/cups
lrwxrwxrwx 1 root root 14 d'abr.   24  2024 S01dbus -> ../init.d/dbus
lrwxrwxrwx 1 root root 14 d'abr.   24  2024 S01gdm3 -> ../init.d/gdm3
lrwxrwxrwx 1 root root 21 d'abr.   24  2024 S01grub-common -> ../init.d/grub-common
lrwxrwxrwx 1 root root 20 d'abr.   24  2024 S01kerneloops -> ../init.d/kerneloops
lrwxrwxrwx 1 root root 17 d'abr.   24  2024 S01openvpn -> ../init.d/openvpn
lrwxrwxrwx 1 root root 18 d'abr.   24  2024 S01plymouth -> ../init.d/plymouth
lrwxrwxrwx 1 root root 15 d'abr.   24  2024 S01rsync -> ../init.d/rsync
lrwxrwxrwx 1 root root 15 d'abr.   24  2024 S01saned -> ../init.d/saned
lrwxrwxrwx 1 root root 23 d'abr.   24  2024 S01spice-vdagent -> ../init.d/spice-vdagent
lrwxrwxrwx 1 root root 13 de set.  17 19:56 S01ssh -> ../init.d/ssh
lrwxrwxrwx 1 root root 14 d'abr.   24  2024 S01sssd -> ../init.d/sssd
lrwxrwxrwx 1 root root 17 d'abr.   24  2024 S01sysstat -> ../init.d/sysstat
lrwxrwxrwx 1 root root 29 d'abr.   24  2024 S01unattended-upgrades -> ../init.d/unattende
lrwxrwxrwx 1 root root 15 d'abr.   24  2024 S01uuidd -> ../init.d/uuidd
lrwxrwxrwx 1 root root 18 d'abr.   24  2024 S01whoopsie -> ../init.d/whoopsie
root@hugogd:/etc/rc5.d#
```

Con el systemd-analyze podremos ver cuanto esta tardando en arrancar el servicio del nivel en el que estamos en este caso el graphical

```
root@hugogd:/etc/rc5.d# systemd-analyze
Startup finished in 2.313s (kernel) + 6.692s (userspace) = 9.006s
graphical.target reached after 6.662s in userspace.
root@hugogd:/etc/rc5.d#
```

Y para ver lo que tardan todos los servicios podemos usar el comando systemd-analyze blame

```
root@hugogd:/etc/rc5.d# systemd-analyze blame
2.991s plymouth-quit-wait.service
2.258s snapd.seeded.service
2.189s snapd.service
1.370s NetworkManager.service
 930ms snapd.apparmor.service
 902ms systemd-resolved.service
 876ms apport.service
 824ms systemd-oomd.service
 823ms systemd-binfmt.service
 820ms systemd-timesyncd.service
 691ms e2scrub_reap.service
 632ms dev-sda2.device
 479ms dev-loop9.device
 474ms dev-loop8.device
 468ms dev-loop10.device
 467ms dev-loop11.device
 464ms dev-loop12.device
 464ms dev-loop13.device
 451ms accounts-daemon.service
 379ms gnome-remote-desktop.service
 334ms rsyslog.service
 320ms power-profiles-daemon.service
 315ms polkit.service
 310ms udisks2.service
 288ms apparmor.service
 283ms gpu-manager.service
 276ms ModemManager.service
 256ms NetworkManager-wait-online.service
 230ms grub-common.service
```