

BOOT E SHUTDOWN

ADMINISTRAÇÃO DE SISTEMAS
2020/2021

ROLANDO MARTINS
(ADAPTADOS DE PEDRO BRANDÃO)

Referências dos slides

- O conteúdo destes slides é baseado no livro da disciplina: “Unix and Linux System Administration Handbook (4ªEd)” por Evi Nemeth, Garth Snyder, Trent R. Hein e Ben Whaley, Prentice Hall, ISBN: 0-13-148005-7
- As imagens usadas têm a atribuição aos autores ou são de uso livre.

Bootstrapping (Boot)

- Carregar sistema sem ter as funcionalidades de um SO.
- “Pull itself by its own bootstraps”.
- “Booting” é uma abreviatura de “Bootstrapping”.



BOOT - Fases

Inicializar máquina por fases. Providenciando serviços cada vez mais complexos às fases seguintes:

- Leitura do boot loader do MBR (Master Boot Record);
 - Lilo ou grub em Linux
- Carregar e inicializar o Kernel
- Detecção de dispositivos e sua configuração
- Criação dos processos de kernel
- Intervenção do Administrador (quando em modo single-user)
- Execução dos scripts de arranque

BOOT – Fases (pré boot-loader)

- Arranque da BIOS:
 - Detecção de Hardware
 - Execução do firmware de inicialização para cada um dos componentes e periféricos da máquina.
 - Localização do programa de boot inicial (lilo; grub; bootmgr).

GRUB

grub

- Grub: bootloader mais sofisticado que lilo.
 - GRand Unified Bootloader
- Versão legacy e Grub 2
- Ambas permitem multi-booting
 - Arranque de mais do que 1 sistema

/boot/grub2/grub.cfg

```
menuentry 'Fedora (5.3.5-200.fc30.x86_64) 30 (Thirty)'{  
    load_video  
    set gfxpayload=keep  
    insmod gzio  
    insmod part_msdos  
    insmod ext2  
    set root='hd0,msdos1'  
    linux /vmlinuz-5.3.5-200.fc30.x86_64 root=UUID=d667244d-  
afaa-4e45-8ce7-7eb7312756a6 ro resume=UUID=4c460e4f-6d1b-492e-  
b822-d22654ba0af3 rhgb quiet  
    initrd /initramfs-5.3.5-200.fc30.x86_64.img  
}
```


BootLoaderSpec

- Fedora assumiu o [BootLoaderSpec \(BLS\)](#)
 - permite que diferentes bootloaders usem as mesmas configurações
 - Parte do [Systemd BLS](#)
- Ver no `/boot/grub2/grub.cfg`
 - `blscfg`
- Ver `/boot/loader/`
- Ver no `/etc/default/grub`
 - `GRUB_ENABLE_BSLCFG`

Nota:

The Boot Loader Specification (BLS) defines a scheme and file format to manage boot loader configuration **for each boot option in a drop-in directory, without the need to manipulate bootloader configuration files.** Directories of individual drop-in configuration files are standard for many purposes on Linux nowadays, so the goal is to also extend this concept for boot menu entries...

Grub: single user

- Em modo interativo, para entrar em single-user:

```
grub> set root=(hd0,msdos1)
```

```
grub> linux /vmlinuz-5.3.5-200.fc30.x86_64 ro \
root=/dev/sda1 rhgb selinux=0 single
```

```
grub> initrd /initramfs-5.3.5-200.fc30.x86_64.img
```

```
grub> boot
```

Para uma partição com sistema de ficheiros
(não tem LVM). Pode-se ver os discos e conteúdos com:
grub> ls

Se for indicado com o UUID (como slide atrás) é menos
suscetível a mudanças no /dev

Grub (legacy): single user

- Em modo interativo, para entrar em single-user:

```
grub> root (hd0,0)
```

```
grub> kernel /boot/vmlinuz ro root=/dev/sda1 single
```

```
grub> initrd /boot/initrd.img
```

```
grub> boot
```

- Argumento `single` é passado ao `init` → entra em single-user mode.

grub

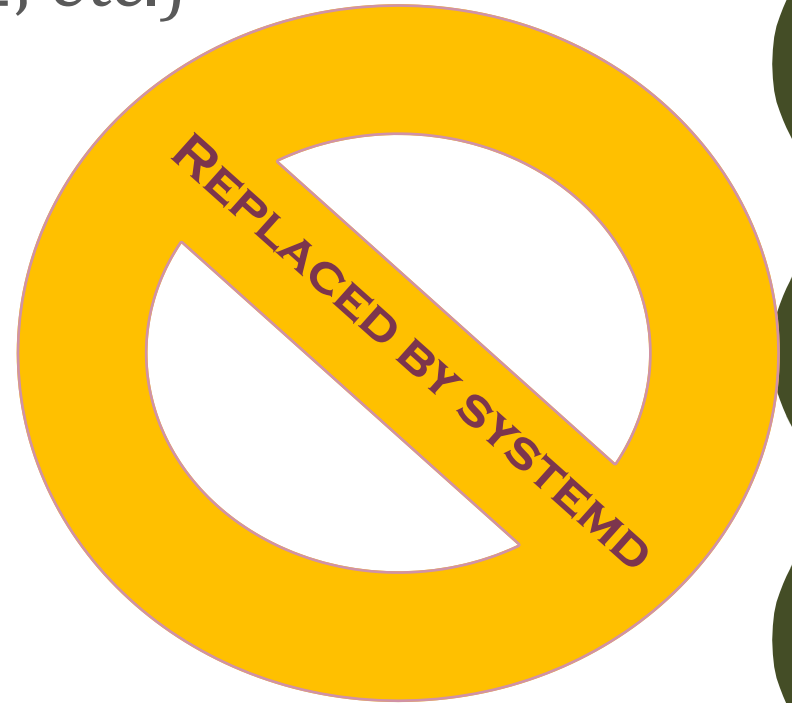
- Ficheiro criado automaticamente em:
 - `/boot/grub2/grub.cfg`
- Configurações:
 - `/etc/default/grub`
 - `/etc/grub.d/`
 - Diretórios com os sistemas
- Comandos
 - `grub-install`: instalar o grub num dispositivo
 - Apenas 1 vez, e quando o grub mudar
 - `grub2-install` (fedora)
 - Que chama `grub2-mkimage`, `grub2-bios-setup`, etc.
 - `grub2-mkconfig/grub-mkconfig`
 - Criar a configuração que o grub acede quando arranca

FICHEIROS DE ARRANQUE

PARA SISTEMAS COM SYSTEMV E SYSTEMD

Ficheiros e pastas - SystemV

- Scripts: `/etc/rc*` (`rc.sysinit`; `rc.local`; etc.)
- Pastas: `/etc/rc*.d` (`rc0.d`; etc.)
 - Associados ao nível de execução (run level)
 - Contêm links para scripts em `/etc/init.d/`
- Comando para gerir: `chkconfig`
- `/etc/inittab`
 - Diz ao `init` o que correr em cada nível
- Ficheiros de configuração específica no dir: `/etc/sysconfig/`.



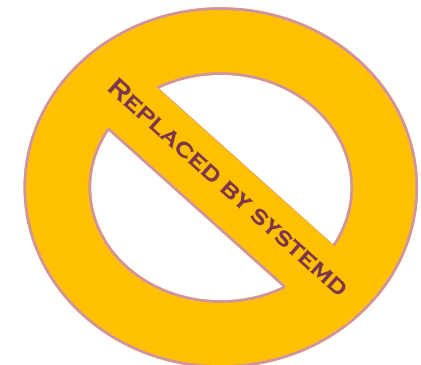
Níveis de execução.

When **init** transitions from a lower run level to a higher one, it runs all the scripts that start with **S** in ascending numerical order with the argument **start**.

When **init** transitions from a higher run level to a lower one, it runs all the scripts that start with **K** (for “kill”) in descending numerical order with the argument **stop**.

```
# ln -s /etc/init.d/cups /etc/rc2.d/S80cups  
# ln -s /etc/init.d/cups /etc/rc0.d/K80cups
```

Although the network and sshd are both configured to start at run level 3 in Red Hat, the **network** script has sequence number 10 and the **sshd** script has sequence number 55, so **network** is certain to be run first. Be sure to consider this type of dependency when you add a new service.



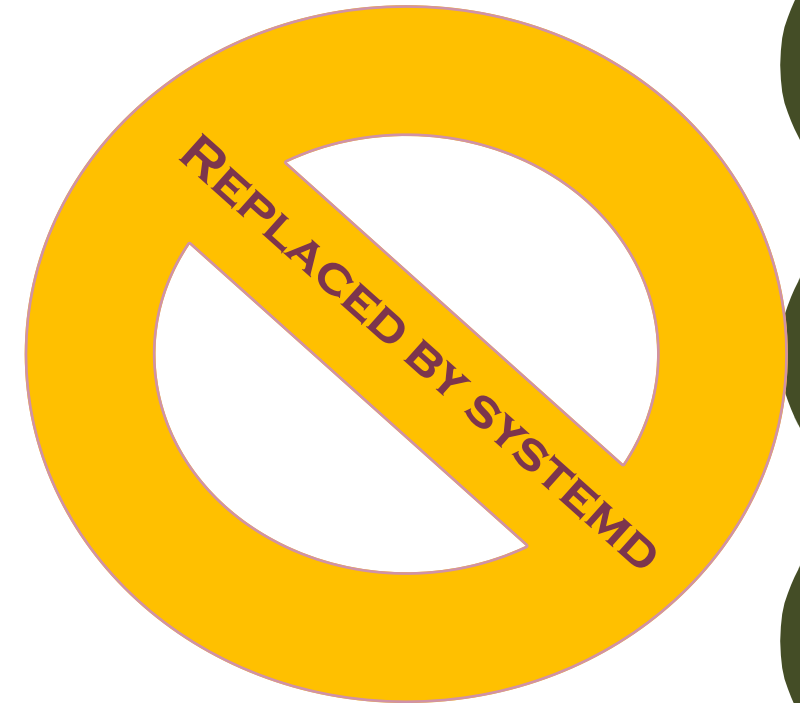
Exemplo Script de arranque SSHD

```
#!/bin/sh
test -f /usr/bin/sshd || exit 0
case "$1" in
start)
    echo -n "Starting sshd: sshd"
    /usr/sbin/sshd
    echo "."
    ;;

stop)
    echo -n "Stopping sshd: sshd"
    kill `cat /var/run/sshd.pid`
    echo "."
    ;;

restart)
    echo -n "Stopping sshd: sshd"
    kill `cat /var/run/sshd.pid`
    echo "."
    echo -n "Starting sshd: sshd"
    /usr/sbin/sshd
    echo "."
    ;;

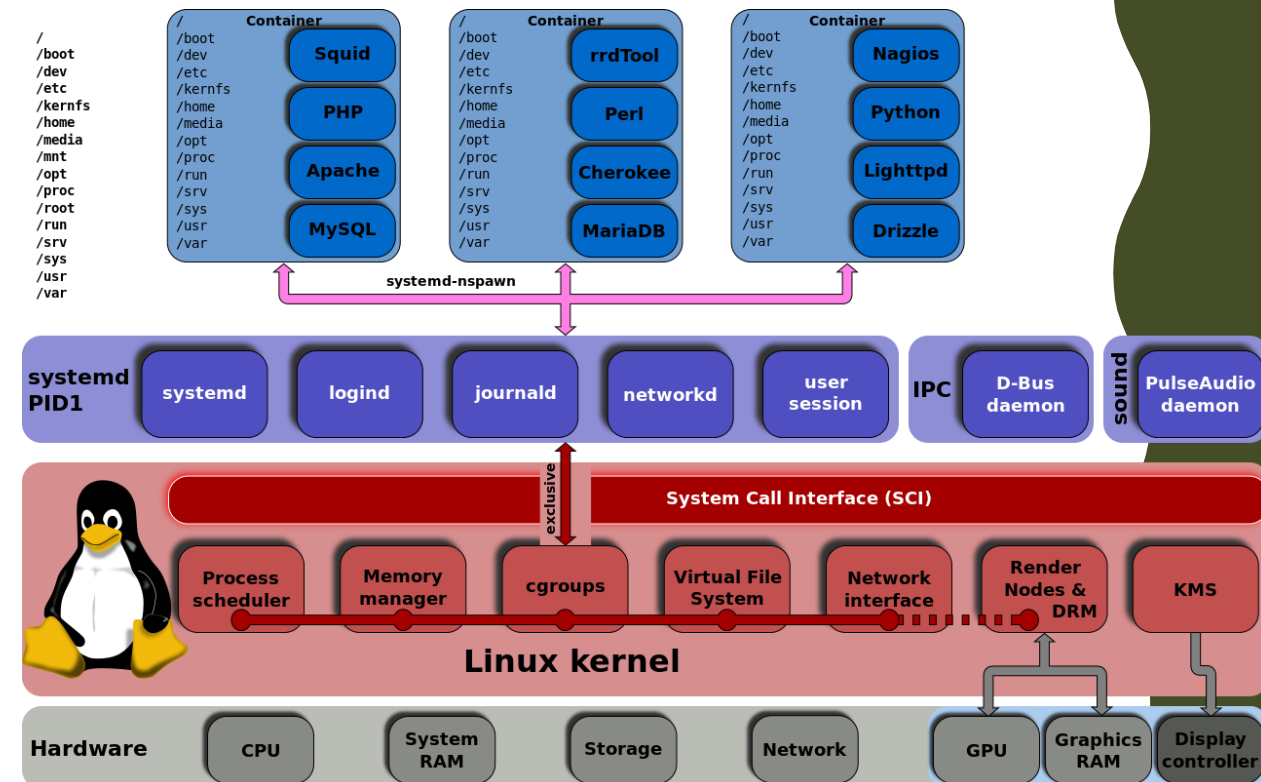
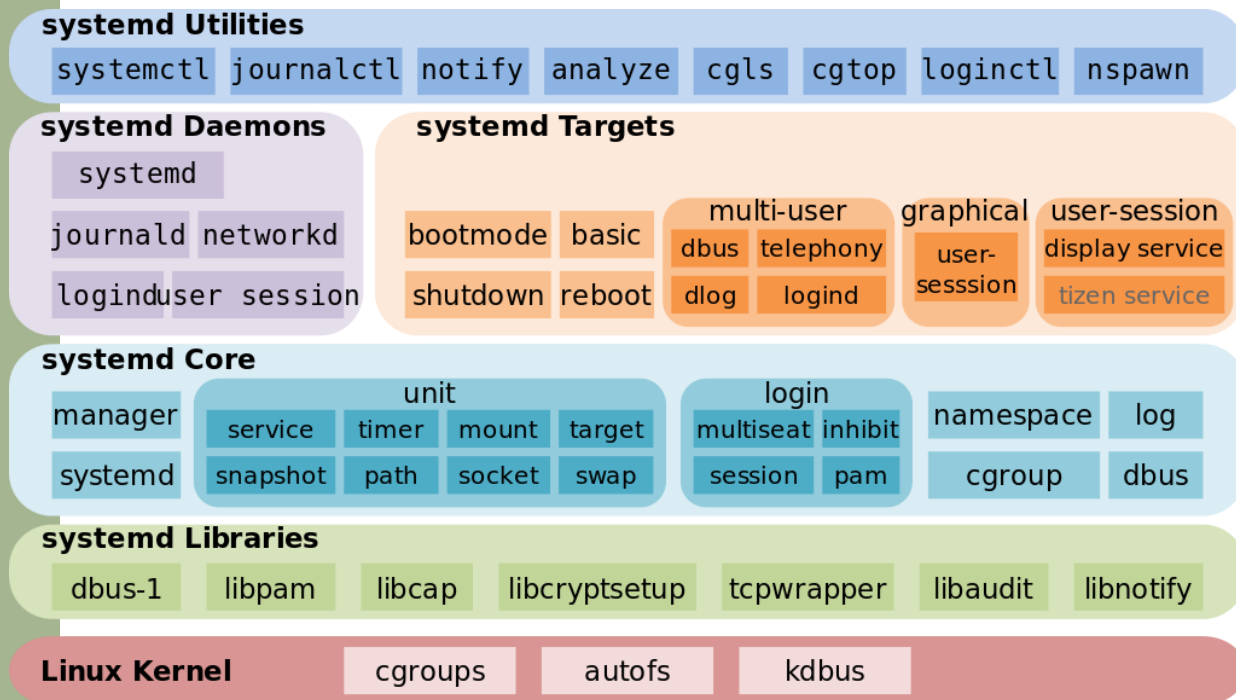
*)
    echo "Usage: /etc/init.d/sshd start|stop|restart"
    exit 1
    ;;
esac
```



systemd

Imagens [Wikipedia](#)
Shmuel Csaba Otto Traian

- Sistema criado pela RedHat e seguido por outros



Systemd (cont)

- Gere o sistema e os seus serviços
 - journald, logind, networkd
- Usa sockets e o D-Bus (comunicação de daemons)
- Suporta paralelização (inicialização de serviços)
- Suporta o SystemV (`init`) (ver [incompatibilidades](#))

- Críticos dizem que tem muita complexidade, monolítico, o que, dizem, vai contra a filosofia Unix

Systemd (cont)

- Ficheiros dos serviços:

`/etc/systemd/system/`

- Vários comandos (systemd-*)

- Exemplos de controlo:

```
# systemctl list-unit-files
```

```
# systemctl status firewalld.service
```

```
# service firewall status #usando o service System V
```

```
# systemctl status sshd
```

```
# systemctl cat sshd.service
```

Systemd (unidades)

Tipo unidade	Extensão Ficheiro	Descrição
Service	.service	A system service.
Target	.target	A group of systemd units.
Automount	.automount	A file system automount point.
Device	.device	A device file recognized by the kernel.
Mount	.mount	A file system mount point.
Path	.path	A file or directory in a file system.
Scope	.scope	An externally created process.
Slice	.slice	A group of hierarchically organized units that manage system processes.
Snapshot	.snapshot	A saved state of the systemd manager.
Socket	.socket	An inter-process communication socket.
Swap	.swap	A swap device or a swap file.
Timer	.timer	A systemd timer.

De [Managing Services with systemd](#)

Systemd (cont)

test_service.sh

```
1
2 DATE=`date '+%Y-%m-%d %H:%M:%S'`
3 echo "Example service started at ${DATE}" | systemd-cat -p info
4
5 while :
6 do echo "Looping...";
7 sleep 30;
8 Done
```

/lib/systemd/system/myservice.service

```
1 [Unit]
2 Description=Example systemd service.
3
4 [Service]
5 Type=simple
6 ExecStart=/bin/bash /usr/bin/test_service.sh
7
8 [Install]
9 WantedBy=multi-user.target
```

Systemd (yet another exemple for mount)

```
$ cat /etc/systemd/system/mnt-scratch.automount
```

```
[Unit]
```

```
Description=Automount Scratch
```

```
[Automount]
```

```
Where=/mnt/scratch
```

```
[Install]
```

```
WantedBy=multi-user.target
```

```
$ cat /etc/systemd/system/mnt-scratch.mount
```

```
[Unit]
```

```
Description=Scratch
```

```
[Mount]
```

```
What=nfs.example.com:/export/scratch
```

```
Where=/mnt/scratch Type=nfs
```

```
[Install]
```

```
WantedBy=multi-user.target
```

Systemd (yet another exemple for socket)

/root/remote-commands.sh:

```
#!/bin/bash
```

```
read MESSAGE
```

```
while [ "$MESSAGE" != "Q" ]
```

```
do
```

```
$MESSAGE
```

```
echo
```

```
read MESSAGE
```

```
done
```

```
echo "Byeeee"
```

/etc/systemd/system/rcmd@.service:

[Unit]

Description=Remote-command Per-Connection Server

[Service]

ExecStart=-/root/remote-commands.sh

StandardInput=socket

```
# systemctl enable rcmd.socket
```

```
# systemctl start rcmd.socket
```

```
# systemctl status rcmd.socket
```

/etc/systemd/system/rcmd.socket:

[Unit]

Description=Remote command for Per-Connection Servers

[Socket]

ListenStream=5900

Accept=yes

[Install]

WantedBy=sockets.target

Running the example:

```
# nc <server> 5900
```

```
pwd
```

```
ls
```

```
Q
```

```
Ctrl-D
```

Reiniciar

```
$ sudo shutdown -h +15 "Vamos desligar..."
```

```
$ sudo halt
```

- Faz o sync dos discos

```
$ sudo reboot
```

- Semelhante a halt, mas reinicia sistema

```
$ sudo shutdown -r
```


INIT E “PROCESSOS” DE KERNEL

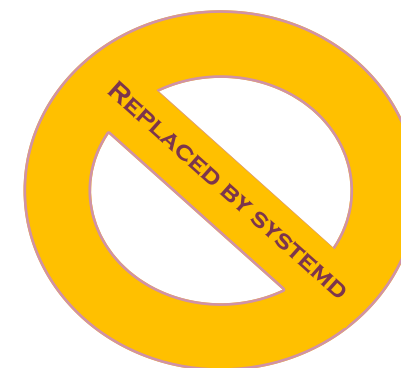
init (PID 1)

- Antepassado de todos os processos
- Responsável pelas seguintes atividades no arranque:
 - Verificar integridade dos sistemas de ficheiros.
 - Montar discos locais (/etc/fstab).
 - Iniciar áreas de swap.
 - Limpar e verificar sistemas de ficheiros.
 - Quotas, remoção de ficheiros temporários.
 - Arrancar rede e montar discos remotos.
 - Arrancar serviços (daemons) para:
 - Impressão, email, logging, cron, etc. (alguns como de log podem ser lançados antes)
 - Ativar mecanismos de login:
 - Getty
 - X11 (gdm, kdm, xdm).



init

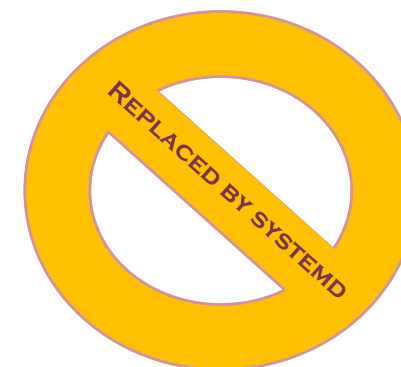
- Em Linux o `init` é configurado de modo a proceder a um arranque ao estilo “System V”.
- Num “System V” existem uma série de estados de execução (run levels) designados por um caracter, geralmente um número.
- Um sistema encontra-se num destes estados.
- O sistema muda de estado através de comandos administrativos.



Níveis de execução.

System V	Description	systemd
0	Halt the system.	poweroff.target
1	Single-user mode (for special administration).	rescue.target
2	Local Multiuser with Networking but without network service (like NFS)	multi-user.target
3	Full Multiuser with Networking	multi-user.target
4	Not Used	multi-user.target
5	Full Multiuser with Networking and X Windows(GUI)	graphical.target
6	Reboot.	reboot.target

De TLDP [Run Levels](#)



Ver:

```
ls -l /lib/systemd/system/runlevel[0-9].target
```

E [Working with systemd Targets](#)

[dcc]

“Processos” de kernel

- São partes do kernel que podem ser geridos como processos

Thread	O que faz
kswapd	Trata do swap dos processos quando a memória física está “cheia”
ksoftirqd	Trata de soft interrupts se não poderem ser tratados na mudança de contexto
System V	
kjournald	Faz o “commit” das atualizações ao sistema de “jornal” dos sist. de ficheiros
khubd	Configura dispositivos USB
Systemd	
kthreadd	Pai/mãe de todas as kernel threads. Gere os pedidos pelo userspace (modprobe, etc.)
ksmd	Daemon para detetar de-duplicação da memória de userspace

Modo single-user

- modo de manutenção
 - init coloca neste modo a pedido
- Para tarefas administrativas críticas necessitando uso exclusivo da máquina.
 - Ex.: Problemas com sistemas de ficheiros que não podem ser solucionados de forma automática.
- Como acionar:
 - passagem de um parâmetro especial de arranque ao kernel.
 - boot: linux single
 - A qualquer momento com:
 - ~~telinit 1~~
 - systemctl rescue

Resumo

- Fases do boot
- grub
- Arranque: SystemV e systemd
- Reiniciar
- init
- single-user

QUESTÕES/ COMENTÁRIOS