

Unix/Linux

ISEN - AP4

3 - Boot / shutdown

2019 - J. Hochart

Infos

Référence

Supports disponibles en ligne après le cours (j+1):

www.hochart.fr/LIN

Contact

Pour toute question sur le cours ou les exos:

jul@hochart.fr

Licence

Merci de ne pas publier / diffuser ces supports.

Références (3 - Boot / Shutdown)

Livres

- Essential System Administration (3eme Ed)
 - Auteur: Aeleen Frisch
 - Editions: O'Reilly

Web

- Master Boot Record - Wikipedia

Introduction

- Comme sur tous les systèmes, le démarrage et le shutdown sont en général **simples**
- Mais il est important de **comprendre comment cela fonctionne** pour *debugguer / customiser*

Bootstrapping

- *Bootstrap* = démarrer un système
- Processus similaire sur tous les unix

Chargement du kernel

1. Détection basique du matériel: disque, clavier, ...
2. Init du firmware
3. Localisation du programme initial de boot (endroit fixe sur le disque)
4. Localisation et exécution du kernel
5. Chargement des drivers et des modules
6. Le kernel lance init (PID 1)
7. Init lance les sous systèmes et les daemons

MBR (Wikipedia)

Le master boot record ou MBR (parfois aussi appelé « zone amorce ») est le nom donné au premier secteur adressable d'un disque dur (cylindre 0, tête 0 et secteur 1, ou secteur 0 en adressage logique) dans le cadre d'un partitionnement Intel. Sa taille est de 512 octets. Le MBR contient la table des partitions du disque dur. Il contient également une routine d'amorçage dont le but est de charger le système d'exploitation, ou le chargeur d'amorçage (boot loader) s'il existe, présent sur la partition active.

Plusieurs schémas de partitionnement: MBR, GPT,...

Kernel

- Le kernel est la partie de l'OS qui **tourne toujours**
- **Endroit fixe**, pour être trouvé par le programme d'amorçage

AIX	<i>/unix</i> (actually a link to a file in <i>/usr/lib/boot</i>)
FreeBSD	<i>/kernel</i>
HP-UX	<i>/stand/vmunix</i>
Linux	<i>/boot/vmlinuz</i>
Tru64	<i>/vmunix</i>
Solaris	<i>/kernel/genunix</i>

Suite / multi-user

1. Init est le **parent** des processus de login shell
2. Init vérifie l'**intégrité** des filesystems
3. Sous system V, / est monté en RO jusqu'à la fin des checks, puis remonté en RW. Sous BSD, c'est le kernel qui gère cela
4. **Montage des disques locaux**
5. Nettoyage de /tmp
6. Démarrage des **daemons**
7. Lancement des **mires** de login (via **getty** ou **gdm**)

Suite / multi-user

- Ces activités sont réalisées par des init scripts, qui sont organisés en 2 écoles possibles
 - SysV, --> **systemd**
 - BSD

Mode single-user

- Lorsqu' init prend la main, il peut mettre le système en mode single user au lieu de compléter les actions du processus précédemment décrit
- Mode utilisé pour **l'administration et la maintenance**
- **Option particulière passée au bootloader** ou touche spéciale pressée au boot

Mode single-user

- En singleuser, **init fork()** pour lancer **/bin/sh**, en tant que **root**
- L'admin est root mais **presque rien ne tourne**
- Sur les anciens unix, il suffisait d'avoir un accès physique à la machine pour booter en single-user
- Sur les unix modernes, il faut **entrer le mdp root** (activé ou désactivé selon les unix / distributions)

Single user avec grub sous linux

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

```
grub> kernel /vmlinuz=new ro root=/dev/hda2  
single
```

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

```
grub> boot
```

/boot

```
root@debian:/boot/grub# ls -al
total 1372
drwxr-xr-x 5 root root      4096 mars   13 15:51 .
drwxr-xr-x 3 root root      4096 mars   13 15:51 ..
drwxr-xr-x 2 root root      4096 mars   13 13:00 fonts
-r--r--r-- 1 root root     7633 mars   13 15:51 grub.cfg
-rw-r--r-- 1 root root     1024 mars   13 13:00 grubenv
drwxr-xr-x 2 root root     12288 mars   13 13:00 i386-pc
drwxr-xr-x 2 root root      4096 mars   13 13:00 locale
-rw-r--r-- 1 root root    1363161 mars   13 13:00 unicode.pf2
```

```
root@debian:/boot# ls -la
total 50456
drwxr-xr-x  3 root root      4096 mars   13 15:51 .
drwxr-xr-x 22 root root      4096 mars   13 12:56 ..
-rw-r--r--  1 root root    186589 févr. 19 10:05 config-4.9.0-8-amd64
-rw-r--r--  1 root root    186547 mars   13 15:00 config-4.9.162-amd64
drwxr-xr-x  5 root root      4096 mars   13 15:51 grub
-rw-r--r--  1 root root  18146373 mars   13 12:58 initrd.img-4.9.0-8-amd64
-rw-r--r--  1 root root  18243933 mars   13 15:51 initrd.img-4.9.162-amd64
-rw-r--r--  1 root root   3196808 févr. 19 10:05 System.map-4.9.0-8-amd64
-rw-r--r--  1 root root   3191295 mars   13 15:00 System.map-4.9.162-amd64
-rw-r--r--  1 root root   4241184 févr. 19 10:05 vmlinuz-4.9.0-8-amd64
-rw-r--r--  1 root root   4241184 mars   13 15:00 vmlinuz-4.9.162-amd64
```

/boot

- Contient
 - Les fichiers du boot loader (Ici Grub)
 - Vmlinuz: Le noyau linux = un executable compilé statiquement
 - Initrd.img: ramdrive de boot
 - System.map: La table des symboles du kernel

vmlinux

- Originellement /boot/unix
- Vm: prefix ajouté lors du support de la mémoire virtuelle
- Z: suffixe indiquant que c'est une image compressée

initrd.img

- Image d'un FS temporaire qui sera monté en RAM avant /. Contient ce qui est nécessaire (modules) pour trouver et monter /
 - Très custom: debian, ubuntu
 - Générique: redhat
- On peut toujours compiler un kernel dont le boot ne necessite pas d'initrd
 - Compiler les drivers / modules nécessaire pour monter / dans le kernel
 - Le device (après load du driver) de / doit avoir un nom fixe

System map

- Map = tableau de correspondance entre des noms (symboles) et des adresses en mémoire (méthodes, fonctions, variables)
- Sert principalement au debug quand le kernel crash
- Peut normalement être delete

Grub

- **Grub.cfg**: Fichier de conf lu une fois le stage 1 passé et / monté
- **Grubenv**: Grub n'écrit pas sur le disque, sauf dans ce fichier (préalloué 1024o pour persistance)

> b	<i>Initiate boot to multiuser mode.</i>
Urizen Ur-Unix boot in progress...	
testing memory	<i>Output from boot program.</i>
checking devices	<i>Preliminary hardware tests.</i>
loading vmunix	<i>Read in the kernel executable.</i>
Urizen Ur-Unix Version 17.4.2: Fri Apr 24 23 20:32:54 GMT 1998	
Copyright (c) 1998 Blakewill Computer, Ltd.	<i>Copyright for OS.</i>
Copyright (c) 1986 Sun Microsystems, Inc.	<i>Subsystem copyrights.</i>
Copyright (c) 1989-1998 Open Software Foundation, Inc.	
...	
Copyright (c) 1991 Massachusetts Institute of Technology	
All rights reserved.	<i>Unix kernel is running now.</i>
physical memory = 2.00 GB	<i>Amount of real memory.</i>
Searching SCSI bus for devices:	<i>Peripherals are checked next.</i>
rdisk0 bus 0 target 0 lun 0	
rdisk1 bus 0 target 1 lun 0	
rdisk2 bus 0 target 2 lun 0	
rmt0 bus 0 target 4 lun 0	
cdrom0 bus0 target 6 lun 0	
Ethernet address=8:0:20:7:58:jk	<i>Ethernet address of network adapter.</i>

Root on /dev/disk0a
Activating all paging spaces
swapon: swap device /dev/disk0b activated.
Using /dev/disk0b as dump device

*Indicates disk partitions used as /,...
...as paging spaces and...
...as the crash dump location.*

INIT: New run level: 3

*Single-user mode could be entered here,...
...but this system is booting to run level 3.
Messages produced by startup scripts follow.
Means "Be patient."*

The system is coming up. Please wait.
Tue Jul 14 14:45:28 EDT 1998

Checking TCB databases
Checking file systems:
fsstat: /dev/rdisk1c (/home) umounted cleanly;
fsstat: /dev/rdisk2c (/chem) dirty
Running fsck:
/dev/rdisk2c: 1764 files, 290620 used, 110315 free
Mounting local file systems.

*Verify integrity of the security databases.
Check and mount remaining local filesystems.
Skipping check.
This filesystem needs checking.*

Checking disk quotas: done.
cron subsystem started, pid = 3387
System message logger started.
Accounting services started.

Daemons for major subsystems start first,...

...followed by network servers,...

Network daemons started: portmap inetd routed named rhwod timed.
NFS started: biod(4) nfsd(6) rpc.mountd rpc.statd rpc.lockd.
Mounting remote file systems.
Print subsystem started.
sendmail started.

...and network-dependent local daemons.

Preserving editor files.
Clearing /tmp.
Enabling user logins.
Tue Jul 14 14:47:45 EDT 1998

*Save interrupted editor sessions.
Remove files from /tmp.
Remove the /etc/nologin file.
Display the date again.*

Urizen Ur-Unix 9.1 on hamlet

The hostname is hamlet.

login:

Unix is running in multiuser mode.

BSD: rc.conf

- `accounting_enable="YES"`
- `check_quotas="YES"`
- `defaultrouter="192.168.29.204"`
- `hostname="ada.ahania.com"`
- `ifconfig_xl0="inet 192.168.29.216 netmask 255.255.255.0"`
- `inetd_enable="YES"`
- `nfs_client_enable="YES"`
- `nfs_server_enable="YES"`
- `portmap_enable="YES"`
- `sendmail_enable="NO"`
- `sshd_enable="YES"`

SYSV: runlevels

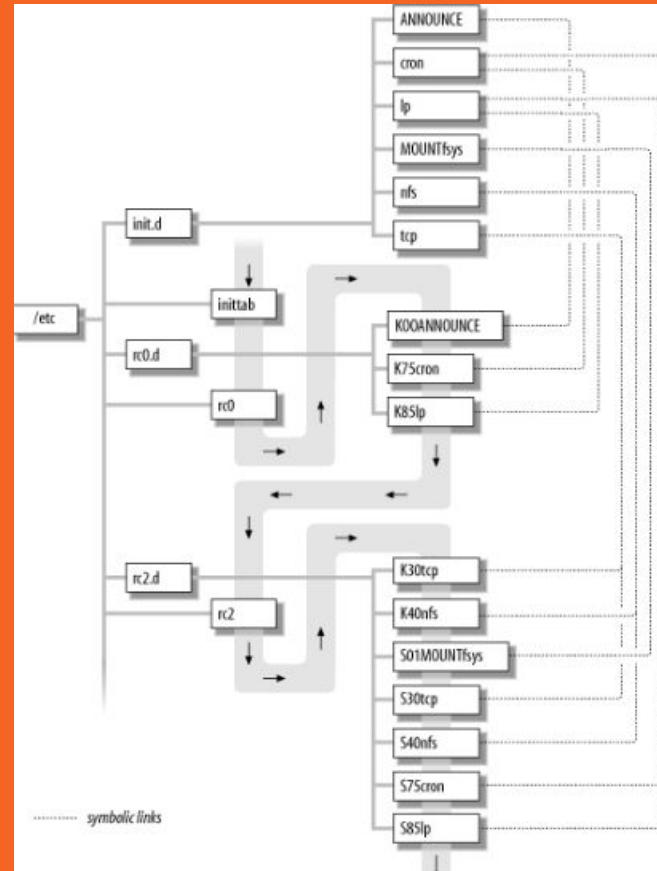
- 0 – Halt
- 1 – Single-user text mode
- 2 – Not used (user-definable)
- 3 – Full multi-user text mode
- 4 – Not used (user-definable)
- 5 – Full multi-user graphical mode (with an X-based login screen)
- 6 – Reboot

```
$ who -r  
. run level 3  Mar 14 11:14  3  0  S
```

Who -r

SysV Init scripts

- Orchestré par /etc/inittab



Systemd

- Systemd est le parent de tous les process, avec un pid de 1
- Lance les sous-processes plus tôt et en parallèle (boot plus court)
- Plein de nouveaux avantages (et d'inconvénients)

- **Systemctl start apache2**
- **Systemctl stop apache2**
- **Systemctl status apache2**
- **Systemctl enable apache2**
- **Systemctl disable apache2**
- **Systemctl restart apache2**
- **Systemctl is-enabled apache2**

Exemple de création d'un service systemd

- `/etc/systemd/system/mydaemon.service`

[Unit]

Description=Boot mydaemon

After=syslog.target network.target remote-fs.target nss-lookup.target

[Service]

Type=forking

PIDFile=/run/mydaemon.pid

ExecStartPre=/usr/bin/mydaemon validateconf -quiet

ExecStart=/usr/bin/mydaemon argstart

ExecStop=/bin/kill -s QUIT \$MAINPID

RemainAfterExit=yes

[Install]

WantedBy=multi-user.target