Chapter 6
Boot time configuration

ict@innovation

FOSSFA

Chapter 6 Outline

- In this chapter we will learn about:

  - ✓ How the system boots

  - ✓ How to configure the boot loaders LILO and GRUB

  - ✓ How to intervene in the boot process

  - ✓ How services are started up

The boot process

- The boot process

    - The linux boot process
    - Linux boot loaders
    - Booting with LILO
    - Booting with GRUB
    - Initial ram disk image
    - Exploring the initial ram disk image

ict@innovation

The linux boot process

- The early stages of the boot sequence are dependent on the machine architecture. We consider the common case of an Intel-based PC

- Initially the BIOS (in firmware) loads the first stage boot loader from the master boot record (MBR) of the first drive

- First stage boot loader loads second stage from the `/boot` partition

    - Relies on the BIOS for disk I/O and a prebuilt block map to find the file

- Loader presents a choice of bootable images for user selection

- Loader loads the selected kernel image (typically `/boot/vmlinuz`)

- Kernel initialised

    - Probes hardware and loads driver modules

    - Mounts the root partition as '/'

    - Runs the program `/sbin/init`

ict@innovation

FOSSFA

Linux boot loaders

- Linux has two widely used boot loaders
    - Both are also capable of loading the boot loader of non-linux systems
- LILO (LInux LOader)
    - LILO is relatively simple and cannot directly decode the linux filesystem
    - Relies on a pre-installed map (stored in the MBR) to find its boot-time configuration data and the kernel image(s) it is expected to boot
    - Map must be re-installed if the boot-time configuration choices change
- GRUB (GRand Unified Boot loader)
    - More flexible at boot time; has its own command 'shell'
    - Second stage can decode the linux filesystem; finds and reads its config file (`/boot/grub/menu.lst`) by itself at boot time
    - Also finds the linux kernel image by name, within the filesystem
    - GRUB is the default for SuSE linux
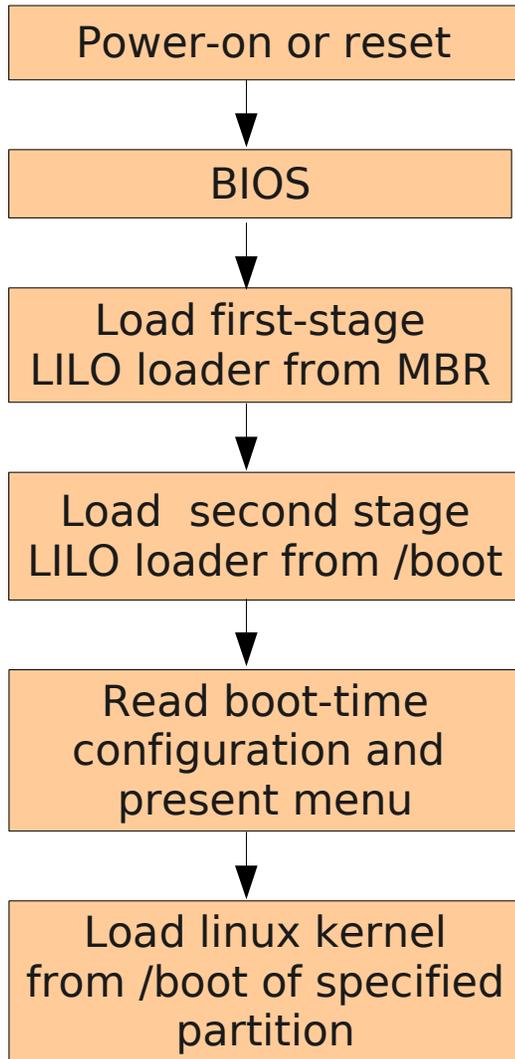
ict@innovation

FOSSFA

Booting with LILO

Power-on or reset

↓

BIOS ← Firmware settings control which devices (and in which order) BIOS will search for a valid MBR. e.g. CD, floppy, hard drive

↓

Load first-stage LILO loader from MBR

↓

Load second stage LILO loader from /boot

↓
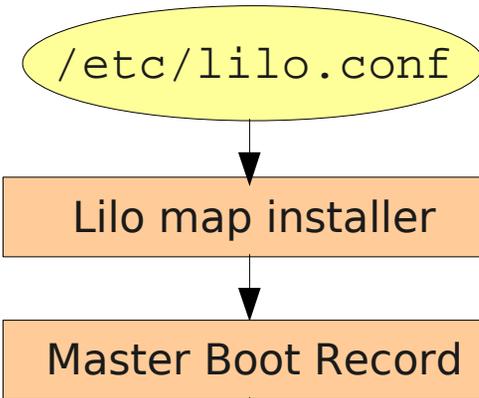
Read boot-time configuration and present menu ← LILO reads boot-time data from a map previously written to the MBR by a map installer (confusingly also called lilo)

↓

Load linux kernel from /boot of specified partition

↓

To be continued ...

/etc/lilo.conf

↓

Lilo map installer

↓

Master Boot Record

↓

ict@innovation

FOSSFA

Booting with GRUB

```
Power-on or reset
```
↓
```
BIOS
```
← Firmware settings control which devices (and in which order) BIOS will search for a valid MBR. e.g. CD, floppy, hard drive
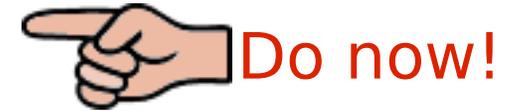↓
```
Load first-stage
GRUB loader from MBR
```
↓
```
Load  second stage
GRUB loader from /boot/grub
```

/boot/grub/menu.lst

↓
```
Read boot-time
configuration and
present menu
```
← GRUB understands the file system sufficiently to find and read its configuration file directly at boot time
↓
```
Load linux kernel
from /boot of specified
partition
```
↓
To be continued …

ict@innovation

FOSSFA

Initial ram disk image

- Some of the functionality of the linux kernel is in dynamically loaded modules

    - The core kernel image (`vmlinuz`) may not contain all the functionality needed to access the root file system

- A small filesystem image is loaded into memory by the boot loader and its address is passed to the kernel

    - The image is stored as a compressed file, usually `/boot/initrd`

    - The kernel decompresses and mounts this image, then runs a shell script (`linuxrc`) to load the modules it needs from the ramdisk

    - The kernel then unmounts the ram disk and mounts the real root partition

- Type `mkinitrd -h` for information on building the ram disk image

- A ram disk image is also used to support initial installation of SuSE linux

ict@innovation

FOSSFA

Exploring the initial ram disk image

- Log in as root, and run these commands: Do now!

  ```
  # cd /boot
  # ls -l
  # cp initrd temp.gz
  # gunzip temp.gz
  # ls -l
  # mount -o loop temp /mnt
  # ls /mnt
  # cat /mnt/linuxrc
  # umount /mnt
  # rm temp
  ```

Note the file `initrd` (it may be a symlink)

Now you should have a (much larger) file called `temp` containing the filesystem image

Mount the ramdisk image

Examine its contents

This is the script that the kernel runs to load modules from the ramdisk. Which module(s) does it load?

-

ict@innovation

Boot loader configuration

- Boot loader configuration

  Configuring LILO

  A sample `/etc/lilo.conf` file

  Configuring GRUB

  GRUB device naming

  The `menu.lst` file

  The GRUB command shell

ict@innovation

Configuring LILO

- There are two programs both known as "lilo"

  - The boot loader

  - The map installer, `/sbin/lilo`, which reads the configuration file `/etc/lilo.conf` and writes the boot map into the MBR

- `/etc/lilo.conf` controls the boot time configuration

  - Global parameters

  - Details of images available for booting

  - See `man lilo.conf` for full details

- The command `lilo` must be re-run if this file is changed, to re-install the map into the MBR

ict@innovation

A sample `/etc/lilo.conf` file

```
boot=/dev/hda

   prompt

   timeout=100

   read-only
```

Where to write the boot sector

Prompt the user to select which image to boot and wait 100 x 0.1 seconds before timing out and defaulting to the first image

Specifies that the root file system should initially be mounted read-only

```
image=/boot/vmlinuz

   label=linux

   root=/dev/hda1

   initrd=/boot/initrd

other=/dev/hda3

   label=windows
```

File name of bootable linux image

Name of image in boot-time menu

Device to mount as root file system

File name of initial ram disk image

This entry allows user to launch the boot loader of a non-linux system

ict@innovation

Configuring GRUB

- GRUB has two configuration files

- `/etc/grub.conf`

  - This file controls the installation of the MBR. The SuSE installation tool constructs and invokes this file; you are unlikely to have to change it

- `/boot/grub/menu.lst`

  - This file is consulted at boot time

  - Defines the grub commands needed to boot each of the operating systems that can be selected

  - The SuSE installation tool will construct this file based on the bootable images it finds on the hard drive

  - You will need to change it if:

    - You frequently want to boot your kernel with different options
    - You build a new kernel under a new name
    - You install an additional operating system and wish to dual boot

ict@innovation

GRUB device naming

- GRUB has its own scheme for naming disk devices
  - It does not distinguish IDE and SCSI disks
  - Drives are simply enumerated in the order they are detected by the BIOS
- For example, on a machine with two IDE drives and one SCSI drive:
  - Grub name `hd0` corresponds to linux IDE drive `/dev/hda`
  - Grub name `hd1` corresponds to linux IDE drive `/dev/hdb`
  - Grub name `hd2` corresponds to linux SCSI drive `/dev/sda`
- Partitions are numbered starting at zero, for example:
  - `(hd0,0)` first primary partition on first hard drive
  - `(hd0,1)` second primary partition on first hard drive
  - `(hd0,4)` first logical partition
  - `(hd0,5)` second logical partition, and so on …

ict@innovation

The `menu.lst` file

```
gfxmenu (hd0,0)/boot/message
color white/blue black/light-gray
default 0
timeout 8

title linux
    kernel (hd0,0)/boot/vmlinuz root=/dev/hda1  hdc=ide-scsi  vga=788
    initrd (hd0,0)/boot/initrd

title floppy
    root (fd0)
    chainloader +1

title failsafe
    kernel (hd0,0)/boot/vmlinuz.shipped root=/dev/hda1 ide=nodma
         apm=off acpi=off vga=normal nosmp disableapic maxcpus=0 3
    initrd (hd0,0)/boot/initrd.shipped
```

Where to find the splash screen background

Colour scheme for text interface
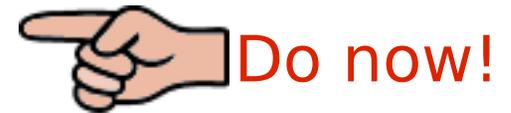
Which O/S to boot if user does not make a selection

Number of seconds to wait for user to make a selection

Parameters to pass to kernel

ict@innovation

The GRUB command shell

- The entries in the `menu.lst` file are actually GRUB shell commands

- Root can run the GRUB shell directly within linux

 Do now!

```
# grub

grub> help                          ← List all available commands

grub> help kernel                   ← Get detailed help on one command

grub> cat (hd0,0)/etc/fstab         ← Examine a file (note the
                                       GRUB partition name)
grub> quit

#
```

- More usefully, you can interrupt the boot sequence and run the GRUB shell *prior to booting linux*

ict@innovation

Intervening in the boot process

- Intervening in
  the boot process

  Interrupting the boot sequence

  Booting single user

  Rescuing a system that won't boot

  Rescue boot

  Kernel message buffer

ict@innovation

FOSSFA

Interrupting the boot sequence

- Without user intervention, GRUB will time out and boot the default selection

- To interrupt the boot sequence, press ESC (followed by ENTER) when you see the GRUB splash screen

  - This takes you to GRUB's main text menu. From here you can:

    - Type 'c' to get to the GRUB command shell

    - Use up and down arrows to select the O/S you want to boot

    - Type 'e' to edit the commands used to boot the selected O/S

    - Type ENTER to boot the selected O/S

- GRUB allows a lot of flexibility

  - Boot kernel images without an entry in menu.lst

  - Boot kernels with user-suppplied parameters

    - For example, to boot into "single user" mode

ict@innovation

Booting single user

- To boot the system in single user (maintenance) mode, you need to append the argument 'single' to the kernel command
    - Interrupt the grub splash screen and enter the main text menu
    - Use the arrow keys to select the system you want to boot
    - Type 'e' from the main text menu to enter edit mode
    - Select the line you need to edit (the "kernel" line)
    - Type 'e' to edit the line
    - Add a space followed by the word 'single' to the end of the line
    - Press ENTER, then press 'b' to boot
- The kernel will initialise
    - You will be asked for the root password, then a shell will be started
    - No other user processes will be run

ict@innovation

FOSSFA

Booting single user (continued)

- Log out, and reboot the computer Do now!

- Following the instructions on the previous slide, interrupt the boot sequence and boot single user

  - Supply the root password when requested

  - Run the command `ps ax`

  - How many user processes are running? _____

  - Run the command `mount`

  - Which file systems are mounted? _____

- A common use for single-user mode is to check and/or repair filesystems while they are unmounted

  - Check the `/new1` filesystem with the command:

  `# fsck /dev/hda5`

- Run the command `reboot` and allow the machine to boot normally

ict@innovation
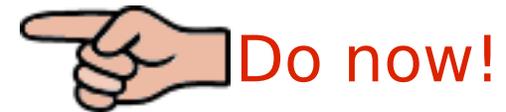
Rescuing a system that won't boot

- Various factors will cause a system to fail to boot, for example:
    - Missing or corrupt kernel image
    - Missing or corrupt ram disk image
    - Badly-configured kernel
    - Corrupt root file system
    - Missing or incorrect `/etc/fstab`
    - Kernel hangs trying to initialise faulty hardware
- Press F2 during boot to see the boot-time messages
    - Try to determine what the *first* error message is
    - How far had the boot process progressed?
        - e.g. Has the root file system been mounted?
    - What was the kernel trying to do when it failed?

ict@innovation

Rescue boot

- For PCs that will boot from CD, the SLES installation CD can be used to perform a *rescue boot*

  - Linux kernel is booted from CD

  - Modest set of tools available in ram disk loaded from image on CD

  - Can run `fsck` to check or repair the filesystems of the target system

  - Can mount the partitions of the target system to check or repair key configuration files

ict@innovation

Exercise: Rescue boot

Please note: This exercise deliberately creates a situation in which the system will not boot. Please follow the instructions carefully to be sure that a working system can be restored!

1. Log in as root and do the following:

```
# cd /boot
# mv initrd initrd.xxx
```

2. Try to reboot. What happens?

3. Boot the computer from the SLES8 installation CD. When you see the splash screen, press the space bar to interrupt the boot process and select "Rescue System" from the menu

   – A kernel will be booted from the CD

4. Select your keyboard layout when requested

   – A ram disk image will be loaded from CD

ict@innovation

Exercise: Rescue boot (continued)

5. When the kernel finishes booting, log in as root at the "Rescue login" prompt. There is no password

6. List the directories `/`, `/sbin`, `/bin` and `/usr/bin`

    – This will give you an idea of the commands available on the ram disk

    – Note that the hard disk has *not* been accessed yet

7. Run the command

    `# fdisk -l /dev/hda`

    – Can you see the partitions on the hard drive?

8. Mount the root directory of the target system with the command

    `# mount /dev/hda1 /mnt`

9. cd to the `/boot` directory of the target system (i.e. `/mnt/boot`)

    – List the directory. Can you see the `initrd.xxx` file?

ict@innovation

Exercise: Rescue boot (continued)

10. Rename the `initrd.xxx` file back to `initrd`

    – Be sure it's left in the right directory

11. Eject the CD

12. Run the command `reboot`

13. Allow the machine to boot without intervention

    – Does the system reboot successfully now?

    – If not ... ask your instructor for help!

End of Exercise

ict@innovation

Kernel message buffer

- As the kernel boots it writes a trace of its actions to an internal buffer

  - Can be a useful source of diagnostic information

- The command `dmesg` displays the kernel's message buffer

  - A portion of the output is shown below

```
# dmesg
Linux version 2.4.21-99-default (root@i386.suse.de)
Kernel command line: root=/dev/hda3 vga=0x317 desktop hdc=ide-scsi
hdclun=0 splash=silent
ide_setup: hdc=ide-scsi
ide_setup: hdclun=0
bootsplash: silent mode.
Initializing CPU#0
Detected 1595.659 MHz processor.
Console: colour dummy device 80x25
Calibrating delay loop... 3145.72 BogoMIPS
Memory: 513932k/523776k available (1590k kernel code, 9392k reserved,
605k data, 160k init, 0k highmem)
```

ict@innovation

FOSSFA

Service startup

- Service startup
  - The init process
  - Run levels
  - Changing the default run level
  - Quiz

ict@innovation

FOSSFA

The `init` process

- When the kernel has finished booting it creates a single user process

    - Process ID 1

    - Runs the program `/sbin/init`

- `init` is (directly or indirectly) the ancestor of every other process in the system

    - Its actions are controlled by the file `/etc/inittab`

Run levels

- The set of services started by `init` depends on the *run level* that it enters

  - Conventionally the run levels are configured as follows

Run Level	Action
0	Halt
1	Single-user (maintenance) – the only user process is a root shell
2	Mutli-user without networking
3	Full multi-user with networking
4	Not used
5	Multi-user plus X windows
6	Reboot

- The run level that the system boots into is determined by the `initdefault` entry in `/etc/inittab`

ict@innovation

FOSSFA

Changing the default run level

- Log in as root

Do now!

- Edit the file `/etc/inittab`

  - Find the line containing the keyword "initdefault"

  - Change the value from 3 to 5

  - Save the file

- Reboot the machine

  - Verify that it now boots directly into a graphical login

ict@innovation

Quiz

- Name the two main boot loaders used by linux

- Describe the purpose of the file `/boot/initrd`

- In single user mode, how many user processes are running?

- In single user mode, how many file systems are mounted?

- On a machine with two IDE drives and no SCSI drives, by what name would GRUB refer to the first primary partition on the second drive?

- True or false?

  - If the GRUB configuration file `/boot/grub/menu.lst` is changed, you must re-run grub to update the MBR

  - If the LILO configuration file `/etc/lilo.conf` is changed, you must re-run lilo to update the MBR

  - LILO can boot a kernel that is not listed in its config file

  - GRUB can boot a kernel that is not listed in its config file

ict@innovation

FOSSFA