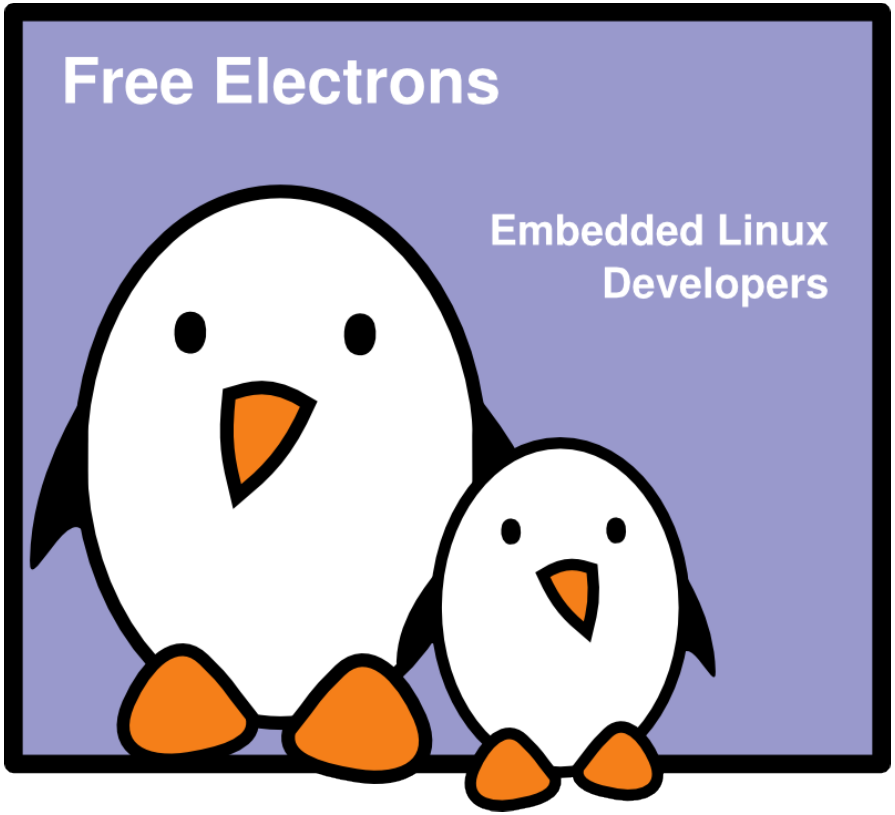
The GRUB bootloader

Michael Opdenacker
Thomas Petazzoni
Free Electrons



Rights to copy

Document sources, updates and translations:
http://free-electrons.com/docs/grub

Corrections, suggestions, contributions and translations are welcome!

Latest update: Sep 15, 2009

Grub features (1)

- Many features and a lot of flexibility!
- Supports booting many operating systems: Linux, Hurd, *BSD, Windows, DOS, OS/2...
- Support for different boot devices: hard disk (of course), cdrom (El Torito), network (tftp)
- Support for many filesystems (unlike LILO, it doesn't need to store the physical location of each kernel): ext2/3, xfs, jfs, reiserfs, dos, fat16, fat32...
- Configuration file: unlike LILO, no need to update the MBR after making changes to the configuration file.

Grub features (2)

- Support for many network cards
  (reusing drivers from the Etherboot bootloader).
- Menu interface for regular users.
  Advanced command line interface for advanced users.
- Remote control from a serial console.
- Supports multiple executable formats:
  ELF by also a.out variants.
- Can uncompress compressed files
- Small: possible to remove features and drivers
  which are not used (`./configure --help`).
  Without recompiling: remove unused filesystem stages.

Grub size

Example from grub 0.97-1ubuntu9 (Ubuntu Dapper):

- Stage 1:
  `/lib/grub/i386-pc/stage1`: 512 bytes
- Stage 1.5:
  `/lib/grub/i386-pc/e2fs_stage1_5`: 7508 bytes
- Stage 2:
  `/lib/grub/i386-pc/stage2`: 105428 bytes

Total: only 113448 bytes!

Installing grub (1)

Install Grub on an embedded target with a blank disk.

- Do it from a GNU/Linux host with Grub installed.
- Access the disk for the embedded target as external storage:
  - Compact Flash disk: use a USB CF card reader.
  - Hard disk drive: use a USB hard disk drive enclosure.
- Create a partition on this disk (useful, but not mandatory):
  `fdisk /dev/sda` (type `m` for a menu of commands)
- Format and mount this partition:

```
mkfs.ext3 /dev/sda1
sudo mount /dev/sda1 /mnt/sda1
```

Installing grub (2)

- Install Grub:
  `grub-install --root-directory=/mnt/sda1 /dev/sda`
  - `/dev/sda`: the physical disk. Grub is installed on its Master Boot Record.
  - `/mnt/sda1`: the directory under which `grub-install` creates a `boot/` directory containing the upper stage and configuration file. Of course, you could have used another partition.
- Grub now needs a kernel to boot. Copy a kernel image to `/mnt/sda1/boot/` (for example) and describe this kernel in `/mnt/sda1/boot/grub/menu.lst`.
- Once you also copied root filesystem files, you can put your storage device back to the embedded target and boot from it.

Naming files

- Grub names partitions as follows: `(hd`*`n`*`,`*`p`*`)`
  n: *n*th disk on the system
  p: *p*th partition on this disk
- Files are described with the partition they belong to.
  Example: `(hd0,2)/boot/vmlinuz-2.6.18`
- You can specify a default partition with the `root` command:
  Example:

```
root (hd0,0)
kernel /boot/vmlinuz-2.6.18
```

Sample configuration file

/boot/grub/menu.lst

```
default 0
timeout 10

title      Ubuntu, kernel 2.6.15-27-386
root       (hd0,2)
kernel     /boot/vmlinuz-2.6.15-27-386 root=/dev/hda3 ro quiet splash
initrd     /boot/initrd.img-2.6.15-27-386
boot

title      Ubuntu, kernel 2.6.15-27-386 (recovery mode)
root       (hd0,2)
kernel     /boot/vmlinuz-2.6.15-27-386 root=/dev/hda3 ro single
initrd     /boot/initrd.img-2.6.15-27-386
boot
```

Network support

Grub can use the network in several ways

- Grub running from disk (floppy, hard drive, cdrom), and downloading kernel images from a tftp server on the network.
- Diskless system:
  - A first stage bootloader (typically Etherboot) is booted from ROM.
  - It then downloads a second stage from Grub: `pxegrub` for a PXE ROM, or `nbgrub` for a NBI loader).
  - Grub can then get kernel images from the network.

Grub security (1)

- Caution: the Grub shell can be used to display any of your files!
- Example:
  - Boot your system
  - Type the `c` command to enter command line mode.
  - `find /etc/passwd`
    Grub displays all partitions containing such a file.
  - `cat (hd0,2)/etc/passwd`
    You can see the names of users on the system!
    Of course, you can access any file. Permissions are ignored.

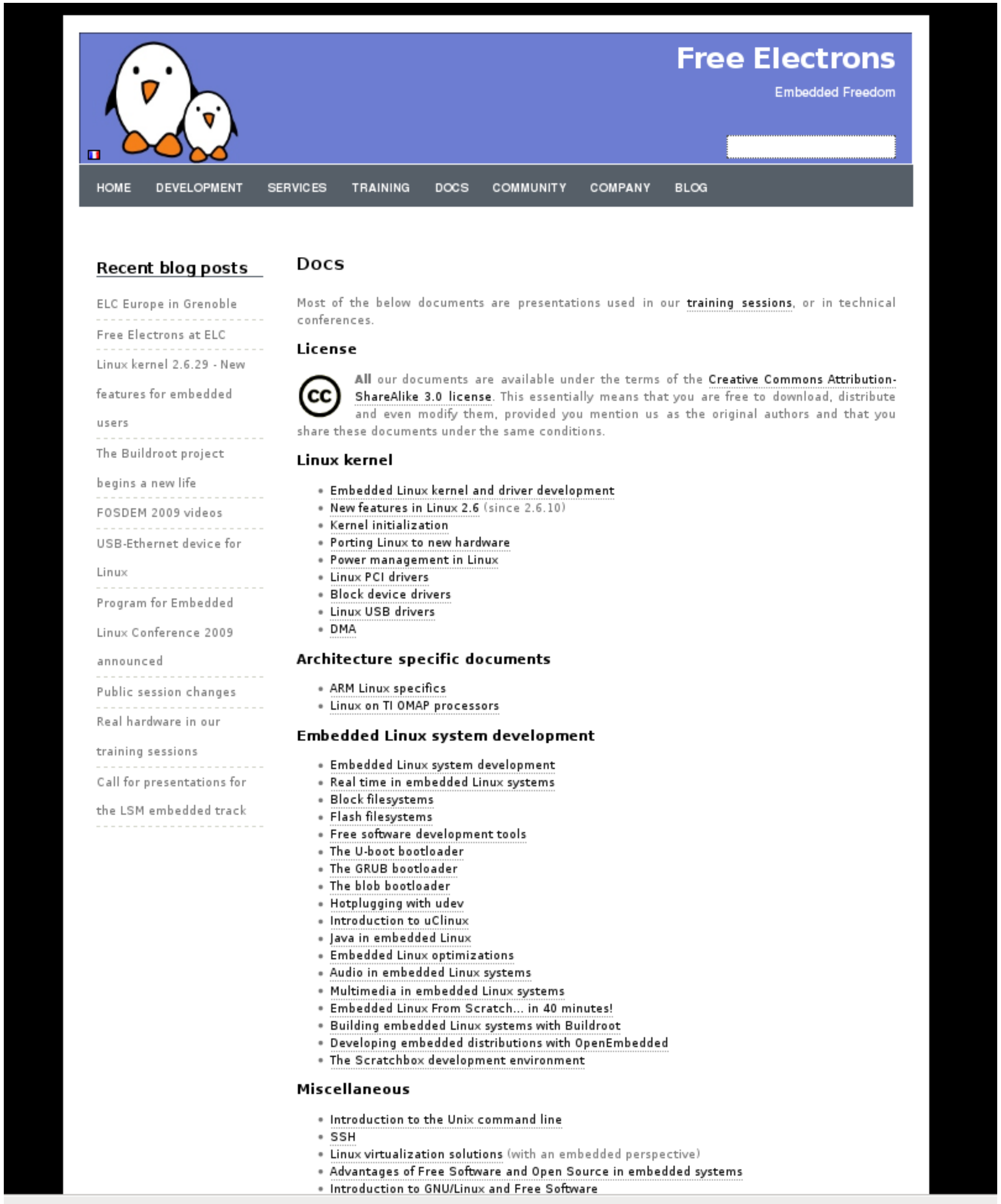
Grub security (2)

- Interactive commands can be protected with a password. Otherwise, people would even be able to view the contents of files from the Grub shell!
- You can also protect menu entries with a password. Useful to restrict failsafe modes to admin users.

Grub resources

- Grub home page:
  http://www.gnu.org/software/grub/
- Grub manual:
  http://www.gnu.org/software/grub/manual/

Related documents



Recent blog posts

- ELC Europe in Grenoble
- Free Electrons at ELC
- Linux kernel 2.6.29 - New features for embedded users
- The Buildroot project begins a new life
- FOSDEM 2009 videos
- USB-Ethernet device for Linux
- Program for Embedded Linux Conference 2009 announced
- Public session changes
- Real hardware in our training sessions
- Call for presentations for the LSM embedded track

Docs

Most of the below documents are presentations used in our training sessions, or in technical conferences.

License

All our documents are available under the terms of the Creative Commons Attribution-ShareAlike 3.0 license. This essentially means that you are free to download, distribute and even modify them, provided you mention us as the original authors and that you share these documents under the same conditions.

Linux kernel

- Embedded Linux kernel and driver development
- New features in Linux 2.6 (since 2.6.10)
- Kernel initialization
- Porting Linux to new hardware
- Power management in Linux
- Linux PCI drivers
- Block device drivers
- Linux USB drivers
- DMA

Architecture specific documents

- ARM Linux specifics
- Linux on TI OMAP processors

Embedded Linux system development

- Embedded Linux system development
- Real time in embedded Linux systems
- Block filesystems
- Flash filesystems
- Free software development tools
- The U-boot bootloader
- The GRUB bootloader
- The blob bootloader
- Hotplugging with udev
- Introduction to uClinux
- Java in embedded Linux
- Embedded Linux optimizations
- Audio in embedded Linux systems
- Multimedia in embedded Linux systems
- Embedded Linux From Scratch... in 40 minutes!
- Building embedded Linux systems with Buildroot
- Developing embedded distributions with OpenEmbedded
- The Scratchbox development environment

Miscellaneous

- Introduction to the Unix command line
- SSH
- Linux virtualization solutions (with an embedded perspective)
- Advantages of Free Software and Open Source in embedded systems
- Introduction to GNU/Linux and Free Software

All our technical presentations on http://free-electrons.com/docs

- Linux kernel
- Device drivers
- Architecture specifics
- Embedded Linux system development

How to help

You can help us to improve and maintain this document...

- By sending corrections, suggestions, contributions and translations
- By asking your organization to order development, consulting and training services performed by the authors of these documents (see http://free-electrons.com/).
- By sharing this document with your friends, colleagues and with the local Free Software community.
- By adding links on your website to our on-line materials, to increase their visibility in search engine results.

Free Electrons

Our services

Custom Development

System integration
Embedded Linux demos and prototypes
System optimization
Application and interface development

Consulting and technical support

Help in decision making
System architecture
System design and performance review
Development tool and application support
Investigating issues and fixing tool bugs

Linux kernel

Linux device drivers
Board support code
Mainstreaming kernel code
Kernel debugging

Embedded Linux Training

All materials released with a free license!

Unix and GNU/Linux basics
Linux kernel and drivers development
Real-time Linux, uClinux
Development and profiling tools
Lightweight tools for embedded systems
Root filesystem creation
Audio and multimedia
System optimization