Programowania niskopoziomowe
Bootloader

Dawid Górka
WFiIS

Czym jest bootloader

Bootloader czyli program rozruchowy, to program uruchamiany jako pierwszy po zakończeniu początkowego programu BIOS-u. Jego zadaniem jest załadowanie systemy operacyjnego do pamięci operacyjnej.

Design bootloadera

- Bootloader jednostopniowy
- Bootloader dwustopniowy
- Bootloader mieszany

Możliwe sposoby sposoby bootowania:

- z dysku twardego
- z dyskietki
- z płyty CD
- z dysku USB
- przez sieć LAN

Multi-booting

O mutli-bootingu mówimy kiedy na urządzeniu jest zainstalowane więcej niż jeden system operacyjny.

Programy rozruchowe:

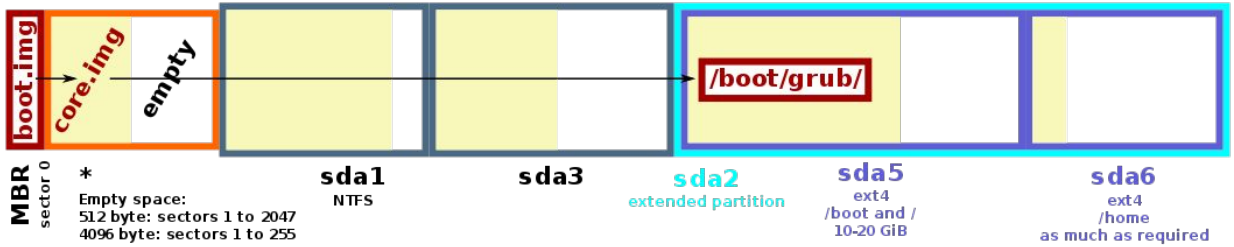
- Boot Camp
- GNU GRUB
- LILO
- SYSLINUX
- NTLDR

GRUB

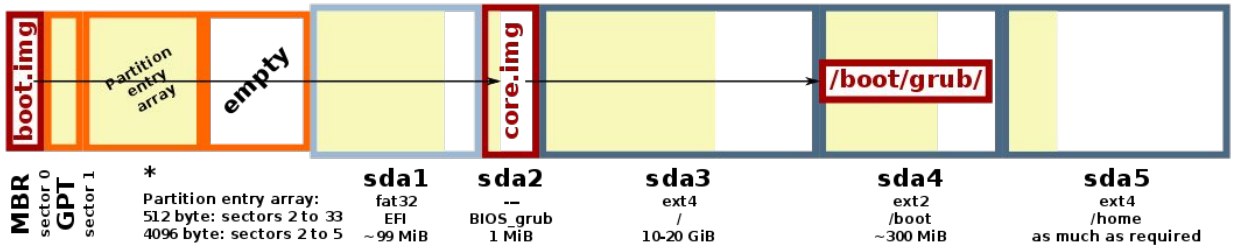
GNU GRUB 2

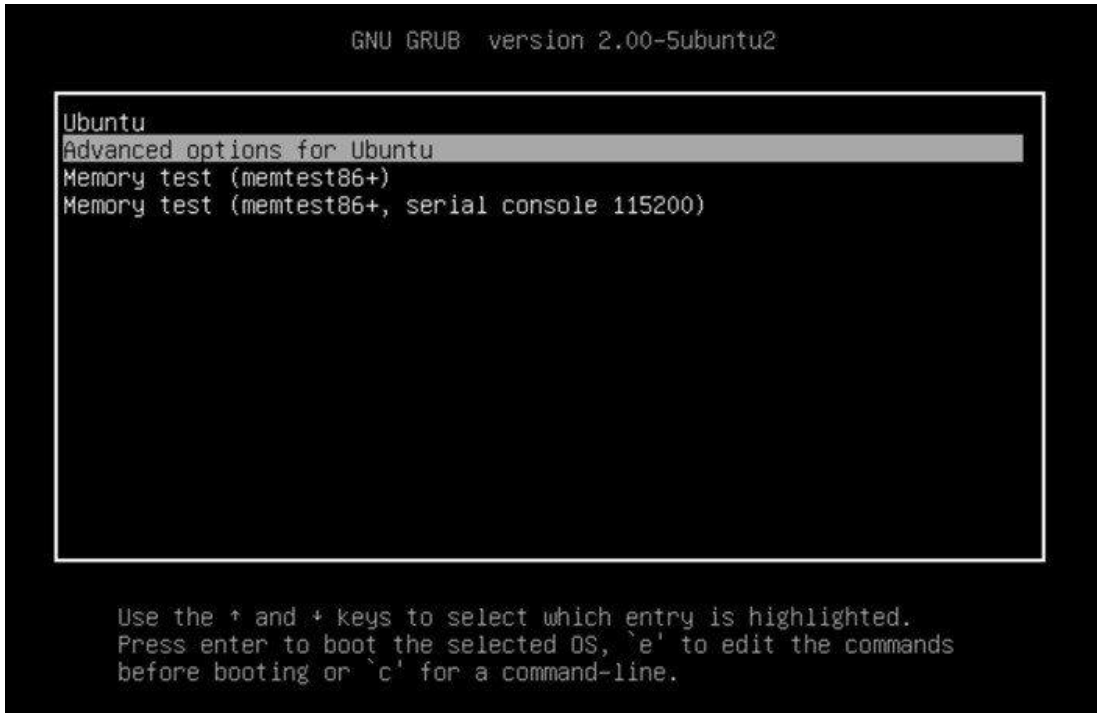
Locations of *boot.img*, *core.img* and the */boot/grub* directory

Example 1: An MBR-partitioned hard disk with sector size of 512 or 4096 bytes



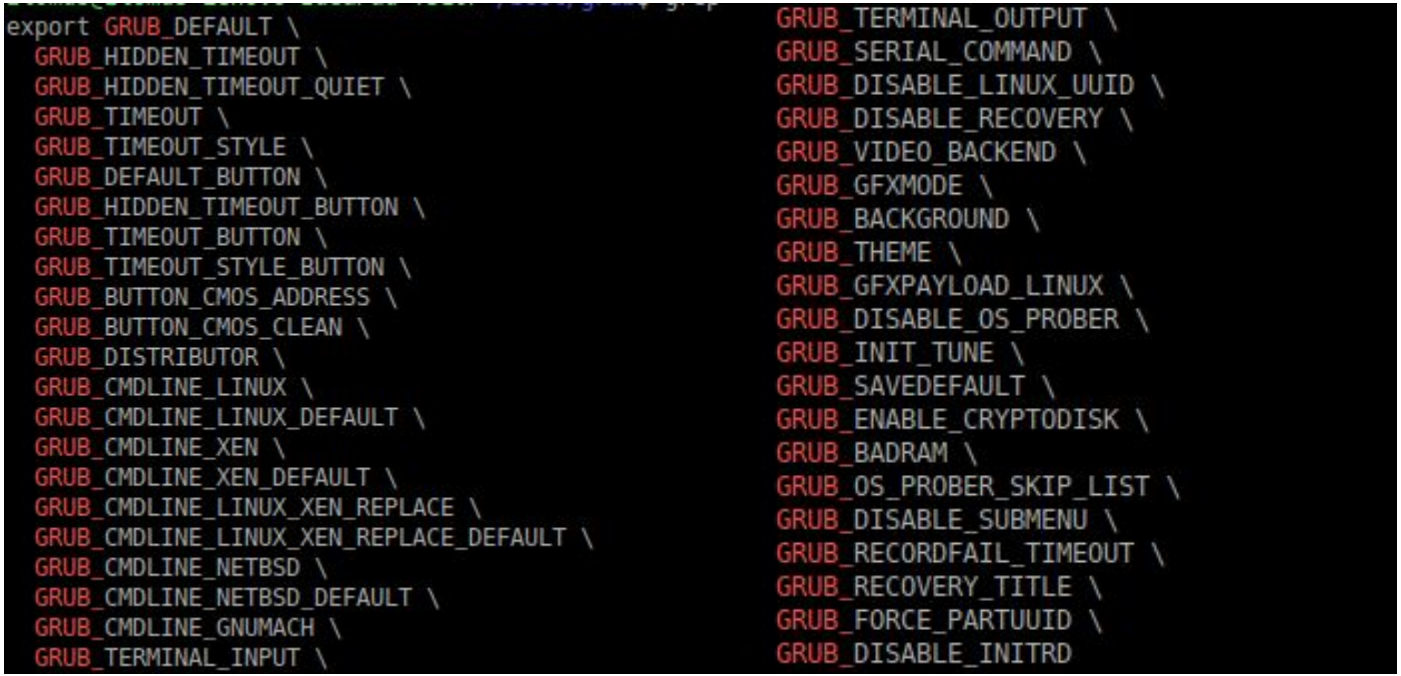
Example 2: A GPT-partitioned hard disk with sector size of 512 or 4096 bytes



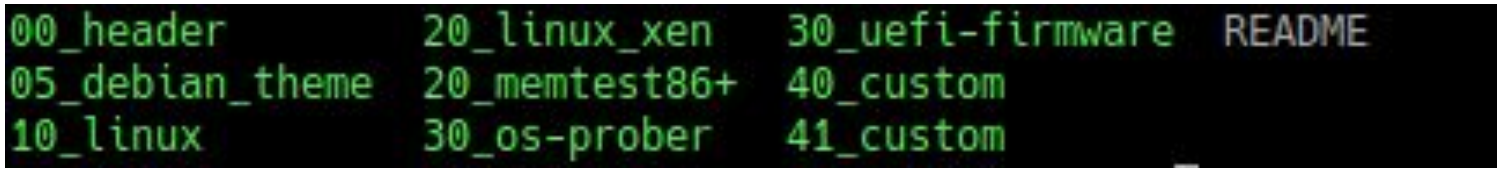

GNU GRUB version 2.00-5ubuntu2
Ubuntu
Advanced options for Ubuntu
Memory test (memtest86+)
Memory test (memtest86+, serial console 115200)
Use the ↑ and ↓ keys to select which entry is highlighted.
Press enter to boot the selected OS, `e' to edit the commands
before booting or `c' for a command-line.

Konfiguracja GRUB

```
#
# DO NOT EDIT THIS FILE
#
# It is automatically generated by grub-mkconfig using templates
# from /etc/grub.d and settings from /etc/default/grub
#
```


export GRUB_DEFAULT \
GRUB_HIDDEN_TIMEOUT \
GRUB_HIDDEN_TIMEOUT_QUIET \
GRUB_TIMEOUT \
GRUB_TIMEOUT_STYLE \
GRUB_DEFAULT_BUTTON \
GRUB_HIDDEN_TIMEOUT_BUTTON \
GRUB_TIMEOUT_BUTTON \
GRUB_TIMEOUT_STYLE_BUTTON \
GRUB_BUTTON_CMOS_ADDRESS \
GRUB_BUTTON_CMOS_CLEAN \
GRUB_DISTRIBUTOR \
GRUB_CMDLINE_LINUX \
GRUB_CMDLINE_LINUX_DEFAULT \
GRUB_CMDLINE_XEN \
GRUB_CMDLINE_XEN_DEFAULT \
GRUB_CMDLINE_LINUX_XEN_REPLACE \
GRUB_CMDLINE_LINUX_XEN_REPLACE_DEFAULT \
GRUB_CMDLINE_NETBSD \
GRUB_CMDLINE_NETBSD_DEFAULT \
GRUB_CMDLINE_GNUMACH \
GRUB_TERMINAL_INPUT \
GRUB_TERMINAL_OUTPUT \
GRUB_SERIAL_COMMAND \
GRUB_DISABLE_LINUX_UUID \
GRUB_DISABLE_RECOVERY \
GRUB_VIDEO_BACKEND \
GRUB_GFXMODE \
GRUB_BACKGROUND \
GRUB_THEME \
GRUB_GFXPAYLOAD_LINUX \
GRUB_DISABLE_OS_PROBER \
GRUB_INIT_TUNE \
GRUB_SAVEDEFAULT \
GRUB_ENABLE_CRYPTODISK \
GRUB_BADRAM \
GRUB_OS_PROBER_SKIP_LIST \
GRUB_DISABLE_SUBMENU \
GRUB_RECORDFAIL_TIMEOUT \
GRUB_RECOVERY_TITLE \
GRUB_FORCE_PARTUUID \
GRUB_DISABLE_INITRD

```
### END /etc/grub.d/10_linux ###

### BEGIN /etc/grub.d/20_linux_xen ###

### END /etc/grub.d/20_linux_xen ###

### BEGIN /etc/grub.d/20_memtest86+ ###
### END /etc/grub.d/20_memtest86+ ###

### BEGIN /etc/grub.d/30_os-prober ###
### END /etc/grub.d/30_os-prober ###

### BEGIN /etc/grub.d/30_uefi-firmware ###
menuentry 'System setup' $menuentry_id_option 'uefi-firmware' {
        fwsetup
}
### END /etc/grub.d/30_uefi-firmware ###

### BEGIN /etc/grub.d/40_custom ###
# This file provides an easy way to add custom menu entries.  Simply type the
# menu entries you want to add after this comment.  Be careful not to change
# the 'exec tail' line above.
### END /etc/grub.d/40_custom ###

### BEGIN /etc/grub.d/41_custom ###
if [ -f  ${config_directory}/custom.cfg ]; then
  source ${config_directory}/custom.cfg
elif [ -z "${config_directory}" -a -f  $prefix/custom.cfg ]; then
  source $prefix/custom.cfg;
fi
### END /etc/grub.d/41_custom ###
```


00_header
05_debian_theme
10_linux
20_linux_xen
20_memtest86+
30_os-prober
30_uefi-firmware
README
40_custom
41_custom

LILO


LILO 22.7 Boot Menu
Windows
Gentoo
Hit any key to cancel timeout 00:18
Use ←↑↓→ arrow keys to make selection
Enter choice & options, hit CR to boot

GRUB a LILO

Name	Advanced command	Scriptable	Supported architecture	Supported filesystem	Supported OS	Supported executable	Supported protocol	Supported decompression	Others
GRUB Legacy	Yes	No	x86 (PC)	FAT16, FAT32, MINIX fs, Linux ext2, ext3, ext4, ReiserFS, JFS, XFS, VSTa fs, Btrfs	FreeBSD, NetBSD, OpenBSD, Linux	ELF	TFTP	gzip	
GRUB 2	Yes	Yes	x86 (PC, UEFI, coreboot, OLPC, Mac), IA-64, ARM (U-Boot, UEFI), PowerPC (Mac, Pegasos II, IBM), MIPS, SPARC (SPARC v9), QEMU	ext2, ext3, ext4, btrfs, zfs, ufs, minix, iso9660, udf, jfs, hfs, hfs+, afs, affs, sfs, xfs, reiserfs, tar, cpio, NTFS, FAT16, FAT32	Linux (PC, mac), FreeBSD (PC), OpenBSD (PC), NetBSD (PC)	Multiboot and others	?	gzip, xz[4]	
LILO	No	No	x86 (PC)	indifferent[citation needed]	?	?	?	bzip2, gzip	

Note: The column **MBR** (Master Boot Record) refers to whether or not the boot loader can be stored in the first sector of a mass storage device. The column **VBR** (Volume Boot Record) refers to the ability of the boot loader to be stored in the first sector of any partition on a mass storage device.

Name	License	Can reside in				Can boot from								Can boot								
		ESP (UEFI)	MBR	VBR	Floppy	Hard disk	Second Hard disk	Logical partitions	CD-ROM	Floppy	USB	Zip	LAN	MS-DOS	Windows 9x/Me	Windows NT series	Windows Vista/7/8/10	Linux	ReactOS	MenuetOS	*BSD	Mac OS X
GNU GRUB	GPLv3	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Calls NTLDR	Calls Windows Boot Manager	Yes	Calls FreeLoader	Yes	Yes	Yes
LILO	BSD license	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	?	?	Yes	Calls NTLDR	Yes	Yes	?	?	Calls biosboot (FreeBSD, PC-BSD, ...)	?

SYSLINUX

- SYSLINUX
- ISOLINUX
- PXELINUX
- EXTLINUX
- MEMDISK
- dwa systemy menu
- środowisko programistyczne

Network booting