

# Процесс загрузки GNU/Linux

Мартынов Семён

Санкт-Петербургский политехнический университет Петра Великого

*semen.martynov@gmail.com*

1 апреля 2015 г.

- 1 Этапы загрузки
- 2 Простейший boot-loader
- 3 Вопросы

# Этапы загрузки

Процесс загрузки выполняет следующие шаги:

- [ Power On => ] Reset CPU (микрокод процессора)  
Передача управления на 0xffff0 [cs = 0xf000, ip = 0xffff0] - BIOS SC  
Power-on self-test (POST)

# Этапы загрузки

Процесс загрузки выполняет следующие шаги:

- [ Power On => ] Reset CPU (микрокод процессора)  
Передача управления на 0xffff0 [cs = 0xf000, ip = 0xffff0] - BIOS SC  
Power-on self-test (POST)
- Определение устройств с собственными BIOS и их инициализация  
Memory Test, настройка параметров устройств  
Выбор загрузочного устройства  
Передача управления загрузчику (Jump at 0x7c00)

# Этапы загрузки

Процесс загрузки выполняет следующие шаги:

- [ Power On => ] Reset CPU (микрокод процессора)  
Передача управления на 0xffff0 [cs = 0xf000, ip = 0xffff0] - BIOS SC  
Power-on self-test (POST)
- Определение устройств с собственными BIOS и их инициализация  
Memory Test, настройка параметров устройств  
Выбор загрузочного устройства  
Передача управления загрузчику (Jump at 0x7c00)
- boot-loader (512 байт) - загружает grub

# Этапы загрузки

Процесс загрузки выполняет следующие шаги:

- [ Power On => ] Reset CPU (микрокод процессора)  
Передача управления на 0xffff0 [cs = 0xf000, ip = 0xffff0] - BIOS SC  
Power-on self-test (POST)
- Определение устройств с собственными BIOS и их инициализация  
Memory Test, настройка параметров устройств  
Выбор загрузочного устройства  
Передача управления загрузчику (Jump at 0x7c00)
- boot-loader (512 байт) - загружает grub
- GRUB - загружает ядро и initramfs

# Этапы загрузки

Процесс загрузки выполняет следующие шаги:

- [ Power On => ] Reset CPU (микрокод процессора)  
Передача управления на 0xffff0 [cs = 0xf000, ip = 0xffff0] - BIOS SC  
Power-on self-test (POST)
- Определение устройств с собственными BIOS и их инициализация  
Memory Test, настройка параметров устройств  
Выбор загрузочного устройства  
Передача управления загрузчику (Jump at 0x7c00)
- boot-loader (512 байт) - загружает grub
- GRUB - загружает ядро и initramfs
- Ядро ищет и стартует оборудование

# Этапы загрузки

Процесс загрузки выполняет следующие шаги:

- [ Power On => ] Reset CPU (микрокод процессора)  
Передача управления на 0xffff0 [cs = 0xf000, ip = 0xffff0] - BIOS SC  
Power-on self-test (POST)
- Определение устройств с собственными BIOS и их инициализация  
Memory Test, настройка параметров устройств  
Выбор загрузочного устройства  
Передача управления загрузчику (Jump at 0x7c00)
- boot-loader (512 байт) - загружает grub
- GRUB - загружает ядро и initramfs
- Ядро ищет и стартует оборудование
- initramfs (/sbin/init) готовит всё для запуска ОС



# Этапы загрузки

Процесс загрузки выполняет следующие шаги:

- [ Power On => ] Reset CPU (микрокод процессора)  
Передача управления на 0xffff0 [cs = 0xf000, ip = 0xffff0] - BIOS SC  
Power-on self-test (POST)
- Определение устройств с собственными BIOS и их инициализация  
Memory Test, настройка параметров устройств  
Выбор загрузочного устройства  
Передача управления загрузчику (Jump at 0x7c00)
- boot-loader (512 байт) - загружает grub
- GRUB - загружает ядро и initramfs
- Ядро ищет и стартует оборудование
- initramfs (/sbin/init) готовит всё для запуска ОС
- Change Root на настоящую систему

# Этапы загрузки

Процесс загрузки выполняет следующие шаги:

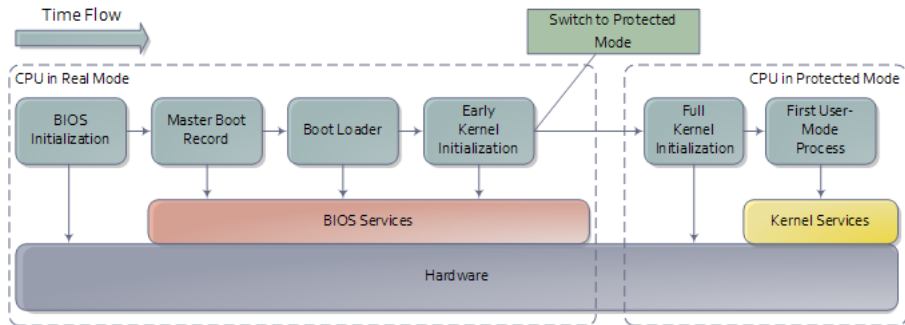
- [ Power On => ] Reset CPU (микрокод процессора)  
Передача управления на 0xffff0 [cs = 0xf000, ip = 0xffff0] - BIOS SC  
Power-on self-test (POST)
- Определение устройств с собственными BIOS и их инициализация  
Memory Test, настройка параметров устройств  
Выбор загрузочного устройства  
Передача управления загрузчику (Jump at 0x7c00)
- boot-loader (512 байт) - загружает grub
- GRUB - загружает ядро и initramfs
- Ядро ищет и стартует оборудование
- initramfs (/sbin/init) готовит всё для запуска ОС
- Change Root на настоящую систему
- /sbin/init настоящей системы запускает программы.

# Этапы загрузки

Процесс загрузки выполняет следующие шаги:

- [ Power On => ] Reset CPU (микрокод процессора)  
Передача управления на 0xffff0 [cs = 0xf000, ip = 0xffff0] - BIOS SC  
Power-on self-test (POST)
- Определение устройств с собственными BIOS и их инициализация  
Memory Test, настройка параметров устройств  
Выбор загрузочного устройства  
Передача управления загрузчику (Jump at 0x7c00)
- boot-loader (512 байт) - загружает grub
- GRUB - загружает ядро и initramfs
- Ядро ищет и стартует оборудование
- initramfs (/sbin/init) готовит всё для запуска ОС
- Change Root на настоящую систему
- /sbin/init настоящей системы запускает программы.
- Стартует getty или даже gdm

# Процесс загрузки



# Простейший boot-loader

```
2⊕ * mbr.c
7
8 /* XXX these must be at top */
9 #include "code16gcc.h"
10 __asm__ ("impl $0, $main\n");
11
12 #define __NOINLINE __attribute__((noinline))
13 #define __REGPARAM __attribute__((regparm(3)))
14 #define __NORETURN __attribute__((noreturn))
15
16 /* BIOS interrupts must be done with inline assembly */
17⊖ void __NOINLINE __REGPARAM print(const char *s) {
18     while (*s) {
19         __asm__ __volatile__ ("int $0x10" : : "a"(0x0E00 | *s), "b"(7));
20         s++;
21     }
22 }
23⊖ /* and for everything else you can use C! Be it traversing the filesystem,
24 * or verifying the kernel image etc.*/
25
26⊖ void __NORETURN main() {
27     print("Hello, World!\r\n:");
28     while (1)
29         ;
30 }
```

# Простейший boot-loader: проблемы

Код не будет работать!

# Простейший boot-loader: проблемы

Код не будет работать!

Проблемы:

- реальный режим работы процессора
- elf файл

# Простейший boot-loader: проблемы

Код не будет работать!

Проблемы:

- реальный режим работы процессора
- elf файл

Решения:

- `__asm__(".code16gcc\n");`
- специальный шаблон линкера



# Шаблон линкера

```
1 ENTRY(main);
2 SECTIONS
3 {
4     . = 0x7C00;
5     .text : AT(0x7C00)
6     {
7         _text = .;
8         *(.text);
9         _text_end = .;
10    }
11    .data :
12    {
13        _data = .;
14        *(.bss);
15        *(.bss*);
16        *(.data);
17        *(.rodata*);
18        *(COMMON)
19        _data_end = .;
20    }
21    .sig : AT(0x7DFE)
22    {
23        SHORT(0xaa55);
24    }
25    /DISCARD/ :
26    {
27        *(.note*);
28        *(.iplt*);
29        *(.igot*);
30        *(.rel*);
31        *(.comment);
32    }
33 }
```

```
$ gcc -c -g -Os -m32 -march=i686 -ffreestanding -Wall -Werror -I. -o  
mbr.o mbr.c  
$ ld -static -melf_i386 -Tlinker.ld -nostdlib -nmagic -o mbr.elf mbr.o  
$ objcopy -O binary mbr.elf mbr.bin  
  
$ dd if=/dev/zero of=floppy.img bs=1024 count=1440  
$ dd if=mbr.bin of=floppy.img bs=1 count=512 conv=notrunc  
  
$ qemu-system-i386 -fda floppy.img -boot a
```

```
QEMU
SeaBIOS (version 1.7.4-20140219_122725-roseapple)

iPXE (http://ipxe.org) 00:03.0 C900 PCI2.10 PnP PMM+07FC1110+07F21110 C900

Booting from Floppy...
Hello, World!
:)
```

# Вопросы?

Исходные коды:

[https://github.com/SemenMartynov/SPbPU\\_0SandComponents](https://github.com/SemenMartynov/SPbPU_0SandComponents)