

Операционные Системы

Процесс загрузки

Первая инструкция

- ▶ Откуда берется первая инструкция?
 - ▶ x86 обращается по адресу `0xFFFFFFF0`;
 - ▶ отвечает ему *материнская карта*.
- ▶ Какой код материнская карта отдает процессору?
 - ▶ BIOS (Basic Input/Output System) - наследство *IBM PC*;
 - ▶ UEFI (Unified Extensible Firmware Interface).

BIOS

- ▶ POST (Power-On Self-Test)
 - ▶ проверяет, что все на месте и "работает";
 - ▶ может выполнять начальную инициализацию устройств;
 - ▶ ищет загрузочное устройство (диск в ОС).

Загрузочное устройство

- ▶ BIOS ищет диск, с которого можно прочитать первые 512 байт
 - ▶ а. к. а. загрузочный сектор;
 - ▶ последние 2 байта сектора должны хранить числа *0x55* и *0xAA*;
 - ▶ сектор загружается в память по физическому адресу *0x7c00*.
- ▶ BIOS передает управление по физическому адресу *0x7c00*
 - ▶ мы добрались до места, где мы можем на что-то повлиять.

Окружение

- ▶ Что нам известно о состоянии системы?
 - ▶ наш код начинается по физическому адресу *0x7c00*;
 - ▶ устройства как-то инициализированы и прерывания отключены;
 - ▶ процессор работает в *Real Mode*.

Real Mode

- ▶ Логический адрес состоит из двух частей:
 - ▶ 16-битного сегмента (SEG) и 16-битного смещения (OFF);
 - ▶ физический адрес получается по формуле $(SEG * 16 + OFF) \bmod 2^{20}$.
- ▶ Сегмент хранится в одном из специальных регистров:
 - ▶ CS, DS, SS, ES, FS, GS .

Real Mode

- ▶ Регистры общего назначения 16-битные:
 - ▶ *SP* - указатель стека;
 - ▶ *BP* - указатель "базы";
 - ▶ *AX, BX, CX, DX, SI, DI*.

Hello, World!

```
1      .code16
2      .text
3      .global start
4  start:
5      ljmp 0x0, $real_start
6  real_start:
7      movw $0, %ax
8      movw %ax, %ds
9      movw %ax, %ss
10
11      movw $0x7c00, %sp
12      addw $0x0400, %sp
13      ...
14  loop:  jmp loop
```


Hello, World!

```
1      movw $0xB800, %ax
2      movw %ax, %es
3      movw $data, %si
4      movw $0, %di
5      movw size, %cx
6      call memcpy
7
8      ...
9
10 data:
11      .asciz "H\017e\017l\017l\017o\017!\017"
12 size:
13      .short . - data
```

Hello, World!

```
1 memcpy:
2         cmpw $0, %cx
3         jz  out
4 next:
5         movb (%si), %ah
6         movb %ah, %es:(%di)
7         incw %si
8         incw %di
9         decw %cx
10        jnz next
11 out:
12        ret
```

Начальный загрузчик

- ▶ Как много кода можно поместить в первые 510 байт?
 - ▶ вряд ли туда поместится целая современная ОС;
 - ▶ задача этого кода прочитать с диска код, не поместившийся в первые 510 байт.
- ▶ Оставшийся код может быть кодом ОС,
 - ▶ а может быть кодом (вторичного) загрузчика;
 - ▶ например, GRUB.