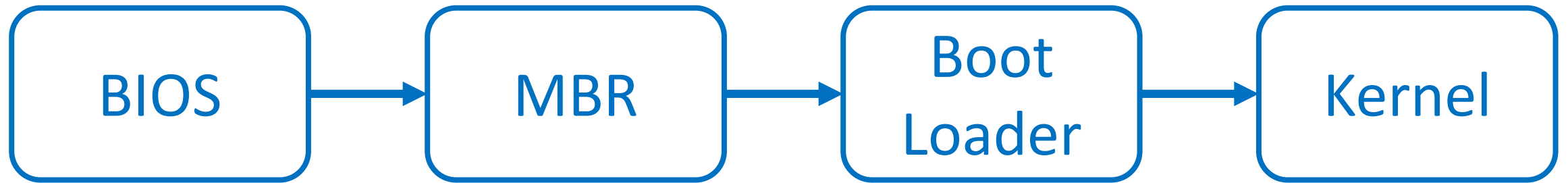


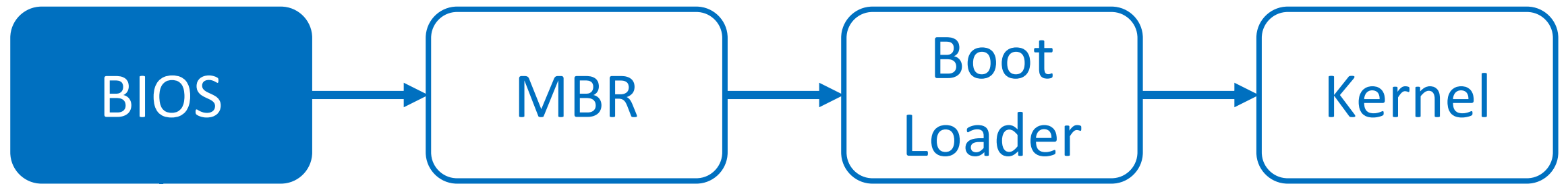
Operating Systems

Practice 2

OS Booting



OS Booting

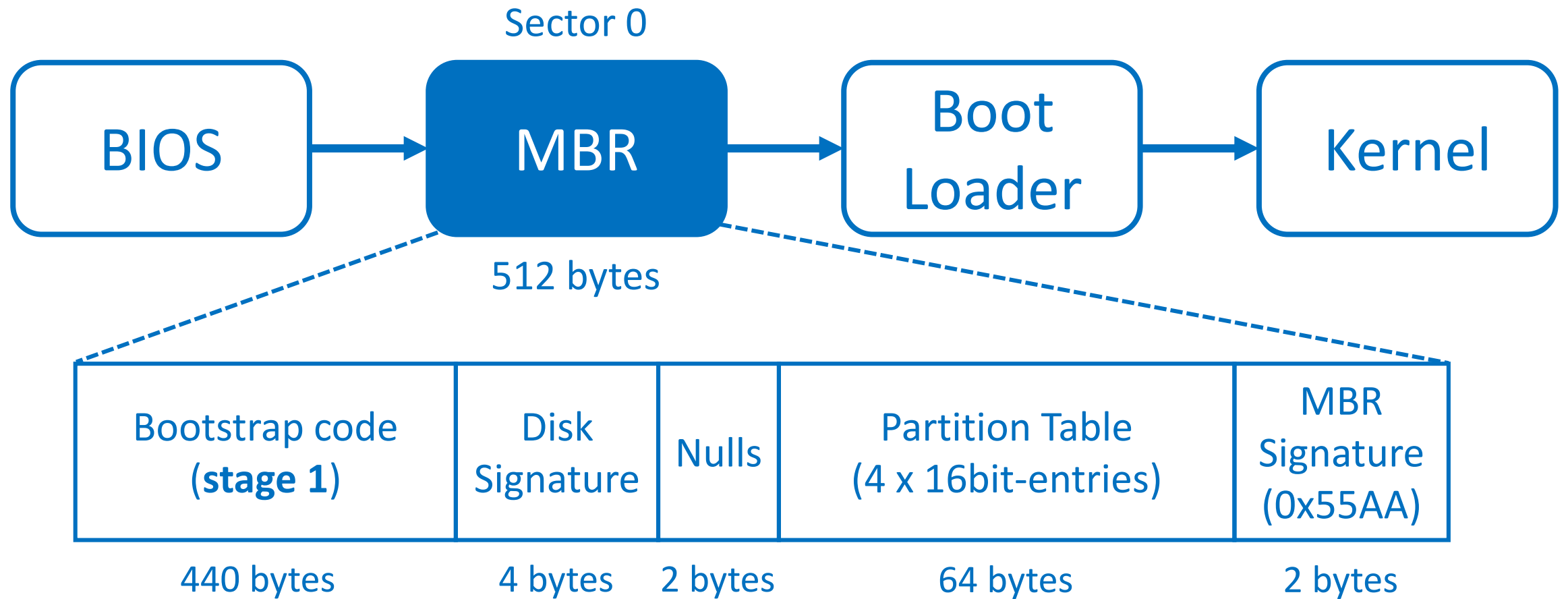


Power-On-Self Test
(POST)

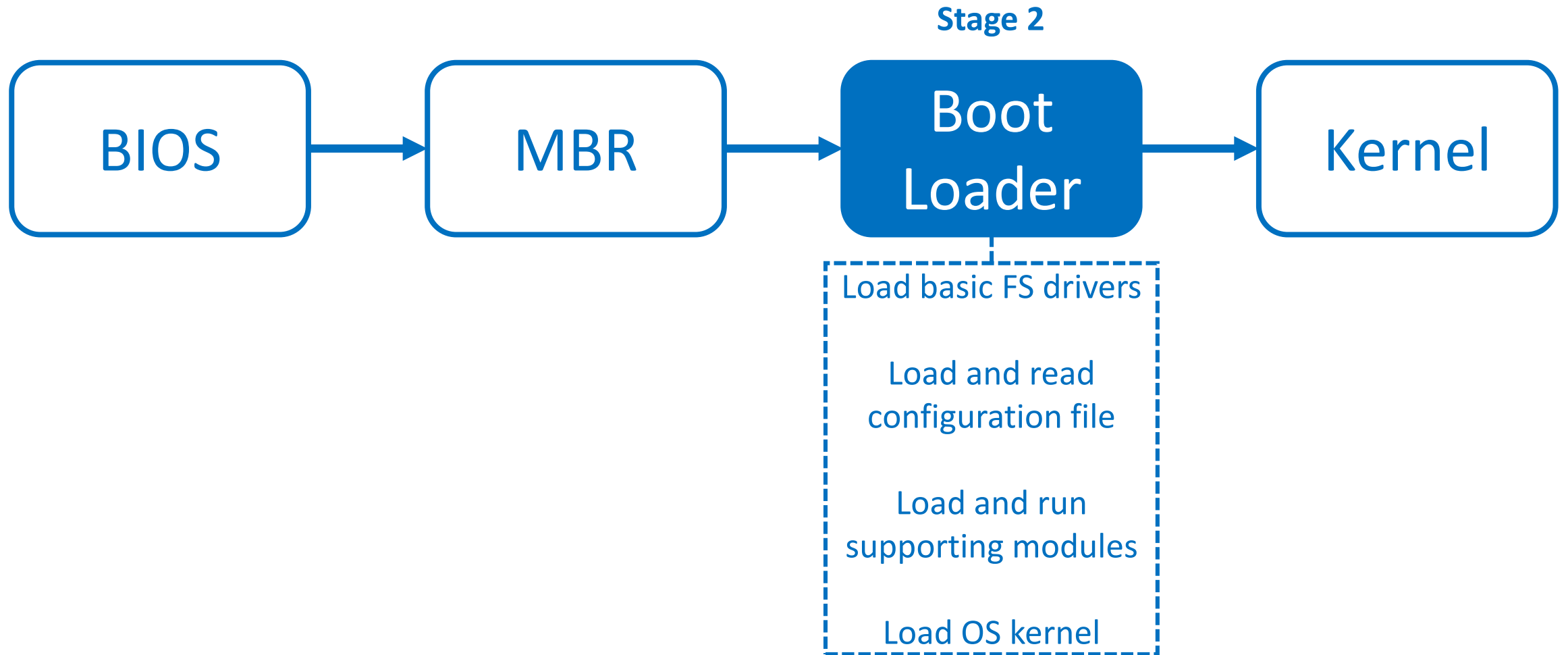
Поиск загрузочного
устройства

Загрузка MBR

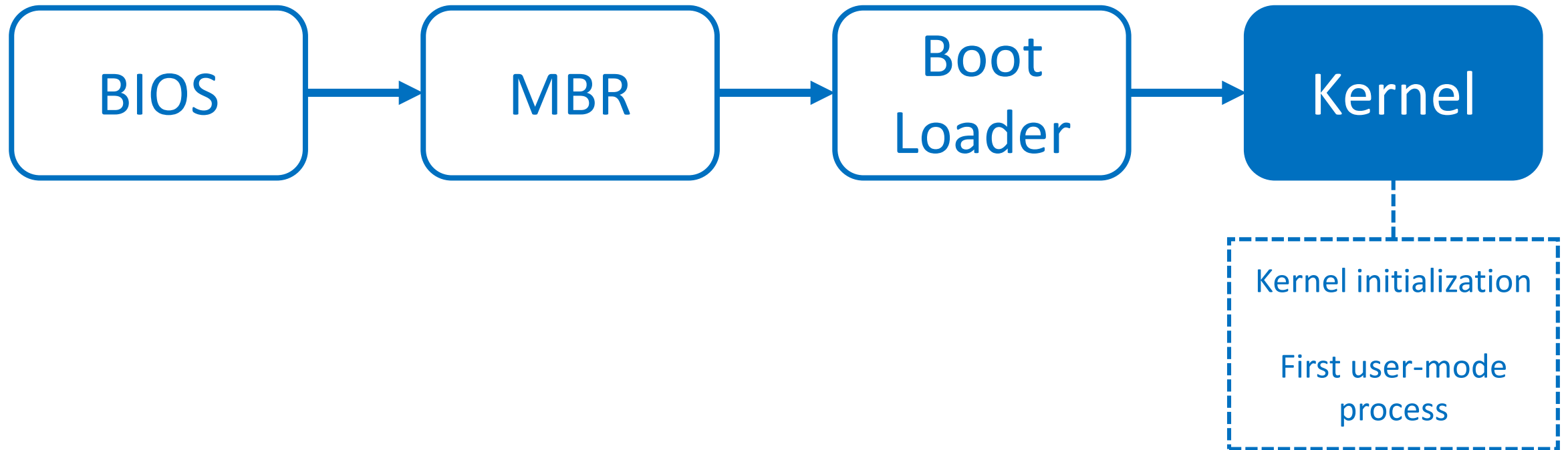
OS Booting



OS Booting



OS Booting



Multiboot Specification

<http://www.gnu.org/software/grub/manual/multiboot/multiboot.html>

x86 Assembly

General purpose registers

AX (accumulator register)
BX (base address register)
CX (count register)
DX (data register)

Stack registers

BP (base pointer)
SP (stack pointer)

Segment registers

CS (code segment)
SS (stack segment)
DS (data segment)
ES (extended segment)

https://en.wikibooks.org/wiki/X86_Assembly/X86_Architecture

https://en.wikipedia.org/wiki/X86_memory_segmentation

https://en.wikibooks.org/wiki/X86_Assembly/GAS_Syntax

Linker script

- Every link is controlled by a linker script. This script is written in the [linker command language](#).
- The [main purpose](#) of the linker script is to [describe how the sections in the input files should be mapped into the output file](#), and to control the memory layout of the output file. Most linker scripts do nothing more than this.
- You may supply your own linker script by using the `-T` command line option. When you do this, your linker script will replace the [default](#) linker script (use the `--verbose` command line option [to display](#) the [default](#) linker script).

http://www.math.utah.edu/docs/info/ld_3.html

http://www.delorie.com/gnu/docs/binutils/ld_6.html

Linker script: Example

```
OUTPUT_FORMAT(elf64-x86-64)
```

```
OUTPUT_ARCH(i386:x86-64)
```

```
ENTRY(mymain)
```

```
SECTIONS
```

```
{  
    . = 0x100000 + SIZEOF_HEADERS;  
    .text : { *(.text) }  
    .data : { *(.data) }  
}
```

Домашнее задание #1

Задание состоит из двух частей:

1. [MBR “Hello, World!” bootloader](#) (a. k. a. legacy bootloader)
2. [multiboot “Hello, World!” bootloader](#)

(см. подробности в гугл-доке на SEWiki)

Требования к выполнению (для группы №2)

Почта: k.novokreshchenov@gmail.com

Результатом выполнения задания должен быть архив с именем **<фамилия>_<группа>_os_hw1.zip**, который следует отправить на указанную почту с темой письма **OS_HW1**.

Внутри архива должны располагаться две папки с именами **1** и **2**, соответствующие первой и второй частям задания.

1. Внутри папки **1** должен быть расположен файл **Makefile**, в результате запуска которого (утилитой **make**) должен появиться **чисто бинарный** файл **hello**.

Тестирование работоспособности: `qemu-system-i386 -hda hello`

2. Внутри папки **2** должен быть расположен файл **Makefile**, в результате запуска которого (утилитой **make**) должен появиться исполняемый файл **hello**, удовлетворяющий **спецификации multiboot**.

Тестирование работоспособности: `qemu-system-i386 -kernel hello`