# Домашнее задание 1 (a. k. a. warm up)

Задание состоит из двух частей:

- MBR “Hello, World!” bootloader (a. k. a. legacy bootloader)
- multiboot “Hello, World!” bootloader

**MBR “Hello, World!” bootloader**

В первой части вам требуется написать bare metal программу (т. е. программу, которую можно запустить на голом железе без ОС), которая выводит на экран “Hello, World!” (мы придерживаемся традиции создавать первой программой “Hello, World!”).

Программа будет записана в MBR диска (как правило первые 512 байт диска). Чтобы BIOS воспринял сектор как загрузочный, он должен содержать в последних двух байтах специальную сигнатуру - число 0xAA55. Добиться этого вы можете используя linker script или явно в ассемблерном коде.

Если BIOS определит, что сектор является загрузочным (т. е. увидит сигнатуру в двух последних байтах сектора), то сектор будет загружен по адресу 0x7C00. Соответственно, код должен быть составлен с учетом того, что он будет загружен по указанному адресу.

Для вывода строки “Hello, World!” на экран вы можете воспользоваться прямой работой с видеопамятью (за подробностями смотрите приложение). А после вывода строки войдите в бесконечный цикл, так как bare metal программу нельзя завершить, с помощью какой-нибудь функции exit или return, т. е. программа продолжит исполнять дальше то, что находится в памяти - кто знает, что из этого может выйти.

Учтите, что эта программа должна быть “голым” бинарным файлом, т. е. никаких заголовков и прочего - только код, только хардкор.

**Multiboot “Hello, World!” bootloader**

Если вы когда-нибудь соберетесь создавать свою ОС, которая затмит Windows, Linux и Mac OS X, то не начинайте с создания своего собственного загрузчика, потому что есть универсальный загрузчик GRUB.

Соответственно, в этом задании вам так же нужно создать программу (уже не совсем bare metal), которая так же как и предыдущая печатает “Hello, World!” на экран и входит в бесконечный цикл. Но на этот раз эту программа будет загружена с помощью GRUB.

Для этого исполняемый файл должен соответствовать спецификации multiboot, которая доступна по ссылке:

https://www.gnu.org/software/grub/manual/multiboot/multiboot.html .

При написании linker-script-а сделайте так, чтобы бинарный файл загружался начиная с адреса 1M, а кроме того учтите, что бинарный файл должен быть в формате ELF (32 бита - особенно следите за этим владельцы 64 ОС-ей), а значит нужно добавить SIZEOF_HEADERS.

**Работа с видеопамятью напрямую**

Для вывода текста на экран, вам предлагается использовать прямой доступ к VGA буферу (но вы можете использовать и другой вариант). Обычно VGA буффер располагается в диапазоне памяти от 0xA0000 до 0xBFFFF. Разным режимам работы соответствуют разные участки буффера.

Обычно, при старте BIOS устанавливает VGA контроллер в режим 7 (звучит круто). Это древний текстовый режим, в котором вам доступно 80 колонок и 25 строк символов (по крайней мере мы можем достаточно надежно полагаться на то, что их будет не меньше).

Память соответствующая 7 режиму начинается по адресу 0xB8000. Каждому символу на экране соответствуют два последовательных байта в памяти. Т. е. первому символу в первой строке соответствуют байты по адресам 0xB8000 и 0xB8001, а второму символу первой строки байты по адресам 0xB8002 и 0xB8003. Первый из двух байт соответствует ASCII коду символа, а второй атрибутам (цвет, фон). Подробнее про атрибуты вы можете почитать здесь:

https://en.wikipedia.org/wiki/VGA-compatible_text_mode

**Способ проверки**

Проверить оба задания можно используя виртуальную машину QEMU. Проверить MBR “Hello, World!” bootloader можно с помощью следующей команды:

qemu-system-i386 -hda hello

где qemu-system-i386 - исполняемый файл QEMU (на вашей системе имя может отличаться), а hello - исполняемый файл bare metal программы. Так как предполагается, что она будет записана в первый сектор диска, то этот исполняемый файл можно просто использовать как образ диска, поэтому и используется опция hda.

Чтобы проверить multiboot “Hello, World!” bootloader можно использовать опцию -kernel QEMU:

qemu-system-i386 -kernel hello

где hello - ваша multiboot программа. Опция -kernel может загружать в QEMU исполняемый файлы соответствующие спецификации multiboot.

Если вы хотите проверить ваши решения на реальном железе, то будьте очень осторожны - не запишите MBR “Hello, World!” bootloader в MBR вашего основного диска (это для вас плохо закончится), а используйте флешку (и все равно будьте внимательны, потому что, например, в linux легко опечаться и написать /dev/sda вместо /dev/sdb).

**Требования к выполнению задания**

- мы не ограничиваем вас в используемых материалах (например, вы можете свободно google-ить), но за преподавателем практики остается право задавать вопросы по вашему решению и вы должны быть готовы на них ответить;
- средства сборки и способ сдачи обговариваются с преподавателем практики, если преподаватель не выдал никаких указаний на этот счет, то стоит ему об этом напомнить;
- задания принимаются до практического занятия через 2 недели, задания присланные после не рассматриваются;
- у преподавателя могут быть замечания по вашему решению, и он может попросить вас их исправить - учтите это при выполнении задания; сроки выполнения задания рассчитаны на то, что вы не будете откладывать его до последнего момента.