### **Задание 1**

1. Какое максимальное количество загрузчиков GRUB2 может быть в системе?
2. Сколько при этом может быть установлено различных ОС?

Чем ограничиваются эти возможности?

**Ответ:**

По 1 - что значит в “системе” в системе - ОС (т.е. для запуска одной ОС?) или в “системе” - в смысле компьютера? Считаю в системе в смысле - “компьютера”.

Как я понял, из лекции, на каждом отдельном физ. диске может быть свой раздел с загрузчиком GRUB2. Поэтому получается число загрузчиков зависит от количества дисков. Если в системе (компьютере) будет много физических дисков (или виртуальных) то, на каждом может быть свой раздел загрузчика GRUB. На каждом может быть своя ОС. Какую ОС загружать - описывается (как я понял) в менеджере загрузки EFI.

Но, правда, мне не понятно, почему не может быть на одном диске много разных загрузчиков? Ведь на диске может быть много разделов. При этом UEFI может работать через драйверы с разделами разными. Получается в теории, что можно бы на одном диске сделать много разных разделов (в теории) т.е. несколько разделов с загрузчиком. И несколько разделов с образами ядер - ОС. Но, видимо ограничение прописывается в самом стандарте UEFI. Видимо загрузчик может быть только один на диск, и загрузчик должен быть в первом разделе диска.

Еще предполагаю:

Если будет один загрузчик на одном диске, но, на диске будет много разделов, то, можно на одном диске сделать несколько разных образов ОС (ядра). Возможно даже можно эти образы на одном разделе разместить. Тогда если вручную зайти в GRUB при загрузке, то, можно видимо выбрать любой образ ОС. Т.е. можно запускать с одного диска - много разных ОС, но, в ручном режиме. Еще предполагаю, что можно в GRUB сделать меню - где спрашивается какой образ запускать, или выбирается один по дефолту и ждется таймаут, например 15 сек.

*Приведите ответ в свободной форме со своим комментарием.*

### **Задание 2**

На лекции мы рассмотрели загрузку ядра и дополнительных модулей. Ответьте на вопрос: почему сразу не сделано ядро со всеми модулями? В таком случае мы смогли бы избавиться от initrd.

**Ответ:**

1. Во первых было бы монолитное ядро большое (предполагаю) что неудобно.
2. Во вторых - речь тут идет не о модулях ядра, а юнитах загрузки, а это как я понял означает, что не только в модуль ядра что-то добавляется (а может и вообще не добавляется), а еще и запускается отдельные процессы в ОС - службы. Значит нужно сначало получить работающее ядро, а потом на нем запускать отдельные процессы. В ядре будет планировщик, который и будет на процессоре выполнять по очереди разные процессы. И у каждого процесса будет свой id. Поэтому не представляю как можно все модули сразу сдедлать в составе ядра? Все равно же они должны после ядра запуститься по отдельности - как отдельный процесс? Плюс как бы было логирование если бы весь функционал был бы в составе ядра? В общем я не знаю как можно в ораз ядра внедрить еще и отдельные сервисы сразу. Я так понимаю есть образ ядра - это исполняемый код машинный. Этот код загружается в память, и потом передается ему управление. И только потом этот код-ПО, в котором есть планировщик, должна быть в любом случае какая-то система инициализация, которая по плану будет запускать остальные ПО, сервисы. Поэтому не представляю как в образ ядра можно сразу включить много сервисов - отдельных процессов.

*Приведите ответ в свободной форме со своим комментарием.*

### **Задание 3\***

В чём отличия процесса загрузки BIOS и UEFI?

BIOS - это один большой кусок исполняемого кода, который хранится в памяти энерго-независимой (т.е. видимо, или подпитываемая батарейкой или флеш память). Этот кусок кода прям сразу попадает на процессор и выполняется. Т.е. в этом коде и и есть сразу - вся программа BIOS монолитная. И при этом по стандартам этот объем не более 1 Мб. может быть (как я понял). И далее это монолитное ПО загруженное на CPU уже ищет MBR область на диске, где будет загрузчик ОС - тоже в исполняемом коде сразу на диске, без какой либо файловой системы. Т.е. на диске будет раздел, и в этом разделе будут сразу байты исполняемого кода машинного, но, не будет разметки диска - файловой системы, не будет файлов, т.к. файлы - это уже некоторая виртуальная абстракция данных - байтов на диске в сочетании с разметкой диска - файловой системы. Исполняемый код загрузчика из MBR попадает в память и выполняется на процессоре, и уже этот загрузчик загружает драйвера, для работы с файловыми системами. И потом уже загружает из соседнего диска файлы - образа ОС. Так работает загрузка BIOS.

Загрузк через UEFI:

В прошивке материнской платы нет уже монолитного ПО - набора исполняемого кода. Есть отдельный стартовый раздел в прошивке платы, где располагается машинный код для старта начального загрузчика UEFI (не знаю как правильно назвать) или, думаю, можно это назвать микро-ядром UEFI. При этом в этом же микро ядре есть уже драйвера, для работы с файловыми системами - разметкой на дисках. Далее есть менеджер загрузки UEFI который согласно настройкам может уже подгружать с из других разделов прошивки уже файлы (т.е. разделы памяти уже с файловой разметкой - файловой системой). При том таких модулей файлов можно загрузить разных много и каждый файл или модуль - это отдельное независимое ПО. Т.е. микро-ядро UEFI это по сути - маленькая ОС, в которой есть драйвера для работы с дисками с разными файловыми системами и свой планировщик, который может обеспечивать работу разных программ независимых одновременно. Далее после загрузки всех модулей UEFI (и проверки работоспособности аппаратной части) менеджер загрузки ищет уже на дисках загрузчик GRUB для загрузки образа ОС. При этом файлы GRUB уже располагаются в файловой системе в виде файлов, а не просто машинного кода на разделе диска. Согласно настройкам менеджер загрузки UEFI выбирает нужный диск, и загружает в память загрузчик GRUB и передает ему управление. Далее уже загрузчик загружает в память кода ядра из образа - отдельный файл на соcеднем разделе диска. Загрузчик GRUB передает управление загруженному коду ядра (код выполняется), далее ПО ядра уже начинает загружать разные юниты - службы системные и пользовательские. При этом загрузка ядра идет так:GRUB монтирует в виде файловой системы в RAM образ ядра. А потом этот образ размонтируется и монтируется уже корневая VFS.

*Приведите ответ в свободной форме со своим комментарием.*