



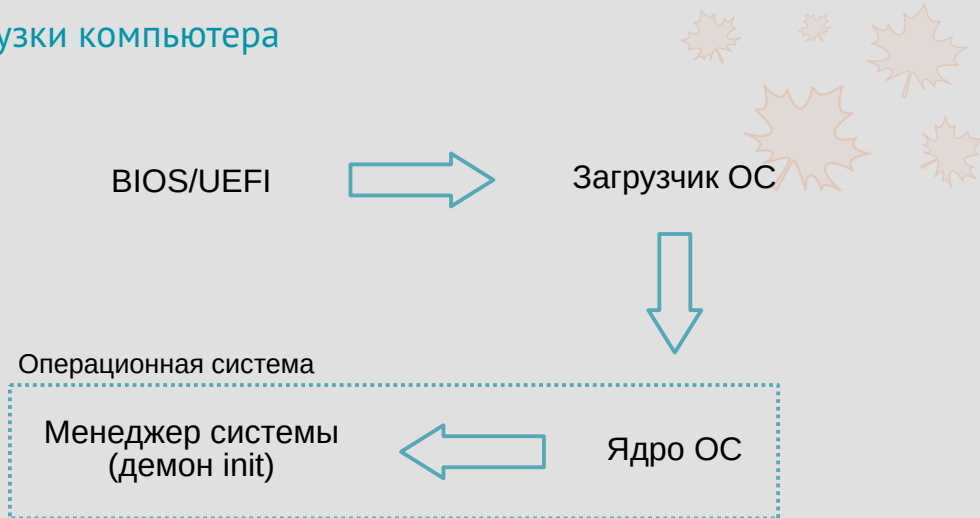
Факультет
биоинженерии и биоинформатики

Московский государственный университет имени М.В.Ломоносова

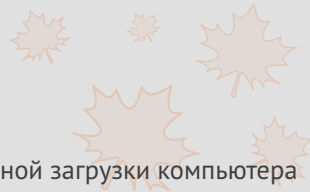


Загрузка компьютера под управлением GNU/Linux

Этапы загрузки компьютера



BIOS/UEFI



BIOS – Basic Input/Output System – устаревший механизм начальной загрузки компьютера

UEFI – Unified Extensible Firmware Interface – современный стандарт

Какие функции выполняет:

- ▶ инициализация и проверка работоспособности оборудования (POST – Power-On Self-Test)
- ▶ выбор/поиск и исполнение кода загрузчика операционной системы
- ▶ предоставление низкоуровневого интерфейса к оборудованию для ПО
- ▶ настройка оборудования пользователем

Загрузка с помощью BIOS



- ▶ Данные на загрузочном устройстве должны начинаться с MBR (Master Boot Record) – как правило, это один сектор 512 байт.
- ▶ MBR содержит таблицу разделов диска и код первичного загрузчика.
- ▶ Единственная задача первичного загрузчика – поиск вторичного загрузчика и передача ему управления.
- ▶ Вторичный загрузчик может находиться в начале одного из разделов диска, или в неразмеченном пространстве до первого раздела.
- ▶ Вторичный загрузчик уже умеет понимать файловые системы, он загружает саму операционную систему.

Пример загрузки BIOS

● Phoenix - AwardBIOS v6.00PG, An Energy Star Ally
✈ Copyright (C) 1984-2005, Phoenix Technologies, LTD

ASUS A8N-SLI Premium ACPI BIOS Revision 1011-001

Main Processor: AMD Athlon(tm) 64 Processor 4000+
Memory Testing : 2097152K OK(Installed Memory: 2097152K)
Memory information: DDR 400 Dual Channel, 128-bit

Chipset Model: nForce 4

Primary IDE Master : PLEXTOR DVD-R PX-716AL 1.02

Primary IDE Slave : None

Secondary IDE Master : CD-W524E 1.0E

Secondary IDE Slave : None



Press **F1** to continue, **DEL** to enter SETUP
12/07/2005-NF-CK804-A8NSLI-P-00

Загрузка с помощью UEFI



- ▶ UEFI работает с таблицей разделов в формате GPT (GUID Partition Table, GUID – Globally Unique Identifier).
- ▶ UEFI умеет понимать файловую систему FAT (File Allocation Table).
- ▶ Для загрузки с устройства, на нем должен быть один специальный раздел с файловой системой FAT, называемый EPS (EFI System Partition).
- ▶ EPS содержит необходимые для дальнейшей загрузки файлы, в том числе код загрузчика ОС.

Пример загрузки UEFI



BIOS vs. UEFI



BIOS+MBR не следует считать "вымершим" стандартом, он встречается не только на старых компьютерах, но и

- ▶ в виртуальных машинах,
- ▶ на некоторых новых компьютерах (UEFI умеет работать в режиме совместимости).

Таблица разделов MBR имеет значительные ограничения на количество и размер разделов диска. При первичной установке ОС на новых компьютерах следует использовать UEFI+GPT+ESP.

Загрузчик ОС



GRUB (GRand Unified Bootloader) – наиболее популярный загрузчик Linux.

- ▶ Графическое меню позволяет пользователю выбрать ядро операционной системы и режим загрузки.
- ▶ Можно редактировать пункты меню перед загрузкой (клавиша `e`), например, поменять параметры ядра или исправить путь к файлу-образу.
- ▶ Есть режим консоли (клавиша `c`), можно ввести все необходимые для загрузки ОС команды вручную.

GRUB загружает выбранное ядро в оперативную память и передает ему управление.

Пример загрузки GRUB



Настройка GRUB

- ▶ Содержание загрузочного меню определяется текстовым файлом `/boot/grub/grub.cfg`.
- ▶ Файл генерируется автоматически программой `update-grub` на основе шаблонов из папки `/etc/grub.d`.
- ▶ Шаблоны являются shell-сценариями, выдача которых формирует файл.
- ▶ Стандартные сценарии содержат код для
 - ▶ определения установленных ядер Linux
 - ▶ формирования пунктов загрузки в режиме отладки
 - ▶ поиска операционных систем на других разделах
 - ▶ ...
- ▶ Часто хватает изменения основных параметров загрузки в файле `/etc/default/grub`, который считывается стандартными шаблонами.

Фрагмент файла grub.cfg

```
#
# DO NOT EDIT THIS FILE
#
...
menuentry 'Debian GNU/Linux' --class debian --class os $menuentry_id_option 'id1b02xxxx' {
    load_video
    insmod gzio
    if [ x$grub_platform = xxen ]; then insmod xzio; insmod lzopio; fi
    insmod part_msdos
    insmod ext2
    set root='hd0,msdos1'
    search --no-floppy --fs-uuid --set=root 1b02f434-xxxxx
    echo 'Loading Linux 5.10.0-26-amd64 ...'
    linux /boot/vmlinuz-5.10.0-26-amd64 root=UUID=1b02f434-xxxxx ro quiet
    echo 'Loading initial ramdisk ...'
    initrd /boot/initrd.img-5.10.0-26-amd64
}
submenu 'Advanced options for Debian GNU/Linux' {
    ...
}
```

Загрузка ядра ОС



GRUB загружает в оперативную память образ ядра Linux (файл `vmlinuz`) и файл `initrd.img` на основании конфигурации `grub.cfg`.

- ▶ Образ ядра поставляется в составе системных пакетов `linux-image-*`.
- ▶ Файл `initrd.img` является специфичным для конкретной системы, он генерируется программой `update-initramfs` из пакета `initramfs-tools`.

Загрузка ядра заканчивается монтированием корневой файловой системы и запуском системного демона `init`.

Временная корневая файловая система initramfs



- ▶ Для монтирования корневой файловой системы нужны драйверы.
- ▶ Включение драйверов для всех возможных устройств, на которых может располагаться корневая ФС, сделало бы ядро слишком большим.
- ▶ Решение – временная (расположена в оперативной памяти) корневая ФС, содержащая всё необходимое для монтирования реальной корневой ФС.
- ▶ Временная корневая ФС заполняется из файла `initrd.img`, в современных системах это просто архив в формате `cpio`.
- ▶ Ядро создает временную ФС (`tmpfs`), заполняет её файлами из архива и запускает программу `/init`, которая монтирует корневую ФС и запускает системный демон `init`.

Инициализация пользовательского пространства



После загрузки ядра ОС требуется запустить системные службы (демоны) и настроить устройства: смонтировать файловые системы, настроить сеть и прочее.

В зависимости от режима работы могут требоваться различные службы и устройства.

Задача системы инициализации – выполнить требуемые загрузочные сценарии (по запуску демонов и подготовке ресурсов) в правильном порядке, учитывая их зависимости.

Менеджер systemd



В большинстве современных дистрибутивов Linux система инициализации представлена менеджером systemd.

- ▶ Все объекты в systemd – юниты разных типов.
- ▶ Юниты описываются простыми текстовыми юнит-файлами.
- ▶ Все юниты связаны зависимостями разных типов, разрешая которые система инициализации определяет порядок активации юнитов.
- ▶ Ручное управление юнитами и менеджером осуществляется с помощью программы systemctl.

Юниты systemd



Основные типы юнитов systemd:

- service** – запуск и управление демонами
- target** – соответствуют состояниям системы, нужны для упрощения сети зависимостей
- mount** – монтирование файловых систем
- device** – управление устройствами
- slice** – распределение системных ресурсов
- timer** – активация юнитов по расписанию
- ...

Пример юнит-файла



```
# /lib/systemd/system/networking.service
```

[Unit]

```
Description=Raise network interfaces  
DefaultDependencies=no  
Requires=ifupdown-pre.service  
Wants=network.target  
After=local-fs.target network-pre.target apparmor.service ...  
Before=network.target shutdown.target network-online.target  
Conflicts=shutdown.target
```

[Install]

```
WantedBy=multi-user.target  
WantedBy=network-online.target
```

[Service]

```
Type=oneshot  
EnvironmentFile=-/etc/default/networking  
ExecStart=/sbin/ifup -a --read-environment  
ExecStop=/sbin/ifdown -a --read-environment --exclude=lo  
RemainAfterExit=true  
TimeoutStartSec=5min
```

Системный демон init



Демон `init` – первый системный демон (всегда имеет `pid 1`), с которого начинается порождение всех остальных процессов в пользовательском пространстве.

Процесс `init (/sbin/init)` запускается ядром ОС после монтирования корневой ФС.

Кроме инициализации системы, `init` выполняет функцию родителя для всех "осиротевших" процессов (получает их код возврата).

Многопользовательский режим

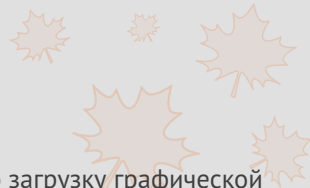


Состояние системы, в котором она готова для входа обычных пользователей.

В systemd этому режиму соответствует специальный юнит `multi-user.target`.

Все юниты, требуемые для входа и работы пользователей, содержат в юнит-файле строчку `WantedBy=multi-user.target` в секции `Install`.

Системные консоли



Переход в многопользовательский режим не предполагает обязательную загрузку графической подсистемы.

Работу с системой пользователи могут осуществлять с помощью системных консолей.

В каждой системной консоли запущена программа `login`, которая ожидает ввода логина и пароля пользователя и запускает командную оболочку от имени пользователя в случае успешной аутентификации.

Вид системной консоли



```
Debian GNU/Linux 11 laptop tty1

laptop login: user
Password:
Linux laptop 4.19.0-18-amd64 #1 SMP Debian 4.19.208-1 (2021-09-29) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Mon Nov 20 15:19:22 MSK 2023 on tty1
You have no mail.
user@laptop:~$
```

Пользовательские сессии systemd



Контроль за сессиями пользователей осуществляется системным демоном `systemd-logind`.

- ▶ Каждому залогиненному пользователю соответствует юнит типа `slice` для гибкого распределения системных ресурсов.
- ▶ Каждой пользовательской сессии – юнит типа `scope` для упрощения контроля и управления.

Взаимодействовать с менеджером сессий можно с помощью программы `loginctl`.

Графический режим



Для полноценной работы пользователей в графическом окружении, система должна перейти в графический режим.

В systemd этому режиму соответствует специальный юнит `graphical.target`.

Для перехода в графический режим требуется запуск `display-manager.service`.

Менеджер графических сессий



Задачи display-manager:

- ▶ запуск X-сервера,
- ▶ графический аналог программы login,
- ▶ контроль графических пользовательских сессий.

Примеры дисплейных менеджеров: LightDM, GDM, SDDM, ...

X Window System

