



# Как осуществляется загрузка системы

Power On Self-Test (POST)

Master Boot Record (MBR)

Boot sector (/dev/hdaX или /dev/sdaX)

Загрузчик ОС (grub, lilo, ntloader...)

Ядро

Процесс init (PID=1)

Система инициализации (BSD, System V, Upstart)

Стартовые скрипты



# Что происходит “за кулисами”

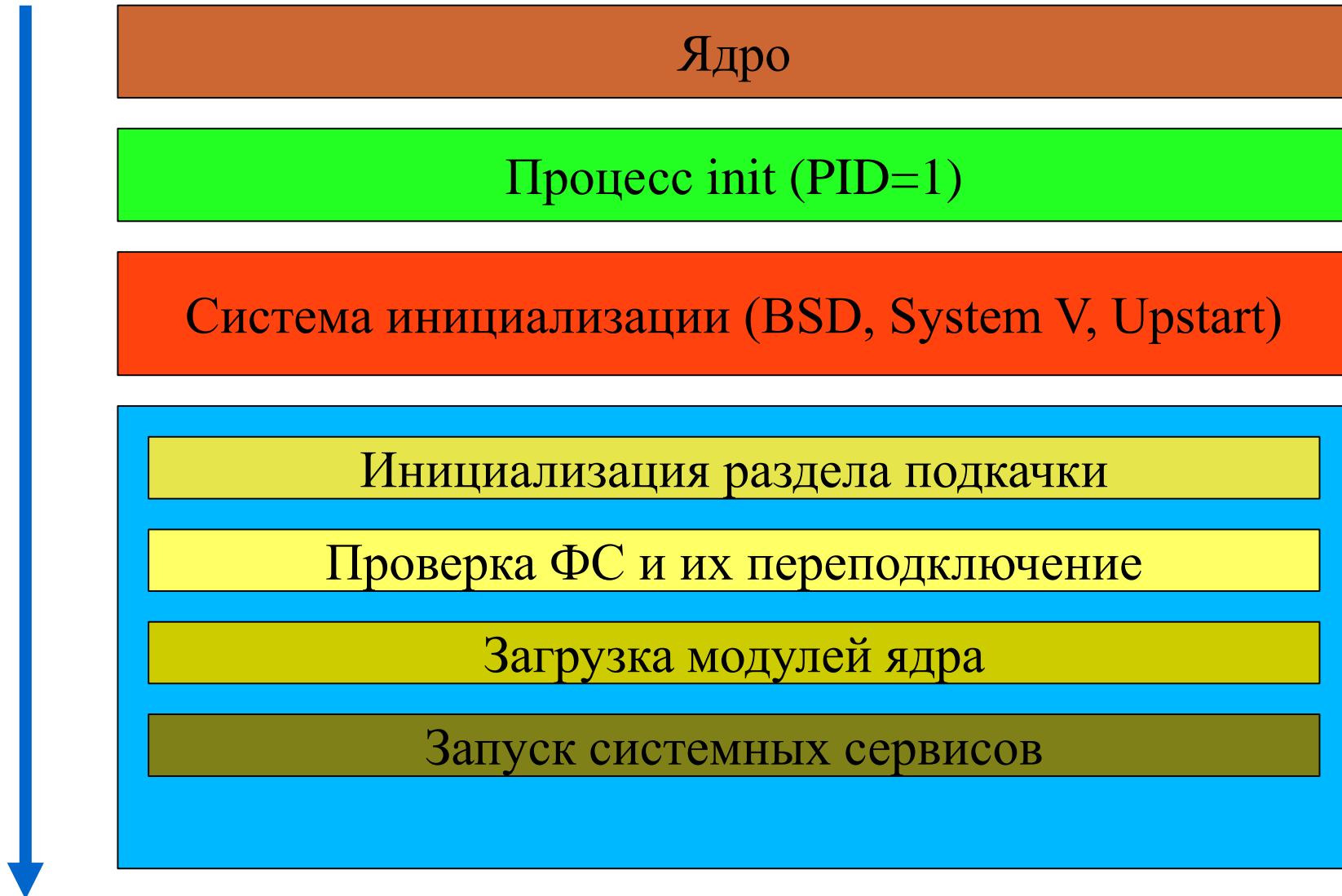
---

Ядро пытается загрузить процесс `init`, для этого:

- Проверяет не указана ли опция `init`, и если указана запускает соответствующую программу
- Если такой опции передано не было, то ядро пытается запустить `/init`
- В случае неудачи с `/init` ядро пытается запустить `/sbin/init`
- При отсутствии `/sbin/init` делается попытка запустить `/etc/init`
- При отсутствии `/etc/init` делается попытка запустить `/bin/init`
- Если `/bin/init` тоже отсутствует ядро пытается запустить `/bin/sh`
- В случае очередной неудачи – выдается “kernel panic”



# Порядок загрузки системы





# Классические системы инициализации

---

## Система инициализации BSD

- Процесс `init` использует `/etc/inittab` в котором прописано соответствие между уровнями выполнения и запускаемыми скриптами и уровень выполнения по умолчанию.
- Все скрипты располагаются в директории `/etc/rc.d`, а для того, чтобы программа запускалась при старте необходимо сделать исполняемым ее скрипт

## Система инициализации System V

- Процесс `init` использует `/etc/inittab` в котором прописан скрипт, который необходимо запустить при старте, соответствие между уровнями выполнения и запускаемыми скриптами и уровень выполнения по умолчанию
- Все скрипты располагаются в директории `/etc/init.d`, а функции используемые в стартовых скриптах в `/etc/rc.d/functions`
- Скрипты начинаются на "K" (останов сервисов) и на "S" (старт сервисов), при загрузке системы запускаются "S"-скрипты в соответствии с их нумерацией: `S01xxx,S02yyy...S11zzz..`



# Система инициализации Upstart

---

Событийно ориентированная конфигурация

Два типа сервисов: служба и задание

Основные возможности

- Задачи и службы запускаются и останавливаются при помощи событий
- При запуске/останове задач и служб генерируются события
- Событие может быть получено от любого процесса в системе
- Сервисы могут автоматически перезапускаться в случае их неожиданного останова
- Двухнаправленная связь с демоном `init`, что позволяет получать больше информации в процессе работы

Файлы конфигурации служб расположены в каталоге `/etc/init`

Все скрипты располагаются в директории `/etc/init.d` а функции используемые в стартовых скриптах в `/lib/lsb/init-functions`

Символические ссылки на скрипты размещаются в `/etc/rcx`, где `x` – соответствующий уровень выполнения

---



# Уровни выполнения

---

- 0 – завершение работы системы
- 1 или S – однопользовательский режим
- 2 – многопользовательский режим
- 3 – многопользовательский режим
- 4 – не используется
- 5 – многопользовательский режим + X Window
- 6 – перезагрузка системы



# Настройка оборудования

---

**lsmod** – получение списка загруженных модулей ядра

**modprobe -l** – получение списка всех доступных модулей ядра

**modprobe *modulename*** – загрузка модуля ядра

**modprobe -r *modulename*** – выгрузка модуля ядра

**modinfo *modulename*** – получение информации по модулю

Сами модуля ядра расположены в ***/lib/modules/\$(uname -r)***

Модули которые требуется загружать при загрузке системы следует указать в ***/etc/modules***



# Монтирование файловых систем

---

Любая файловая система которую планируется использовать должна быть подключена (смонтирована) к общему дереву каталогов.

Монтирование производится к любой выбранной директории, но следует иметь ввиду, что если директория не пустая, то после монтирования в нее файловой системы ее старое содержимое станет недоступно, до отключения (размонтирования) соответствующей файловой системы.

После монтирования, файлы находящиеся на смонтированной файловой системе будут отражены на содержимое директории в которую она смонтирована.

При подключении файловых систем допускается указывать параметры специфичные для данной ФС и необходимые для ее корректной работы.





# Монтирование файловых систем

---

**mount** – утилита для подключения файловых систем.

Опции:

**-a** – подключить все файловые системы описанные в **/etc/fstab**

**-t *fstype*** – указывает тип подключаемой файловой системы

**-o *options*...** - определяет опции для подключаемой файловой системы

***rw*** — подключение с правами на чтение и запись

***ro*** — подключение с правами на чтение

***remount*** — переподключение смонтированной файловой системы с новыми опциями



# Монтирование файловых систем Windows

---

Опции монтирования файловой системы **vfat**

***codepage=866*** – определяет кодировку в которой Windows сохраняет имена файлов

***iocharset=utf-8*** – определяет кодировку в которой работает Linux

***utf8=true*** – новая опция, заменяет две предыдущие.

Опции монтирования файловой системы **ntfs**

***nls=utf-8*** - определяет кодировку в которой работает Linux

***locale=ru\_RU.UTF-8*** – новая опция, заменяет предыдущую, указывает, что преобразование кодировок выполняется в соответствии с указанной локалью



# Отключение смонтированных файловых систем

---

**umount** – утилита для отключения (отмонтирования) смонтированных файловых систем.

В качестве параметра указывается точка монтирования или файл.

Опция:

**-a** – отключить все файловые системы описанные в **/etc/fstab**



# Повышение привилегий до суперпользователя

---

В **ASTRALinux** по умолчанию отключена возможность входа в систему для суперпользователя. Для выполнения команд с правами суперпользователя используется команда **sudo**

Примеры:

```
sudo ls -la /root  
sudo -i
```



# Просмотр содержимого директории

---

`ls` - утилита позволяет просмотреть содержимое директории в соответствии с указанными опциями

Опции:

- a* – показывать все файлы, в том числе и скрытые
- l* – вывести подробную информацию по каждому из файлов
- F* – для каждого имени каталога добавлять суффикс ``/'`, для каждого имени FIFO - ``|'` и для каждого имени исполняемого файла - ``*'`



# Пример использования umount и mount

---

```
mount
```

```
ls /media/cdrom
```

```
sudo umount /dev/sr0
```

```
ls /media/cdrom
```

```
sudo mount -o ro /dev/sr0 /media/cdrom
```

```
sudo mount -o utf8,remount /media/cdrom
```

```
ls /media/cdrom
```



# Утилиты настройки сети

---

*ifconfig* – утилита предназначена для конфигурации сетевых интерфейсов

Опции:

имя устройства ( сетевого интерфейса - eth0,eth1...)

IP-адрес интерфейса

маска подсети

широковещательный адрес

Пример:

```
ifconfig eth0 10.10.103.230 netmask 255.255.255.0 \  
broadcast 10.10.103.255 up
```



# Утилиты настройки сети

---

*route* – утилита настройки таблицы маршрутизации

Опции:

*add* – добавление маршрута в таблицу маршрутизации

*-net* – добавление маршрута к сети

*-host* – добавление маршрута к хосту

*default* – добавление маршрута по умолчанию

*del* – удаление маршрута из таблицы

*gw* – указание адреса шлюза

Примеры:

```
route add -net 192.168.10.0/24 gw 10.10.103.10
```

```
route add -host 172.16.171.3 gw 10.5.11.2
```

```
route add default gw 10.10.103.100
```





# Управление пользователями и их членством в группах

---

## Управление пользователями и группами

```
useradd -m -g users -G audio,video -s /bin/bash user1  
passwd user1  
groupadd mygroup1  
adduser user1 mygroup1
```



# Управление запуском сервисов (демонов)

---

Файлы конфигурации служб расположены в каталоге */etc/init*

Все скрипты располагаются в директории */etc/init.d* а функции используемые в стартовых скриптах в */lib/lsb/init-functions*

Символические ссылки на скрипты размещаются в */etc/rcx.d*, где *x* – соответствующий уровень выполнения. По умолчанию используется 2-й уровень выполнения, что соответствует каталогу */etc/rc2.d* поэтому если требуется отключить запуск сервиса на этом уровне — удаляется соответствующая символическая ссылка, а если требуется включить - создается.



# Изучение ключевых файлов конфигурации системы

---

Каталог */etc* является централизованным хранилищем настроек системы и приложений.

Настройки приложений хранятся в конфигурационных файлах, формат которых может сильно отличаться в зависимости от приложения.

Если приложение не предполагает иметь более одного конфигурационного файла, то оно располагает его непосредственно в */etc*.

Иначе, в */etc* создается каталог для размещения конфигурационных файлов приложения.



# Ключевые конфигурационные файлы системы

---

***/etc/fstab*** – определяет настройки для файловых систем подключаемых как в процессе загрузки системы, так и в процессе работы с ней, что актуально для сменных носителей.

***/etc/mtab*** – отражает настройки файловых систем смонтированных в настоящий момент, заполняется из */proc/mounts*, при любом монтировании или отмонтировании ФС.

***/etc/ld.so.conf*** – определяет пути поиска системных библиотек программой *ldconfig*, которая ведет их учет и преоставляет эту информацию приложениям по запросу.

***/etc/hosts*** – содержит соответствия между именами компьютеров и их IP-адресами

***/etc/resolv.conf*** – содержит настройки DNS-клиента

***/etc/host.conf*** и ***/etc/nsswitch.conf*** – содержат настройки порядка определения IP-адресов на основе доменных имен.

---



# Иерархия файловой системы

---

*/bin* – исполняемые файлы  
*/sbin* – исполняемые файлы  
*/dev* – файлы устройств  
*/etc* – файлы конфигурации  
*/lib* – системные библиотеки  
*/home* – каталоги пользователей  
*/root* – каталог суперпользователя  
*/usr* – размещение приложений  
*/var* – данные приложений

*/tmp* – временный каталог системы  
*/var/tmp* – временные каталоги приложений  
*/proc* – файловый интерфейс ОС  
*/opt* – аналог “Program Files”  
*/mnt* – монтирование сетевых ФС  
*/media* – монтирование съемных ФС  
*/boot* – загрузчик и ядро системы  
*/sys* – интерфейс к устройствам ОС  
*/srv* – размещение Web-сайтов, FTP...



# Основы работы в терминале

---

- Программы-оболочки
- Настройка терминала
- Основные команды для работы с файлами
- Примеры использования команд
- Стандартные ввод, вывод и вывод ошибок
- Дополнительные команды для работы с файлами
- Примеры использования дополнительных команд



# Настройка терминала

---

Программы оболочки:

Bourne shell (*sh*)

Korn shell (*ksh*)

Bourne again shell (*bash*)\*

C shell (*csh*)

TC shell (*tcsh*)

\*В Linux стандартной оболочкой по умолчанию является *bash*



# Настройка терминала

---

Переменные окружения:

*\$SHELL* – содержит путь к shell текущего пользователя

*\$LS\_COLORS* – определяет соответствие между расширениями файлов и теми цветами которыми те отражаются в при выводе командой ls

*\$USER* – текущий пользователь

*\$HOME* – домашний каталог пользователя *\$USER*

*\$PATH* – содержит пути для поиска файлов по умолчанию

*\$PWD* – указывает на текущий каталог

*\$LANG* – определяет текущие настройки локали





# Настройка терминала

---

Управление терминалом:

*set* – позволяет мониторить и управлять поведением функций и переменных окружения

*env* – выводит список переменных окружения

*export* – экспортирует переменные окружения, делая их доступными для других программ

*echo* – выводит на терминал то, что передано в качестве параметра, в том числе и *esc-последовательности*\*

*reset* – возврат настроек терминала к значениям по умолчанию

*logout* – завершение текущего пользовательского сеанса

*exit* – завершение сеанса работы с оболочкой

\*традиционным способом управления терминалом является отправка на него *esc-последовательностей*, для чего ***echo*** выполняется с ключами ***-ne***



# Основные команды для работы с файлами

---

*ls* – вывод содержимого каталога

*pwd* – выводит на консоль путь к текущему каталогу

*cd* – смена текущего каталога

*touch* – создание файла или изменение его временных меток

*mkdir* – создание каталога

*rm / rmdir* – удаление файла / каталога, поддерживается

рекурсия

*cp / mv* – копирование / переименование / перенос файлов и каталогов, поддерживается рекурсия

*more / less* – постраничный просмотр текстовых файлов

*ln* – создание ссылок на файлы (hard & soft)

*cat / tac* – вывод содержимого файла в прямом и обратном порядке



# Примеры использования команд

---

```
ls -alF /etc
```

```
pwd
```

```
cd /etc
```

```
pwd
```

```
cd ~
```

```
touch test
```

```
ls -l test
```

```
mkdir -p dir1/dir2/dir3
```

```
cp test dir1/dir2
```

```
mv test mytest
```

```
rmdir dir1/dir2/dir3
```

```
cat /etc/passwd
```

```
tac /etc/group
```

```
more /etc/services
```

```
less /etc/syslog.conf
```

```
ln mytest test
```

```
ln -s dir1/dir2/test mytest2
```

```
ls -l *test*
```

```
rm mytest
```

```
rm -rf dir1
```

```
ls -l *test*
```

```
rm *test*
```



# Стандартные ввод, вывод и вывод ошибок

---

Файлы с индексными дескрипторами:

0 – стандартный ввод (stdin)

1 – стандартный вывод (stdout)

2 – стандартный вывод ошибок (stderr)



# Перенаправление ввода-вывода и ошибок

---

```
cat > testfile
```

Введите строку и нажмите Enter

Нажмите Ctrl+D – (EOF) для завершения работы

```
cat testfile > testfile 2> errfile
```

```
cat errfile
```

```
cat /etc/passwd > testfile2
```

```
cat < /etc/group > testfile
```

```
ls -l /etc > mylist
```

```
touch /bin/mycustomfile 2> errfile
```

```
cat errfile
```



# Дополнительные команды для работы с файлами

---

*df* - отчёт об использовании дискового пространства

*du* - оценка места на диске, занимаемого файлами и каталогами

*sort* – сортировка строк в текстовых файлах

*cut / paste* – работа с секциями файлов (вырезать / вставить)

*head / tail* – вывод (первых / последних) строк файла на стандартный вывод

*wc* – подсчет (размера файла, числа символов, слов, строк и т.п.)

*tr* – замена символов по шаблону

*dd* - преобразовать и копировать файл

*tee* – трансляция *stdin* в *stdout* с ведением лога

*uniq* – нахождение дублирующихся строк

*grep* – поиск по шаблону



# Примеры использования дополнительных команд

---

```
df -h
du -h /var/log
ls /etc | sort | less
ls /etc/*.conf | wc -l
cat /etc/services | head
ls -l /etc | tr 'rwx' 'RWX'
ls -l /etc | tee test | tail
wc -c test
dd if=/dev/cdrom of=~ /my.iso
cat | uniq -d
grep -rsni pppd
/usr/share/doc
```

```
cat > test
line1:the 1st
line2:the 2nd
line3:the 3rd
{нажмите Ctrl+D}
```

```
cut -f 1 -d: test > tmp1
cut -f 2 -d: test > tmp2
paste tmp2 tmp1 > test
rm tmp* && cat test
```



# Cron

---

Представляет собой демон обеспечивающий выполнение заданий по расписанию\*.

Конфигурационный файл `/etc/crontab` (глобально) и файлы в `/var/spool/cron/*` по файлу на каждого пользователя

Структура конфигурационного файла:

*Min Hour DayOfMonth Month DayOfWeek process*

В данных полях можно использовать следующие значения:

*Min – 0-59 (можно указывать дробные значения)*

*Hour – 0-23*

*DayOfMonth – 1-31*

*Month – 1-12*

*DayOfWeek – 0-7 (0 и 7 – воскресенье)*

*\*Задания, которые были просрочены из-за того, что компьютер был выключен – не выполняются*





# Содержимое /etc/crontab

---

```
# m h dom mon dow user  command
17 * * * *   root    cd / && run-parts --report /etc/cron.hourly
25 6        * * *   root    test -x /usr/sbin/anacron || ( cd / &&
run-parts --report /etc/cron.daily )
47 6        * * 7    root    test -x /usr/sbin/anacron || ( cd / &&
run-parts --report /etc/cron.weekly )
52 6        1 * *   root    test -x /usr/sbin/anacron || ( cd / &&
run-parts --report /etc/cron.monthly )
```



# Пример настройки Cron

---

```
sudo su -
```

```
crontab -e
```

**#добавьте строку**

```
*/2 * * * * /bin/date >> /var/log/date.log
```

**Ctrl+O Ctrl+X**

**Ждите 2 минуты, затем смотрите содержимое лога**

```
cat /var/log/date.log
```



# Резервное копирование пользовательских данных

---

Резервное копирование домашних каталогов  
пользователей

```
sudo tar cjvf /root/home.tar.bz2 /home/user1
```

Удаление домашних каталогов  
пользователей

```
sudo rm -fR /home/user1
```

Восстановление данных из архива

```
sudo tar xjvf /root/home.tar.bz2 -C /
```

**Спасибо за внимание!**

