

# Основы администрирования и безопасности Linux

УЦ "Специалист"

Программа соответствует требованиям  
международного стандарта сертификации  
Linux Professional Institute Level 1 (LPIC-1).



# Инсталляция системы

---

- Выбор языка для инсталлятора
- Выбор текущей локали и региональных настроек
- Выбор раскладки клавиатуры
- Разметка диска
- Заведение пользовательского аккаунта
- Установка системы



# 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



# Система помощи

---

man

--help

info

Документация к программам



# man [опции] [раздел] manpage

---

Программа предназначена для просмотра страниц руководства (manpages).

man присутствует во всех версиях UNIX и является старейшей системой помощи.

Для получения справки о программе, функции, формате файла, в командной строке

необходимо набрать man имя\_программы.

Документация хранится в специально форматированных текстовых файлах, в директории /usr/share/man.



# Разделы man

---

- man1 Системные утилиты общего пользования
- man2 Функции системы
- man3 Библиотечные функции
- man4 Описание устройств
- man5 Форматы конфигурационных файлов
- man6 Игры
- man7 Различные описания
- man8 Административные утилиты
- man9 Дополнительная документация по ядру



## --help

---

Для получения краткой информации о программе, написанной сообществом GNU, следует использовать параметр --help.

Примеры:

ls -help

cat --help





# info [menu-item]

---

Система помощи, разработанная сообществом GNU.

В основном содержит описание программ, созданных GNU сообществом.

Информация хранится в специально отформатированных текстовых файлах.

В отличие от программы man, info позволяет создавать меню и переходы. Система, чем-то напоминает WEB страницы.

---



# Документация к программам

---

С программами, входящими в дистрибутивы, поставляется документация.

Документация к программам находится в директории `/usr/share/doc`. В ней находятся директории с именами программ, в которых, собственно, и расположена документация по конкретной программе.



# Терминология

---

- **Терминал** — устройство ввода/вывода, рабочее место на многопользовательских ЭВМ, монитор с клавиатурой
- **Оболочка** операционной системы (от англ. shell - оболочка) — интерпретатор команд операционной системы (ОС), обеспечивающий интерфейс для взаимодействия пользователя с функциями системы
- **Консоль** — интерфейс командной строки в котором инструкции компьютеру даются только путём ввода с клавиатуры текстовых строк (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



# Работа в терминале

---

## Команды оболочки:

- `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
```

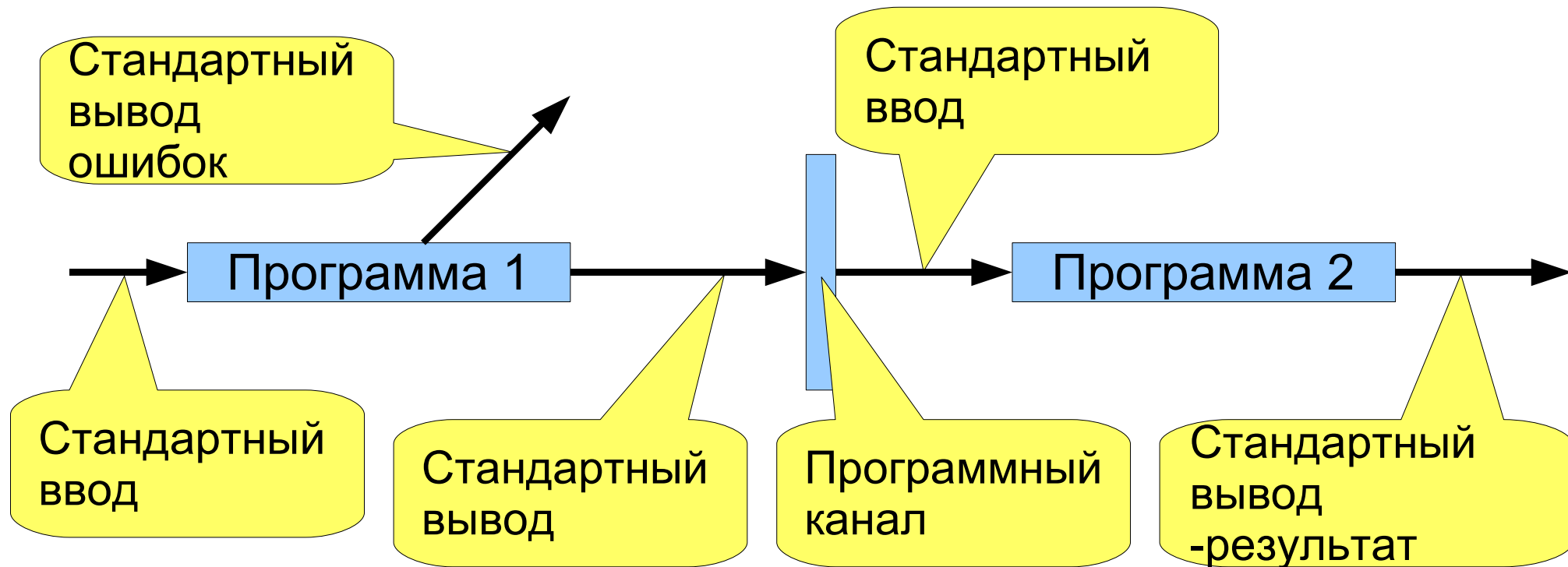
```
ls -l *test*
```

```
rm *test*
```





# Стандартные ввод, вывод и вывод ошибок



Стандартные потоки (файлы):

- 0 – Стандартный ввод (stdin)
- 1 – Стандартный вывод (stdout)
- 2 – Стандартный вывод ошибок (stderr)

Программный канал (конвейер) направляет вывод одной программы на ввод другой.



# 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
```



# Типы файлов

---

Тип файла можно определить по первой букве вывода программы `ls -l`.

- `f` или `-` — обыкновенный файл
- `l` — символическая ссылка
- `d` — директория
- `c` — символьное устройство
- `b` — блочное устройство
- `p` — pipe (FIFO) файл
- `s` — файл типа socket



# Управление правами доступа

---

Система безопасности UNIX построена на определении прав доступа к файлам.

- `chmod` — изменение прав доступа
- `umask` — маска прав доступа
- `chown` — изменение хозяина
- `chgrp` — изменение группы



# Типы прав доступа к файлам

---

r — право на чтение из файла

w — право на запись в файл

x — право на исполнение



# Интерпретация прав доступа к каталогам

---

r — право на просмотр содержимого директории

w — право на создание, удаление файлов в директории

x — право на «прохождение» в и сквозь директорию





# Числовой формат записи прав

---

R	W	X	
0	0	1	1
0	1	0	2
1	0	0	4

**660**     **`rw-rw-----`**

**744**     **`rwxr--r--`**



# Символьный формат записи прав

---

**ugo**a + - = **rwX**

**rw-rw-----**

**o+r**

**rw-rw-r--**

**g-w**

**rw-r--r--**

**ug+x**

**rwXr-Xr--**

**o=rw**

**rwXr-Xrw-**

**u** — права хозяина

**g** — права группы

**o** — права всех остальных

**a** — все права

**+** — установить бит

**-** — сбросить бит

**=** — установить относительно

**r** — право на чтение

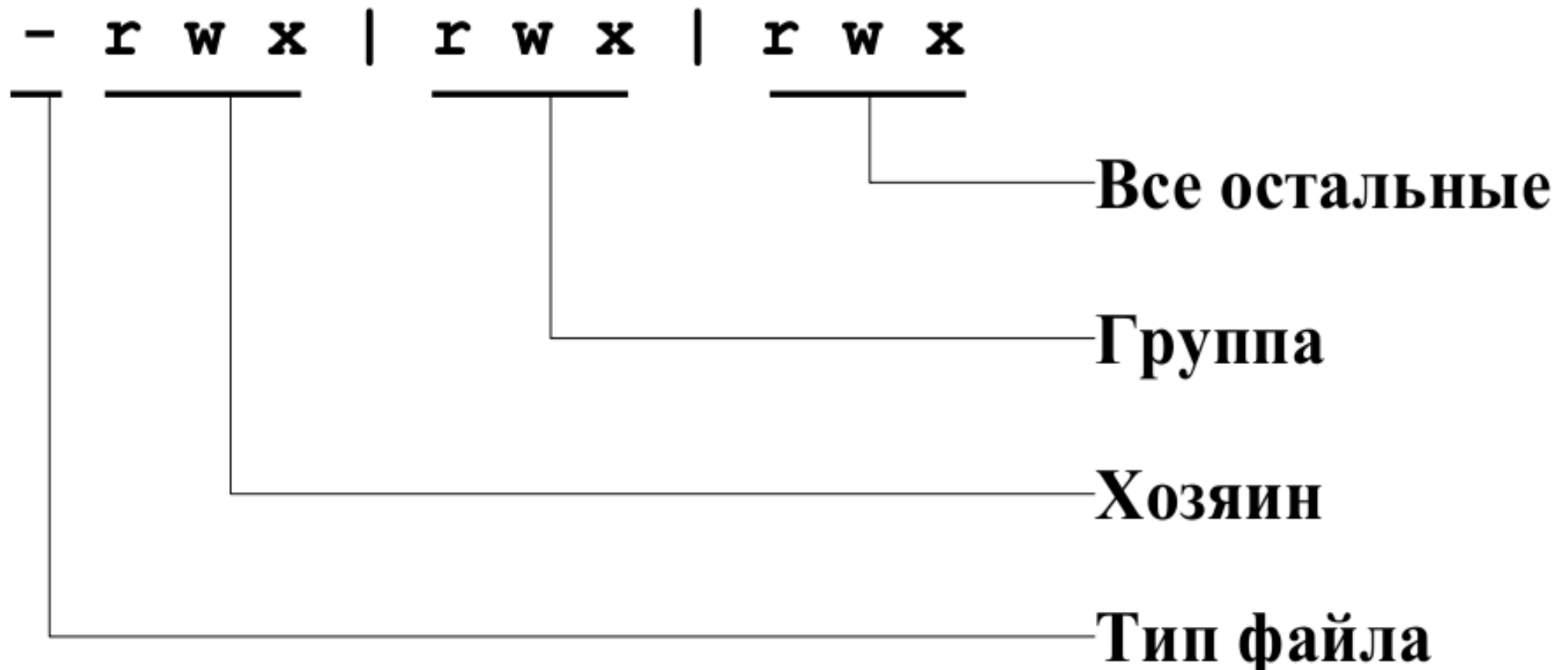
**w** — право на запись

**x** — право на исполнение



# Система безопасности

**Система безопасности UNIX построена на определении прав доступа к файлам**





# `chmod [-R] [опции] права файл ...`

---

Программа предназначена для изменения прав доступа.

Обыкновенный пользователь может менять права только у принадлежащих ему файлов.

Суперпользователь может изменять права у всех файлов системы.



`chown [-R] user[:group] файл ...`

---

Программа изменяет хозяина файла.

Доступна только для суперпользователя.

Опции:

- R — рекурсивная смена пользователя.



## chgrp [-R] group файл ...

---

Программа изменяет группу файла.

Обыкновенный пользователь может устанавливать только те группы, членом которых он является.

Опции:

- R — рекурсивная смена группы.



# Специальные права

---

SUID	4
SGID	2
Sticky	1

- **SUID** — программы выполняются с правами хозяина файла.
- **SGID** — программы выполняются с правами группы файла. Для директорий — создаваемые файлы принадлежат группе, которой принадлежит директория.
- **Sticky bit** — в директории файл может удалить только его хозяин.



# Права доступа по умолчанию

---

Для директорий — 777

Для файлов — 666





## umask [маска]

---

Встроенная в shell команда, позволят определить маску для вновь создаваемых файлов.

Маска — это число, которое необходимо вычесть из прав доступа по умолчанию, для получения реальных прав вновь создаваемых файлов.



# POSIX ACL

---

Все современные файловые системы Linux поддерживают POSIX ACL.

POSIX ACL позволяет указать права доступа для конкретных пользователей и групп.



# getfacl [опции] file ...

---

Программа показывает список ACL, установленных на файл.

Опции:

- R — рекурсивный просмотр.



# setfacl [опции] file ...

---

Программа устанавливает и удаляет ACL в

указанном файле.

Опции:

- R — рекурсивный просмотр.
- m — изменение или установка ACL.
- x — удаление ACL.



# Примеры установки ACL

---

Установка права чтение, запись для пользователя user1:

```
setfacl -m u:user1:rw file
```

Установка права на чтение для группы users:

```
setfacl -m g:users:r file
```

Установка маски в значение чтение, запись:

```
setfacl -m m::rw file
```



# Процессы

---

- Список процессов
- Сигналы
- Мониторинг
- Приоритеты
- Приостановка выполнения



# Список процессов

---

- Каждому выполняемому процессу
- присваивается уникальный номер — PID
- (Process ID)
- После завершения процесса PID освобождается



# Потомок — Родитель

---

У всех процессов в системе, кроме самого первого, есть «родители» — процессы, которые запускают данный процесс.

Любой потомок может запустить другой процесс, для которого он будет родителем.

Самый первый процесс в системе — `init` с `PID=1`.

После завершения работы родительского процесса у потомка родителем становится процесс `init`.





# Потомок — Родитель

---

Если shell заканчивает свою работу, все процессы, запущенные в этой оболочке будут завершены.

Для того, что бы программа продолжала работу после закрытия оболочки, ее необходимо запускать при помощи программы `nohup`.



# nohup программа

---

nohup отключает программу от терминала, что позволяет продолжить выполнение программы после его выключения.

Пример запуска программы:

```
nohup programt -p1 -p2
```



# Демоны

---

Демоны написаны таким образом, что сразу после запуска отключаются от терминала и могут продолжать работать после выхода пользователя из оболочки.



## ps [опции]

---

Показывает список процессов в системе.

Опции:

- help — выводит экран помощи.

- a — показывает список всех процессов «привязанных» к терминалам.

- x — показывает список процессов не «привязанных» к терминалу.

- e — показывает все процессы системы.

- f — показывает дерево процессов.

- u user — список процессов

пользователя.

---



# pstree

---

Показывает дерево процессов.

Примеры:

`ps`

`ps xa | less`

`ps -e`

`ps xaf`

`pstree`

`ps xa | grep cupsd`

`pgrep cupsd`



# Сигналы

---

Процессы могут «общаться» друг с другом при помощи сигналов.

Сигнал — это число, которое одна программа

может послать другой программе.

Реакция программы на получаемый сигнал

зависит от программиста, написавшего ее.

Для того, чтобы послать сигнал процессу можно воспользоваться программой `kill`.



# kill [-сигнал] PID ...

---

Программа посылает сигнал процессу.

Процесс определяется его PID.

Опции:

- l — показать список всех сигналов в системе.

Примеры:

```
ps
```

```
kill PID_программы_bash
```

```
kill -9 PID_программы_bash
```



## killall [-сигнал] имя ...

---

Программа посылает сигнал процессу.

Процесс определяется по его имени.

Пример:

```
killall firefox
```





# Режимы работы программы

---

Программа может работать в режимах:

- foreground** — занимает консоль пользователя.

- background** — запускается как параллельный процесс. После запуска программы пользователю доступна командная строка.



# Режимы работы программы

---

Для запуска программы в `background` режиме в конце командной строки необходимо написать символ `&`

Если программа, запущенная в `background`-режиме попытается что либо прочитатъ со стандартного ввода, ее выполнение будет приостановлено.



# Управление задачами

---

Ctrl+Z — приостановка выполнения программы

jobs — показывает список приостановленных и запущенных в background режиме программ.

fg [число] — продолжает выполнение программы в foreground режиме.

bg [число] — продолжает выполнение программы в background режиме.



# 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.
- **/etc/syslog.conf** – содержит настройки системной службы ведения журналов



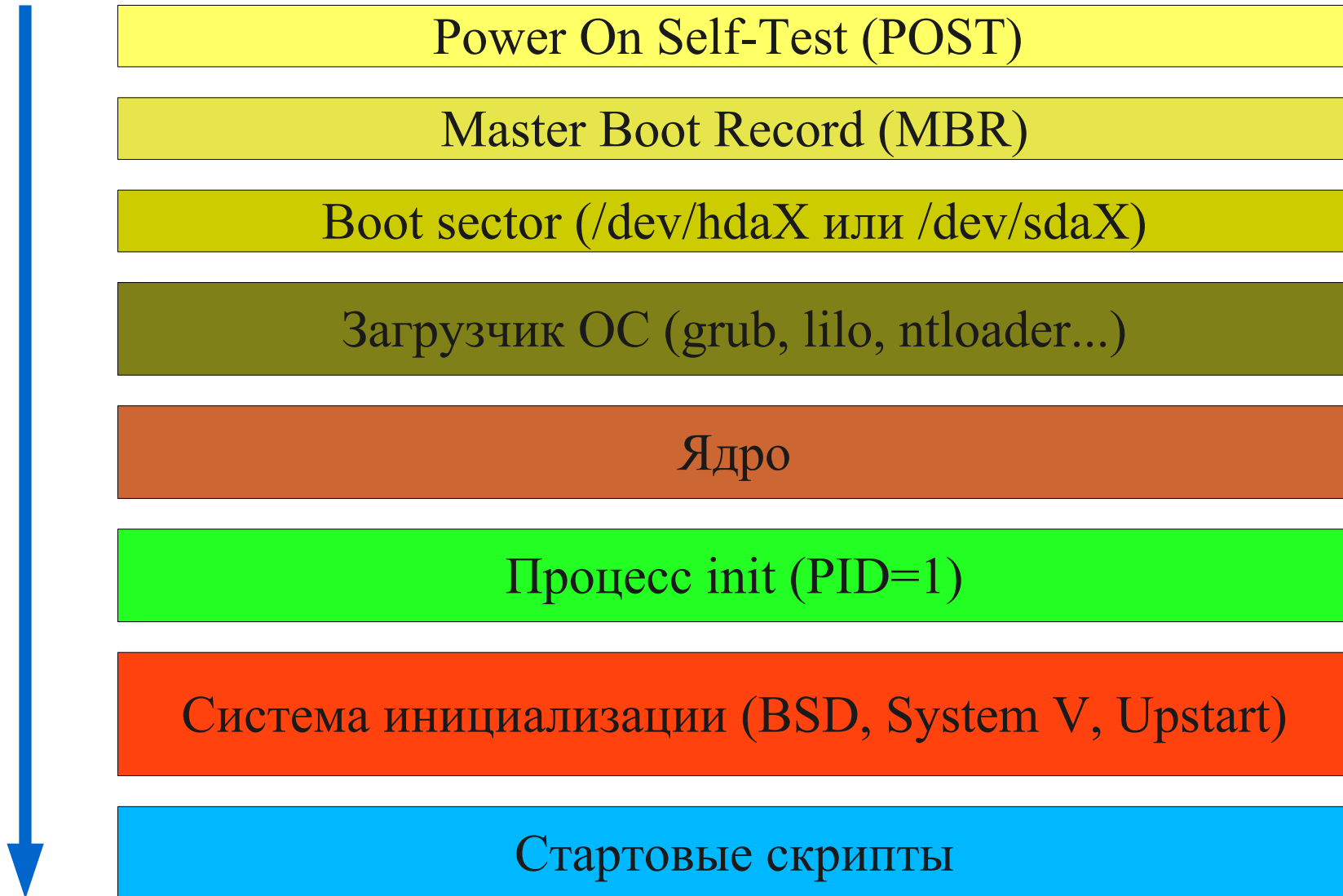
# Загрузка системы

---

- Как осуществляется загрузка системы
- Как выглядит загрузка системы
- Что происходит “за кулисами”
  - порядок загрузки системы
  - классические системы инициализации (BSD / System V)
  - система инициализации upstart
  - уровни выполнения
  - установка оборудования
  - повышение полномочий и работа с правами суперпользователя
  - монтирование файловых систем



# Как осуществляется загрузка системы





# Что происходит “за кулисами”

---

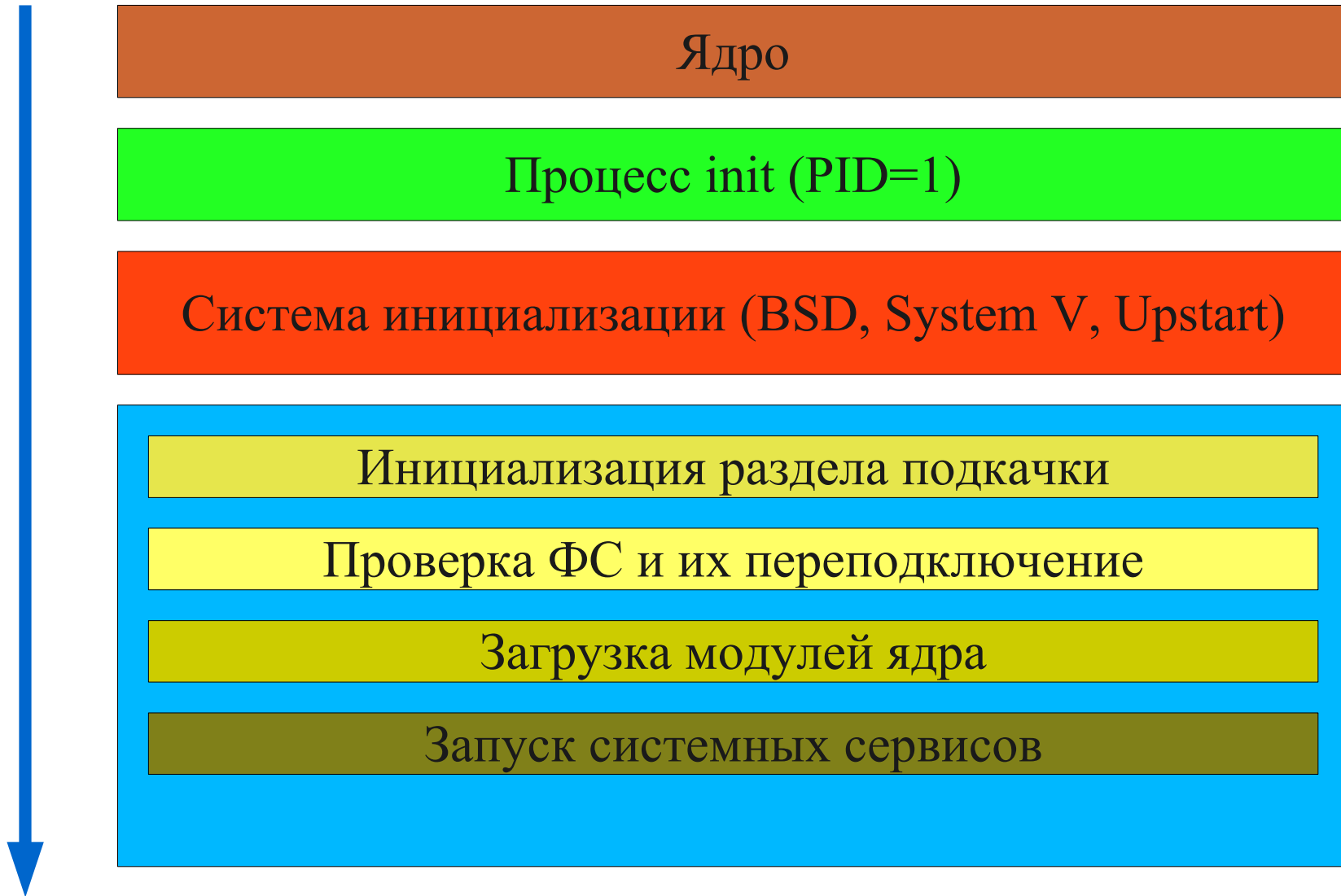
Ядро пытается загрузить процесс `init`, для ЭТОГО:

- 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/event.d`
  - 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
-



# Уровни выполнения

---

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



# Повышение привилегий до суперпользователя

---

- В Ubuntu по умолчанию отключена возможность входа в систему для суперпользователя
- Для выполнения команд с правами суперпользователя используется команда `sudo`

Пример:

```
sudo su -
```



# 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=utf8** – определяет кодировку в которой работает Linux

Пример:

```
mount -t vfat -o codepage=866,iocharset=utf8 /dev/sd1 /mnt
```

- Опция монтирования файловой системы **ntfs**
  - **nls=utf8** - определяет кодировку в которой работает Linux

Пример:

```
mount -t ntfs-3g -o nls=utf8 /dev/sd1 /mnt
```



# Отключение смонтированных файловых систем

---

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

В качестве параметра указывается точка монтирования или файл.



# Пример использования umount и mount

---

- Вставьте CD или DVD диск
- Дождитесь пока откроется окно с его содержимым
- Закройте окно
- Запустите терминал и все дальнейшие действия выполняйте в нем

```
ls /media/cdrom0
```

```
sudo umount /media/cdrom0
```

```
ls /media/cdrom0
```

```
sudo mount -t iso9660 -o ro /dev/cdrom /media/cdrom0
```

```
ls /media/cdrom0
```



# Настройка системы после установки

---

- Настройка сети
- Управление пользователями и их членством в группах
- Управление запуском сервисов (демонов)
- Изучение ключевых файлов конфигурации системы



# Утилиты настройки сети

---

`ifconfig` – утилита предназначена для конфигурации сетевых интерфейсов

Опции:

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

IP-адрес интерфейса

маска подсети

Пример:

```
ifconfig eth0 172.16.1.X/24 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 172.16.1.254
```

```
route add -host 10.10.1.1 gw 172.16.1.254
```

```
route add default gw 172.16.1.254
```



# Конфигурация сети

---

- `/etc/network/interfaces` – настройка сетевых интерфейсов (адаптеров)
- `/etc/resolv.conf` -файл настройки DNS-клиента
- `/etc/hostname` – настройка имени хоста
- `/etc/hosts` – соответствие IP адресов именам хостов (предок DNS)



# Настройка сети

---

Настройка сети без привязки к дистрибутиву:

```
sudo nano /etc/init.d/rc.local
```

**# Добавьте эти строки в файл**

```
ifconfig eth0 172.16.1.X/24 up
```

```
route add default gw 172.16.1.254
```

```
echo "search any.com" > /etc/resolv.conf
```

```
echo "nameserver 172.16.1.254" >> /etc/resolv.conf
```

```
hostname c230
```





# Управление пользователями и их членством в группах

---

## Управление пользователями и группами без использования GUI

```
useradd -m -g cdrom -G audio,video -s /bin/bash user1
```

```
passwd user1
```

```
groupadd mygroup
```

```
adduser user2 mygroup
```



# Управление запуском сервисов (демонов)

---

Файлы конфигурации служб расположены в каталоге `/etc/event.d`

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 в Ubuntu 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.



# Регистрация активности в системе

---

В состав системы журнальной регистрации входят:

- Демон `syslogd`
- Библиотеки, с помощью которых программы могут отсылать сообщения демону `syslogd`
- Программа `logger` предназначенная для отправки сообщений демону `syslogd`

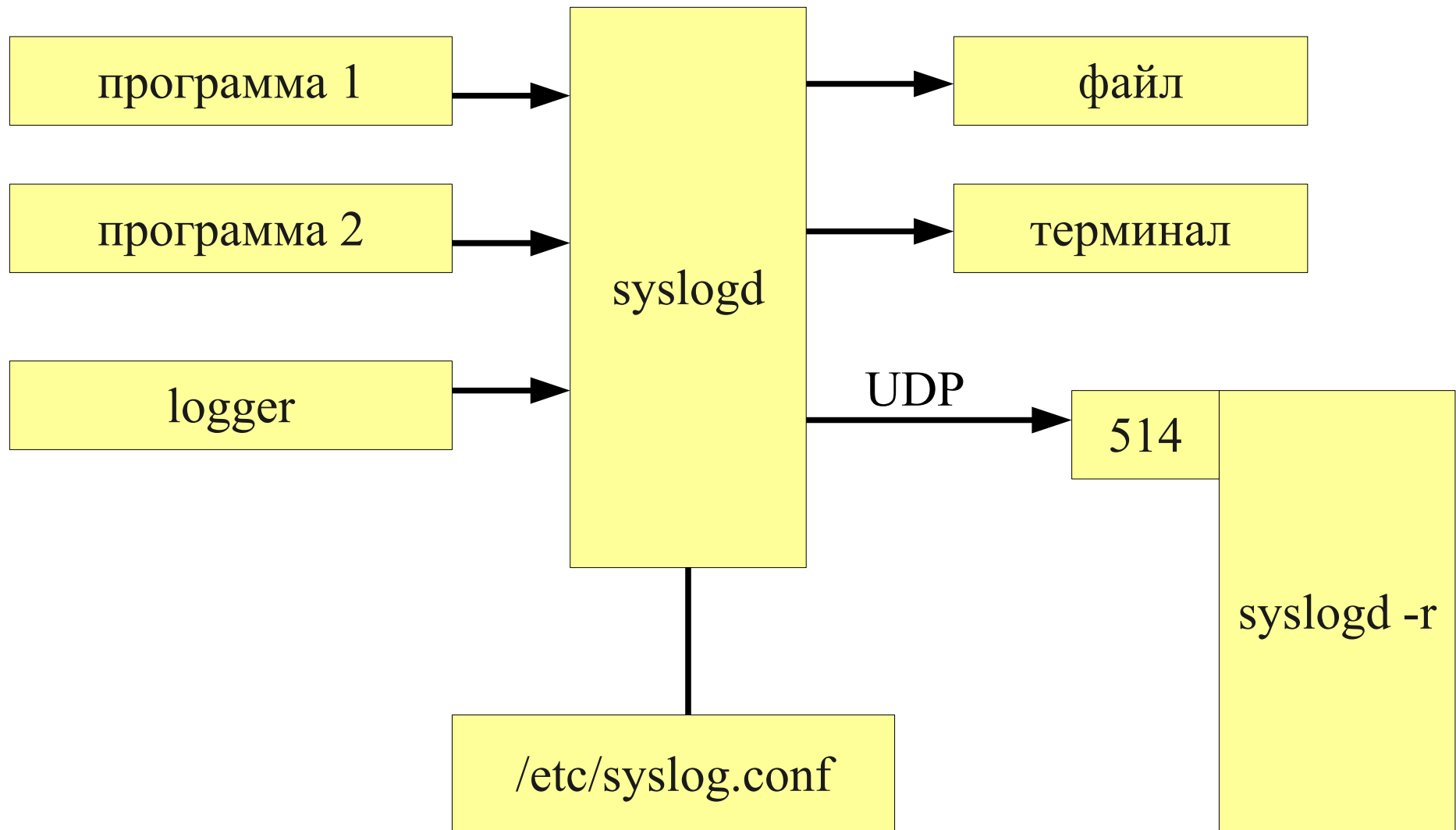
Информация, полученная от программ, фильтруется демоном `syslogd` в соответствии с настройками фильтров, указанными в `/etc/syslog.conf`.

Отфильтрованные сообщения могут отправляться

- В файл
- На терминал
- На другой хост, с запущенным демоном `syslogd`



# Регистрация активности в системе





# Формат фильтра syslogd

---

## Формат строки файла syslog.conf

- Фильтр
- Действие

{в качестве разделителя используется символ табуляции}

## Примеры:

auth.\*                /var/log/auth.log

daemon.\*             /var/log/daemon.log



# Запись в журнал из скрипта

---

```
nano log_write
```

```
#!/bin/bash
```

```
logger -p auth.notice -t $0 $1
```

```
{сохраните файл Ctrl+O, затем Ctrl+X }
```

```
chmod +x log_write
```

Пример использования:

```
./log_write "test"
```



# Ротация журналов

---

С каждым дистрибутивом Linux поставляется утилита `logrotate*`, которая позволяет ограничивать размер журнальных файлов и сохранять архивные копии журналов за предыдущие периоды, это утилита – `logrotate`.

**Ротация** – это процесс архивации журнала по достижении одного из заданных условий: размера файла или временного периода и последующей очистки текущего файла журнала, что позволяет контролировать его размер.

---

\*Конфигурационный файл программы - `/etc/logrotate.conf`



# Возможности logrotate

---

Ротация файлов производится в соответствии с условиями:

- раз в день (daily)
- раз в неделю (weekly)
- раз в месяц (month)
- при превышении определенного размера

Утилита умеет выполнять следующие действия:

- хранить указанное число экземпляров журнальных файлов (архивы за период)
- отсылать по почте файл, который будет подвергнут ротации, с последующим его удалением
- до и после ротации запускать на выполнение программы





# Настройка logrotate.conf

---

**compress** – сжатие файлов журнала после ротации с помощью gzip

**create [mode] [owner] [group]** – после ротации файл журнала имеет указанных владельца, группу и режим доступа

**include** – включение содержимого указанного файла в основной конфигурационный файл.

**email** – указывает почтовый ящик на который высылается файл лога после ротации

**mailfirst / maillast** – отсылать по почте 1-ю / последнюю копию журнального файла

**missingok** – если файла лога нет, то перейти к обработке следующего, не выдавая сообщения об ошибке

**prerotate / postrotate** – определяет программы, которые должны быть выполнены перед началом ротации

**rotate** – определяет количество хранимых журнальных файлов за прошлые периоды / события

**sharedscripts** – позволяет выполнить prerotate / postrotate программы единовременно после завершения ротации всех логов

**daily / weekly / monthly** - определяют частоту ротации

**size** – устанавливает ограничение на размер лога



# Пример настройки logrotate.conf

---

```
sudo su -  
nano  
/etc/logrotate.d/messages  
/var/log/messages {  
compress  
size=100  
}
```

Запустите команду ротации:  
`logrotate /etc/logrotate.conf`

Проверьте содержимое  
директории `/var/log`

`exit`

Сохраните файл



# Выполнение заданий по расписанию

---

Существуют 3 стандартных программы для выполнения заданий по расписанию:

- cron
- anacron\*
- at

*\*Не поставляется с Ubuntu Server*

---



# 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

---

```
crontab -e
```

```
#добавьте строку
```

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

```
Ctrl+O Ctrl+X
```

Ждите 2 минуты, затем смотрите содержимое  
лога

```
cat /tmp/date.log
```



# Anacron

---

- Представляет собой демон обеспечивающий выполнение заданий по расписанию, как и Cron, но существенно отличается от него\*.
- Конфигурационный файл `/etc/anacrontab`
- Структура конфигурационного файла:  
`days minutes id_task process`
- В данных полях можно использовать следующие значения:
  - `days` – периодичность в днях, например – 7 (раз в неделю)
  - `minutes` – 0-59
  - `id_task` – строковый идентификатор задачи

\*Задания, просроченные из-за выключения компьютера выполняются сразу по его включению

---



# Содержимое /etc/anacrontab

---

# Настройки по умолчанию

```
1    5  cron.daily    nice run-parts --report /etc/cron.daily
7    10 cron.weekly  nice run-parts --report /etc/cron.weekly
```





# Утилиты для работы с сетью

---

- **arp** – просмотр и настройка таблицы соответствия mac и ip-адресов
- **ping** – утилита для проверки доступности хостов в сети
- **traceroute** – утилита для отслеживания маршрута от одного хоста до другого
- **netstat** – просмотр статистики по сетевым интерфейсам, отчетов по сетевым подключениям, службам и маршрутизации пакетов
- **nslookup** – позволяет взаимодействовать с DNS-серверами
- **nmap** – сканер портов на предмет поиска уязвимостей, с целью их устранения
- **tcpdump** – утилита для прослушивания сетевого трафика
- **iptraf** – многофункциональная утилита мониторинг сетевого трафика
- **wireshark** – средство анализа сетевых протоколов



# arp

---

- **arp** – просмотр и настройка таблицы соответствия mac и ip-адресов
- Опции:
  - **a [hostname]** – показывает значение соответствия mac и ip-адреса для указанного хоста. Если не указать хост, будут показаны все значения таблицы.
  - **d hostname** – удаляет запись из таблицы
  - **s hostname mac** – вручную добавляет запись в таблицу
- Пример:
  - **arp -a**  
*ivanova (192.168.213.24) at 00:1A:4D:41:0F:F5 [ether] on eth0*  
*petroff (192.168.213.213) at 00:18:71:71:96:66 [ether] on eth0*  
*sidorov (192.168.213.89) at 00:1A:4D:41:09:DF [ether] on eth0*



# ping

- **ping** – утилита для проверки доступности хостов в сети
- Опции:
  - R – включить опцию сохранения маршрута в передаваемых пакетах
  - b – разрешить широковещательную рассылку
  - c – ограничить число отправляемых пакетов
  - i – установить интервал между отправкой пакетов (по умолчанию 1 секунда)
  - s – установить размер пакета (по умолчанию 56 байт)
- Пример:  
**ping -c4 www.rbc.ru**  
PING www.rbc.ru (194.186.36.229) 56(84) bytes of data.  
64 bytes from www-gnocci.rbc.ru (194.186.36.229): icmp\_seq=1 ttl=56 time=4.60 ms  
64 bytes from www-gnocci.rbc.ru (194.186.36.229): icmp\_seq=2 ttl=56 time=4.54 ms  
64 bytes from www-gnocci.rbc.ru (194.186.36.229): icmp\_seq=3 ttl=56 time=4.63 ms  
64 bytes from www-gnocci.rbc.ru (194.186.36.229): icmp\_seq=4 ttl=56 time=4.46 ms  
  
--- www.rbc.ru ping statistics ---  
4 packets transmitted, 4 received, 0% packet loss, time 3002ms  
rtt min/avg/max/mdev = 4.467/4.561/4.633/0.062 ms



# Установка traceroute

---

Данная утилита не устанавливается по умолчанию, поэтому надо ее доустановить:

```
sudo apt-get install traceroute*
```



# traceroute

---

- ***traceroute*** – утилита для отслеживания маршрута от одного хоста до другого
- Опции:
  - n – отключить преобразование ip-адресов в DNS-имена
  - m – установка максимального количества контрольных точек (хопов) через которые пройдет отправленный пакет (по умолчанию 30)
- Пример:

```
traceroute -n www.1web.ru
```

```
traceroute to www.1web.ru (213.152.131.199), 30 hops max, 40 byte packets
```

```
1 192.168.1.1 (192.168.1.1) 1.133 ms 1.415 ms 1.882 ms
2 213.219.200.4 (213.219.200.4) 5.898 ms 6.945 ms 8.785 ms
3 213.219.200.1 (213.219.200.1) 9.552 ms 11.449 ms 8.424 ms
4 193.232.244.209 (193.232.244.209) 10.238 ms 13.247 ms 11.186 ms
5 213.152.128.81 (213.152.128.81) 12.578 ms 15.621 ms 16.070 ms
6 213.152.131.199 (213.152.131.199) 17.009 ms 16.889 ms 18.970 ms
```



# netstat

***netstat*** – просмотр статистики по сетевым интерфейсам, отчетов по сетевым подключениям, службам и маршрутизации пакетов

## Опции:

- n - отключить преобразование ip-адресов в DNS-имена
- l – показать порты, открытые для прослушивания
- i – показать статистику по сетевым интерфейсам
- r – показать таблицу маршрутизации
- s – показать статистику по каждому протоколу
- p – показывает имя и PID-программы

## Пример:

```
netstat -i
```

Kernel Interface table

Iface	MTU	Met	RX-OK	RX-ERR	RX-DRP	RX-OVR	TX-OK	TX-ERR	TX-DRP	TX-OVR	Flg
eth0	1500	0	0	0	0	0	0	0	0	0	BMU
eth1	1500	0	639668	0	0	0	445915	0	0	0	BMNRU
lo	16436	0	617	0	0	0	617	0	0	0	LRU



# nslookup

---

- ***nslookup*** – позволяет взаимодействовать с DNS-серверами
- Пример:

nslookup [www.specialist.ru](http://www.specialist.ru)

Server: 10.0.0.1

Address: 10.0.0.1#53

Non-authoritative answer:

www.specialist.ru canonical name = websrv.specialist.ru.

Name: websrv.specialist.ru

Address: 213.189.207.228



# Установка nmap

---

Данная утилита не устанавливается по умолчанию, поэтому надо ее доустановить:

```
sudo apt-get install nmap
```





# nmap

**nmap** – сканер портов на предмет поиска уязвимостей, с целью их устранения.

- Опции:
  - A – включить распознавание ОС и ее версии
  - sU – сканировать UDP-порты
  - sT – сканировать TCP-порты
- Пример:

**nmap -A my.router**

**Starting Nmap 4.20 ( <http://insecure.org> ) at 2008-08-26 14:39 MSD**

**Interesting ports on my.router (10.0.11.18):**

**Not shown: 1692 closed ports**

PORT	STATE	SERVICE	VERSION
------	-------	---------	---------

22/tcp	open	sshd	
--------	------	------	--

53/tcp	open	domain	ISC Bind dnsmasq-2.22
--------	------	--------	-----------------------

80/tcp	open	http	Linksys wireless-G WAP http config (Name WL500g.Deluxe)
--------	------	------	---

5000/tcp	open	UPnP?	
----------	------	-------	--

9100/tcp	open	jetdirect?	
----------	------	------------	--

**Service detection performed. Please report any incorrect results at**

**<http://insecure.org/nmap/submit/> .**

**Nmap finished: 1 IP address (1 host up) scanned in 111.471 seconds**



# Способы установки ПО

---

- Установка ПО из пакетов
    - Пакетные менеджеры – rpm, dpkg, pkg
    - Продвинутые пакетные менеджеры – apt-get, yum, yast2
  - Установка из исходных кодов
    - Утилита make
    - Сборка и установка ПО
  - Установка бинарных файлов из архивов
    - С использованием инсталлятора
    - Распаковка в корневой директории
-



# Файл /etc/apt/sources.list

---

```
deb http://ru.archive.ubuntu.com/ubuntu/ hardy main restricted
deb-src http://ru.archive.ubuntu.com/ubuntu/ hardy main restricted
deb http://ru.archive.ubuntu.com/ubuntu/ hardy-updates main restricted
deb-src http://ru.archive.ubuntu.com/ubuntu/ hardy-updates main restricted
deb http://ru.archive.ubuntu.com/ubuntu/ hardy universe
deb-src http://ru.archive.ubuntu.com/ubuntu/ hardy universe
deb http://ru.archive.ubuntu.com/ubuntu/ hardy-updates universe
deb-src http://ru.archive.ubuntu.com/ubuntu/ hardy-updates universe
deb http://ru.archive.ubuntu.com/ubuntu/ hardy multiverse
deb-src http://ru.archive.ubuntu.com/ubuntu/ hardy multiverse
deb http://ru.archive.ubuntu.com/ubuntu/ hardy-updates multiverse
deb-src http://ru.archive.ubuntu.com/ubuntu/ hardy-updates multiverse
deb http://security.ubuntu.com/ubuntu hardy-security main restricted
deb-src http://security.ubuntu.com/ubuntu hardy-security main restricted
deb http://security.ubuntu.com/ubuntu hardy-security universe
deb-src http://security.ubuntu.com/ubuntu hardy-security universe
deb http://security.ubuntu.com/ubuntu hardy-security multiverse
deb-src http://security.ubuntu.com/ubuntu hardy-security multiverse
deb http://ftp.debian.org sarge main
```



# apt-get

---

Обновление списка доступных пакетов

```
sudo apt-get update
```

Поиск пакета

```
sudo apt-cache search ssh | grep server
```

Установить/переустановить пакет

```
sudo apt-get [--reinstall] install alien
```

Удаление пакета

```
sudo apt-get [--purge] remove usplash
```

Обновление пакета/дистрибутива

```
sudo apt-get [-u] upgrade [mc]
```

```
sudo apt-get [-u] dist-upgrade
```



# Установка антивируса clamav

---

**clamav** является полностью бесплатным антивирусом, в отличие от многих своих коммерческих собратьев.

Это значит, что вы можете абсолютно бесплатно скачать, установить (на любое количество машин) и использовать данный антивирус.

Обновления тоже бесплатны.

Для установки ClamAV введите команду:

***sudo apt-get install clamav***



# Использование антивируса clamav

---

После установки нужно сразу обновить антивирусные базы:

***sudo freshclam***

Для проверки всей файловой системы следует выполнить команду:

***sudo clamscan -r /home***

Если нужно проверить отдельный каталог, тогда укажите его имя:

***sudo clamscan -r <каталог>***

Можно добавить в файл ***/etc/crontab*** команды для автоматической ежедневной проверки файловой системы и обновления антивирусных баз. Проверку всего компьютера целесообразно делать ночью, чтобы работа антивируса не отображалась на производительности системы.



# dpkg

---

- Преобразование файлов '.rpm' в файлы '.deb'

`sudo alien файл_пакета.rpm`

- Установка пакета

`sudo dpkg -i package_file.deb`

- Удаление пакета

`sudo dpkg -r имя_пакета`



# Установка ПО из исходных кодов

---

- Распространяется в tar-архивах сжатых с помощью gzip или bzip2.
- Сборка осуществляется одним из компиляторов семейства gcc (GNU Compilers Collection).
- Процесс сборки и установки содержит определенную последовательность шагов, которая описана в специальном файле.
- Иногда, процессы сборки/установки инициируется простым shell-скриптом.





# Утилита make

---

- Предназначена для сборки программ из исходных кодов.
- После запуска ищет в текущей директории файл `makefile` или `Makefile`, в котором должны быть описаны действия необходимые для сборки программы.
- Все действия описанные в `makefile` группированы по целям: сборка, установка и т.п.
- Перед сборкой как правило требуется создать `makefile` в соответствии с конфигурацией системы, для чего запускается скрипт `configure`
- Например:  
**`./configure`**  
**`make`**  
**`make install`**



# Пример установки ПО из ИСХОДНЫХ КОДОВ

---

Сборка и установка

```
sudo su -  
apt-get install build-essential  
cp ndis*.tar.gz /tmp  
cd /tmp  
tar zxf ndis*.tar.gz  
cd ndis*  
make && make install
```

Настройка

```
cd /tmp/drivers/winXP/broadcom-4306  
ndiswrapper -i bcmw4306.inf
```



# Установка бинарных файлов из архивов

---

Архивы с бинарными файлами, либо содержат в себе инсталлятор, либо соответствуют структуре каталогов системы в которую устанавливаются, например так:

```
sudo tar xzf squid-bin.tar.gz -C /
```

После распаковки в /opt окажется уже установленная программа, а в /etc/profile.d скрипт настраивающий переменную PATH с учетом установленной программы



# X Window System

---

Графическая оболочка, которая имеет клиент-серверную архитектуру.

Х-сервер – выполняется на локальном компьютере и представляет из себя “холст” на котором Х-клиенты отображают свои данные.

Х-клиенты – используют Х-сервер для отображения информации.

Взаимодействие клиентской и серверной части осуществляется по стандартному протоколу – X Window System Protocol.



# X.org

---

Является бесплатной реализацией системы X Window и полностью с ней совместима. Поставляется со всеми дистрибутивами Linux и BSD-систем.

Способы поставки:

- архив с исходным кодом (обычно tar.gz или tar.bz2)
- pkg-пакеты slackware (в формате tgz)
- rpm-пакеты (бинарные и/или исходный код)
- deb-пакеты (бинарные и/или исходный код)



# Файл конфигурации

---

Полный путь к файлу – `/etc/X11/xorg.conf`

Генерация нового файла конфигурации:

```
sudo su -
```

```
mv /etc/X11/xorg.conf /etc/X11/xorg.old
```

```
rm /tmp/.X*-lock
```

```
X -configure
```

```
cp ~/xorg.conf.new /etc/X11/xorg.conf
```

```
killall Xorg
```

Восстановление исходного файла  
конфигурации:

```
cp /etc/X11/xorg.old /etc/X11/xorg.conf
```



# Секции файла xorg.conf

---

- **Files** – описывает директории, в которых хранятся ресурсы (шрифты, палитры и модули). Если используется сервер шрифтов, то он явно должен быть указан в данной секции.
- **Module** – определяет модули, которые будут использоваться X-сервером. Каталоги, в которых находятся эти модули должны быть указаны в секции “Files”.
- **InputDevice** – определяет устройство ввода, которым может являться клавиатура или мышь. Поэтому в xorg.conf как минимум 2 секции InputDevice.
- **Monitor** – содержит характеристики и настройки монитора.
- **Device** – указывает характеристики чипсета видеосистемы.
- **Screen** – описывает монитор, видеокарту, глубину цвета и доступные разрешения.
- **ServerLayout** – содержит описания X-серверов и ссылки на используемые ими устройства ввода-вывода из-других секций.



# Перезапуск X-сервера

---

- Из консоли X-сервер можно запустить командой `X` или `startx` (рекомендуется).
- Переключение в текстовую консоль: ***Ctrl+Alt+Fn***, где *n* – от 1 до 6, т.к. на 7-й консоли работает сам X-сервер
- Для переключения с текстовой консоли в сессию X-сервера следует использовать комбинацию клавиш: ***Alt+F7***





# Менеджеры дисплеев

---

Традиционно, менеджеры дисплеев выполняют функцию авторизации пользователей.

Существуют следующие менеджеры дисплеев:

- kdm – менеджер дисплеев от разработчиков KDE
- gdm – менеджер дисплеев от команды GNOME
- xdm – классический менеджер дисплеев



# Оконные менеджеры

---

- Продвинутые
  - KDE – K Desktop Environment, основан на библиотеке Qt, является наиболее развитым из всех
  - GNOME – основан на gtk2, является наиболее популярным.
- Легкие
  - WindowMaker
  - FluxBox
  - FwWm
  - .....



# Установка 3D-возможностей

---

- Выберите “Приложения” → “Установка/удаление приложений” .
- В поле “Поиск” введите “compiz”, затем из списка приложений выберите:
  - Advanced Desktop Effects Settings (ccsm)
  - Compiz Fusion Icon
  - Desktop Effects



# Мультимедиа и кодеки

---

По умолчанию в Ubuntu установлены кодеки только для свободных форматов, таких как ogg, это напрямую связано с идеологией дистрибутива — свобода во всём, в том числе и в спецификациях и форматах.

Однако при встрече с незнакомым форматом Ubuntu автоматически предложит подходящий для него кодек, и не удивляйтесь, если это предложение поступит 2 раза — сначала видеоплеер наткнется на неизвестную аудиодорожку, а потом на видео, соответственно для каждой из них и установит кодеки.

Если после установки рекомендованных кодеков не воспроизводится аудио или видео, то попробуйте вручную установить их:

```
sudo aptitude install gstreamer0.10-plugins-ugly
sudo aptitude install gstreamer0.10-plugins-ugly-multiverse
sudo aptitude install gstreamer0.10-plugins-bad
sudo aptitude install gstreamer0.10-plugins-good
sudo aptitude install gstreamer0.10-plugins-bad-multivers
sudo aptitude install gstreamer0.10-ffmpeg
```

В случае возникновения проблем можно перевести движок видеопроигрывателя с gstreamer на xine:

```
sudo aptitude install totem-xine ffmpeg libxine-extracodecs
```



# Резервное копирование и восстановление

---

- Резервное копирование пользовательских данных
  - Поиск файлов в системе
  - Архиваторы
  - Утилиты компрессии данных
- Восстановление резервных копий
- Создание архива с образом системного раздела и его сохранение на сервере
- Восстановление системного раздела



# Программы для поиска файлов в системе

---

- `which` – производит поиск файла в директориях, описанных переменной `PATH`.
- `whereis` – программа ищет в файл в директориях, описанных переменной `PATH` и в `manpages`.
- `locate` – индексирует файловую систему в собственную базу данных, и впоследствии ищет файлы по индексам в БД.
- `find` – осуществляет рекурсивный поиск файлов в файловой системе, не использует базы данных и переменные окружения.



# which

---

which – производит поиск файла в директориях, описанных переменной PATH.

Опции:

-a – показать все найденные файлы

Примеры:

which pppd

which ls



# whereis

---

whereis – программа ищет в файл в директориях, описанных переменной PATH и в manpages.

Опции:

- b – искать только в директориях, описанных в переменной PATH
- m – искать только в manpages

Примеры:

whereis dd

whereis -b dd

whereis -m dd





# locate

---

updatedb – индексирует файловую систему в собственную базу данных и locate впоследствии ищет файлы по индексам в БД.

Пример:

```
sudo updatedb
```

```
locate ls | grep ls$
```



# find

- **find** – осуществляет рекурсивный поиск файлов в файловой системе, не использует базы данных и переменные окружения.
- Условия поиска:
  - **-mount** или **-xdev** – осуществлять поиск только в пределах одной физической файловой системы
  - **-name шаблон** – поиск файла по его имени
  - **-iname шаблон** – то же, но без учета регистра
  - **-regex шаблон** – то же, что и name, но шаблон – регулярное выражение
  - **-type тип\_файла** – поиск файлов указанного типа
  - **-user пользователь** – искать файлы, принадлежащие пользователю
  - **-group группа** – искать файлы, принадлежащие группе
  - **-atime N** – искать файлы, доступ к которым был N суток назад
  - **-mtime N** – искать файлы, которые менялись N суток назад
  - **-size N** – искать файлы, размер которых N блоков
- Команды:
  - **-exec программа** – выполнить указанную программу передав ей имя файла
  - **-ok программа** – то же, что exec, но с запросом подтверждения для каждого файла
- Пример:

```
find /home -user user1 -exec ls -l {} \;  
find /usr -name *.gif -ok lpr -P hp {} \;
```



# Архиватор tar

---

Программа предназначена для работы с архивами в формате tar.

Опции:

- f имя\_файла*** – определяет имя архива.
- v*** – вывод дополнительной информации
- c*** – создание архива
- x*** – распаковка архива
- t*** – просмотр содержимого архива

Примеры:

```
tar cvf archive.tar .bash_* .mc .ssh  
tar -xvf archive.tar
```



# Архиватор cpio

---

Программа предназначена для работы с архивами в формате cpio.

Опции:

- p** – режим копирования файлов.
- v** – вывод дополнительной информации
- o** – создание архива
- i** – распаковка архива
- t** – просмотр содержимого архива
- d** – создание необходимых директорий

Примеры:

```
find /usr -name *.gif | cpio -o > gifs.cpio  
cpio -id < gifs.cpio  
find /usr -name *.gif | cpio -pd gifs
```



# Утилиты компрессии данных

---

***compress [-c] [-d]***

***uncompress*** – идентично ***compress -d***

***gzip [-c] [-d]***

***gunzip*** – идентично ***gzip -d***

***bzip2 [-c] [-d]***

***bunzip2*** – идентично ***bzip2 -d***

Примеры:

```
compress test.tar
```

```
uncompress test.tar.Z
```

```
gzip -c test.tar > test.tar.gz
```

```
gunzip test.tar.gz
```

```
bzip2 -c test.tar > test.tar.bz2
```

```
bunzip2 test.tar.bz2
```



# Использование программ компрессии в tar

---

Опции ***tar*** для вызова программ  
компрессии:

**-Z** – вызов программы ***compress***

**-z** – вызов программы ***gzip***

**-j** – вызов программы ***bzip2***

Примеры:

***tar -czvf file.tar.gz file1 file2 ...*** - создание архива

***tar -zxvf file.tar.gz*** – извлечение данных



# 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 /
```



# Резервное копирование и восстановление системного раздела

---

```
sudo su -  
mkdir /mnt/zip  
mount -t cifs -o username=user1 //server/public /mnt/zip  
telinit 1  
mount -o ro,remount /  
cd /  
dd if=/dev/sda2 | bzip2 -q9c > /mnt/zip/system.bz2  
bzip2 -dc /mnt/zip/system.bz2 | dd of=/dev/sda2
```





# Виртуализация

---

- **Виртуализация** — это общий термин, охватывающий абстракцию ресурсов для многих аспектов вычислений.
- Некоторые наиболее характерные способы виртуализации приведены ниже:
  - **Виртуальная машина** — это программно-аппаратное окружение, которое представляется для «гостевой» операционной системы, как аппаратное.
  - **Виртуализация на уровне операционной системы** — виртуализирует физический сервер на уровне ОС, позволяя запускать изолированные и безопасные виртуальные серверы на одном физическом сервере.
  - **Виртуализация ресурсов** (partitioning) — это разделение единого, обычно достаточно большого для этого, ресурса (например дисковое пространство или пропускной канал сети) на некоторое количество меньших, легче используемых ресурсов того же типа.
  - **Виртуализация прикладных приложений** — включает в себя рабочую среду для локально выполняемого приложения, использующего локальные ресурсы. Такая виртуальная среда работает как прослойка между приложением и операционной системой, что позволяет избежать конфликтов между приложениями.



# Виртуальная машина

---

- Виртуальная машина эмулирует работу реального компьютера.
- На виртуальную машину, так же как и на реальный компьютер можно установить операционную систему.
- У виртуальной машины так же есть BIOS, оперативная память, жёсткий диск (выделенное место на жёстком диске реального компьютера), могут эмулироваться периферийные устройства.
- На одном компьютере может функционировать несколько виртуальных машин.
- Примеры виртуальных машин:
  - Vmware
  - VirtualBox
  - Qemu



# Установка виртуальных машин

---

Все выполняется суперпользователем

```
sudo su -
```

Установка Virtualbox пакетов

```
apt-get install virtualbox-ose
```

```
apt-get install virtualbox-ose-modules-$(uname -r)
```

```
adduser имя_учетной_записи vboxusers
```

**Повторно войдите в систему**



# Сборка и установка ядра

---

В каких случаях требуется сборка ядра:

- Текущая версия ядра не поддерживает ваше оборудование
- Установка патчей, устраняющих критические уязвимости ядра
- Повышение безопасности ядра путем устранения неиспользуемого кода
- Оптимизация производительности системы (можно выиграть от 5% до 15%)
- Необходимо сократить объем памяти занимаемой ядром (характерно для встраиваемых устройств).



# Сборка нового ядра

---

```
sudo su -
```

```
apt-get install linux-source-2.6.XX kernel-package
```

```
apt-get install libncurses*
```

```
cd /usr/src
```

```
tar jxf linux-source-2.6.XX.tar.bz2
```

```
cd linux-source-2.6.XX
```

```
make menuconfig
```

```
make-kpkg clean
```

```
make-kpkg --initrd --revision=mybuild1 kernel_image kernel_headers
```



# Установка нового ядра

---

Все команды должны выполняться суперпользователем:

```
sudo su -
```

```
cd /usr/src
```

```
dpkg -i *.deb
```

Ваше ядро (файл `vmlinuz-2.6.XX-YYY`) будет помещено в каталог `/boot` (все предыдущие ядра тоже никуда не денутся, останутся на своих местах), а в каталоге `/lib/modules`, рядом с каталогом с модулями обычного ядра появится каталог с модулями вашего нового ядра.

Новое ядро будет автоматически прописано в `/boot/grub/menu.lst`.

В принципе, уже можно перегрузиться, и в экране загрузки Grub появится новый пункт с вашим ядром. Новое ядро появится в начале списка.



# Система печати

---

Существуют два основных типа систем печати:

- System V
- BSD

Наибольшую популярность в Linux получила система печати CUPS.

CUPS – современная система печати, поддерживающая следующие протоколы сетевой печати:

- bsd (515 порт)
- ipp (631 порт)
- smb (требуется Samba).

Совместима с классическими системами печати BSD и System V



# Система печати CUPS

---

Демон `cupsd` запускается при старте системы и открывает на прослушивание 631 порт.

Для эмуляции BSD системы печати требуется запуск демона `cups-lpd`.

Для полноценной поддержки принтеров необходимо наличие PPD-файлов, описывающих эти принтеры.

Такие файлы поставляются как с CUPS, так и в виде отдельных пакетов.

Если в системе нет PPD-файла принтера, его можно найти либо на диске с драйверами к принтеру, либо на сайте [www.linuxprinting.org](http://www.linuxprinting.org)

Установить принтер и управлять им можно из командной строки и при помощи Web-интерфейса ( <http://localhost:631> )





# Настройка cups с командной строки

---

Для добавления принтера в командной строке следует пользоваться программой `lpadmin`

- Нужно указать `ppd`-файл, включая путь к нему
- Так же потребуется указать устройство принтера, полное название которого можно посмотреть запустив `lprinfo -v`

Пример добавления локального принтера:

```
lpadmin -p Laser -E -v usb:/dev/usb/lp0 \  
-m foomatic-ppds/HP/HP-Laserjet_1300-hpijs.ppd.gz
```

Пример добавления удаленного принтера\*:

```
lpadmin -p Laser -E -v http://IP:631/printers/Printer
```

\*На сервере необходимо, чтобы стояли разрешения на печать в `/etc/cups/cupsd.conf`



# Печать с консоли

---

***lpr*** – утилита для помещения задания в очередь печати

***lpq*** – утилита отображает состояние очереди печати

***lprm*** – удаляет задание из очереди печати

Опция (общая для всех этих утилит):

***-P Printer***

Пример:

```
find /usr -name *.gif -exec lpr -P PDF {} \;
```

```
lpq
```

```
lprm 10
```



## Подключение дополнительного раздела жесткого диска

---

Создайте новый раздел

```
sudo su -
```

```
LANG=en_US.UTF-8 cfdisk
```

Создайте на нем файловую систему

```
mkfs -t ext3 /dev/sdaX
```

Отредактируйте /etc/fstab и добавьте в него строку

```
nano /etc/fstab
```

```
/dev/sdaX /mnt/disk ext3 defaults 0 1
```

Смонтируйте раздел

```
mount /dev/sdaX
```

**Спасибо за внимание!**

