

GNU/Linux

Файловая система.

Команды.

Переменные окружения.

(Практика)

Задание на разминку:

На сколько 2^{100} больше 2^{99} ?

Работа в командной оболочке

- В ОС Ubuntu работать в ком. строке можно:
 - через консоль;
 - через графический терминал;
- Для работы через консоль используем сочетание клавиш **Ctrl+Alt+F1**, для возврата в графический интерфейс — сочетание клавиш **Ctrl+Alt+F7**.
- Для запуска Терминала нажмите сочетание клавиш **Ctrl+Alt+T**.
- Внизу экрана располагается приглашение командной строки — последовательность символов, извещающая о готовности к обработке команд:

kamilla@kamilla-Aspire-V3-371:~\$

kamilla — имя текущего пользователя
(суперпользователь/администратор);

kamilla-Aspire-V3-37 — имя хоста (сервер);

: — разделитель;

~ — текущая директория (в какой папке выполняется команда)
пользователя (по умолчанию /home/<username>);

\$ — символ приглашения ввода для пользователей с обычными правами;

— символ приглашения ввода для суперпользователя/администратора root (например, root@cs49647:~#).



```
kamilla@kamilla-Aspire-V3-371: ~  
To run a command as administrator (user "root"), use "sudo <command>".  
See "man sudo_root" for details.  
  
kamilla@kamilla-Aspire-V3-371:~$
```

Управление файлами и каталогами

- **rm** – (remove) – удаление файла:

`$ rm myfile.txt`

- **mkdir** -(makeDirect) – создание директории(каталога):

`$ mkdir mydir/`

- **touch** - создание файла:

`$ touch myfile.txt`

- **cd** – (changeDir) – смена директории,

`$ cd /home/dir1/dir2`

- **ls** - (list) – отображение директории или файла:

`$ ls /home`

- **cp** – (copy) – копирование файла: *cp <что копируем> <куда копируем>*

`$ cp file1.txt /home/lena/dir01/dir11`

Установка и обновление пакетов

В каждом дистрибутиве есть свой пакетный менеджер, который может запрашивать и устанавливать пакеты из удаленных репозиториев. Это очень удобно — требуется знать только название пакета, который желаете установить и команду. Пакетный менеджер по умолчанию в Debian и дистрибутивах на основе Debian — **apt** (Advanced Packaging Tool).

- Для обновления пакетов введем команду:

\$ *sudo apt update*

- Для установки пакета с помощью **apt** необходимо ввести команду:

\$ *sudo apt install <имя_пакета>*

Можно встретить **apt-get** вместо **apt** — это тот же самый пакетный менеджер, только более старая его версия. Тем не менее оба варианта существуют и работают. Пакетный менеджер **apt** объединяет сразу несколько прежних команд (**apt-get** и **apt-cache** в **apt install** и **apt search**), а также оптимизирует процесс установки пакетов.

- Команда для удаления установленного пакета с помощью **apt**:

\$ *sudo apt remove <имя_пакета>*

- Для удаления зависимостей вместо **remove** используется **purge**:

\$ *apt purge <имя_пакета>*

- Для автоматического удаления всего лишнего из системы используется:

\$ *apt autoremove*

Автоматическое удаление лишнего — это как встроенный в систему менеджер очистки, команда удаляет все файлы и зависимости, которые больше не требуются в системе, например, потому что соответствующего им приложения больше нет в системе.

Узнать тип файловой системы.

- Способ №1: Использование команды **df**. Команда **df** отображает информацию об использовании дискового пространства файловой системы. Для указания того, что нам нужно вывести тип файловой системы, используйте следующую команду:

```
$ df -Th
```

```
$ df -Th | grep "^/dev"
```

- Способ №2: Использование команды **fsck**. Команда **fsck** применяется для проверки и, при необходимости, восстановления файловых систем Linux. При этом она также может отображать и тип файловой системы на указанных разделах диска, например:

```
$ fsck -N /dev/sda1
```

- Способ №3: Использование команды **lsblk**. Команда **lsblk** отображает информацию о блочных устройствах. Добавив опцию **-f**, мы также получим и информацию о типе файловой системе:

```
$ lsblk -f
```

- Способ №4: Использование команды **mount**. Команда **mount** применяется для монтирования файловой системы в Linux. Её также можно использовать для монтирования ISO-образа, удаленной файловой системы Linux и многого другого. Чтобы узнать тип файловой системы, используйте следующую комбинацию:

```
$ mount | grep "^/dev"
```

```
~: bash — Konsole
Файл  Правка  Вид  Закладки  Настройка  Справка

diego@debian:~$ df -Th
Файловая система Тип      Размер  Использовано  Дост  Использовано%  Смонтировано в
udev             devtmpfs 1,2G      0             1,2G      0%             /dev
tmpfs             tmpfs     250M      1,1M          249M      1%             /run
/dev/sda1         ext4       78G      23G           52G      31%            /
tmpfs             tmpfs     1,3G      0             1,3G      0%             /dev/shm
tmpfs             tmpfs     5,0M      0             5,0M      0%             /run/lock
tmpfs             tmpfs     250M      88K           250M      1%             /run/user/1000
diego@debian:~$ df -Th | grep "^/dev"
/dev/sda1         ext4       78G      23G           52G      31%            /
diego@debian:~$
```

```
~: bash — Konsole
Файл  Правка  Вид  Закладки  Настройка  Справка

root@debian:~# fsck -N /dev/sda1
fsck from util-linux 2.36.1
[/usr/sbin/fsck.ext4 (1) -- /] fsck.ext4 /dev/sda1
root@debian:~#
```

```
~: bash — Konsole
Файл  Правка  Вид  Закладки  Настройка  Справка

diego@debian:~$ lsblk -f
NAME FSTYPE FSVER LABEL        UUID                                 FSAVAIL FSUSE% MOUNTPOINT
sda
├─sda1
│   ext4   1.0      5a943b64-438b-4c78-a4e4-23a9707dde07  51G     29%    /
└─sda2

sr0   iso9660 Jolie Debian 10.8.0 amd64 n
      2021-02-06-10-36-59-00
diego@debian:~$
```

```
~: bash — Konsole
Файл  Правка  Вид  Закладки  Настройка  Справка

diego@debian:~$ mount | grep "^/dev"
/dev/sda1 on / type ext4 (rw,relatime,errors=remount-ro)
diego@debian:~$
```

Узнать тип файла

Для определения типа файла достаточно воспользоваться всего одной командой:

```
$ ls -ld <file_name>
```

Например:

```
$ ls -ld /etc/services
```

```
-rw-r--r-- 1 root root 19281 Feb 14 2012 /etc/services
```

Команда **ls** отобразит тип файла в виде первого символа из списка. В данном случае этим символом будет **-**, обозначающий «обычный файл». Не следует путать типы файлов с расширениями файлов.

- **Каталог:**— это второй по распространенности тип файлов в Linux. Каталог можно создать с помощью команды **mkdir** (сокр. от «make directory»):

```
$ mkdir ravesli
```

```
$ ls -ld ravesli
```

```
drwxr-xr-x 2 diego diego 4096 фев 20 12:14 ravesli
```

d (от «directory») выходных данных команды **ls**. Для удаления пустого каталога используется команда **rmdir** (сокр. от «remove directory»):

```
$ rmdir ravesli
```

```
$ ls -ld ravesli
```

ls: невозможно получить доступ к 'ravesli': Нет такого файла или каталога

При попытке удалить каталог, содержащий внутри себя файлы, вы получите сообщение об ошибке:

```
$ mkdir ravesli
```

```
$ touch /home/diego/ravesli/ravesli.txt
```

```
$ rmdir ravesli
```

```
rmdir: не удалось удалить 'ravesli':
```

Каталог не пуст В этом случае вам нужно использовать команду **rm -r**:

```
$ rm -r ravesli/
```

```
$ ls -ld ravesli
```

ls: невозможно получить доступ к 'ravesli': Нет такого файла или каталога

- **Обычный файл:** наиболее распространенный тип файлов в Linux. Текстовые файлы, изображения, двоичные файлы, общие библиотеки и т.д. — все они относятся к обычному типу файлов. Обычный файл создается с помощью команды

```
$ touch ravesli.txt
```

```
$ ls -ld ravesli.txt
```

```
-rw-r--r-- 1 diego diego 0 фев 20 11:14 ravesli.txt
```

Первый символ вывода команды **ls**, в данном случае **-**, обозначает, что перед нами обычный файл. Для удаления обычного файла используется команда **rm** (сокр. от «remove»):

```
$ rm ravesli.txt
```

```
$ ls -ld ravesli.txt
```

ls: невозможно получить доступ к 'ravesli.txt': Нет такого файла или каталога

- **Символьные ссылки:**

Обозначаются как **l**.

С помощью символьных ссылок администратор назначает файлу или каталогу несколько идентификаторов. Символьную ссылку можно рассматривать как указатель на исходный файл.

Существует два типа символьных ссылок в Linux:

- Мягкая ссылка является указателем на некоторый файл или каталог (сродни ярлыкам в Windows). Если вы переместите файл, связь с символьной ссылкой разорвется (но сама ссылка все еще будет существовать, указывая на файл, которого нет). Если вы замените файл другим, сохранив имя, символьная ссылка будет указывать на новый файл. Символьные ссылки могут охватывать различные разделы файловой системы.
- Жесткая ссылка привязывается только к файлам, охватывает только один раздел файловой системы и, по сути, является тем же файлом, на который ссылается.

Для создания символьной мягкой ссылки используется команда **ln -s**:

```
$ echo file1 > file1
```

```
$ ln -s file1 file2
```

```
$ cat file2 file1
```

```
$ ls -ld file2
```

```
lrwxrwxrwx 1 diego diego 5 фев 20 14:11 file2 -> file1
```

Чтобы удалить символьную ссылку, мы можем использовать команду **unlink** или **rm**.

«Everything is a File»

- Используем команду **cat**, чтобы посмотреть содержимое `/etc`:

```
$ cat /etc
```

```
$ sudo cat /usr/lib/systemd/systemd-ac-power
```

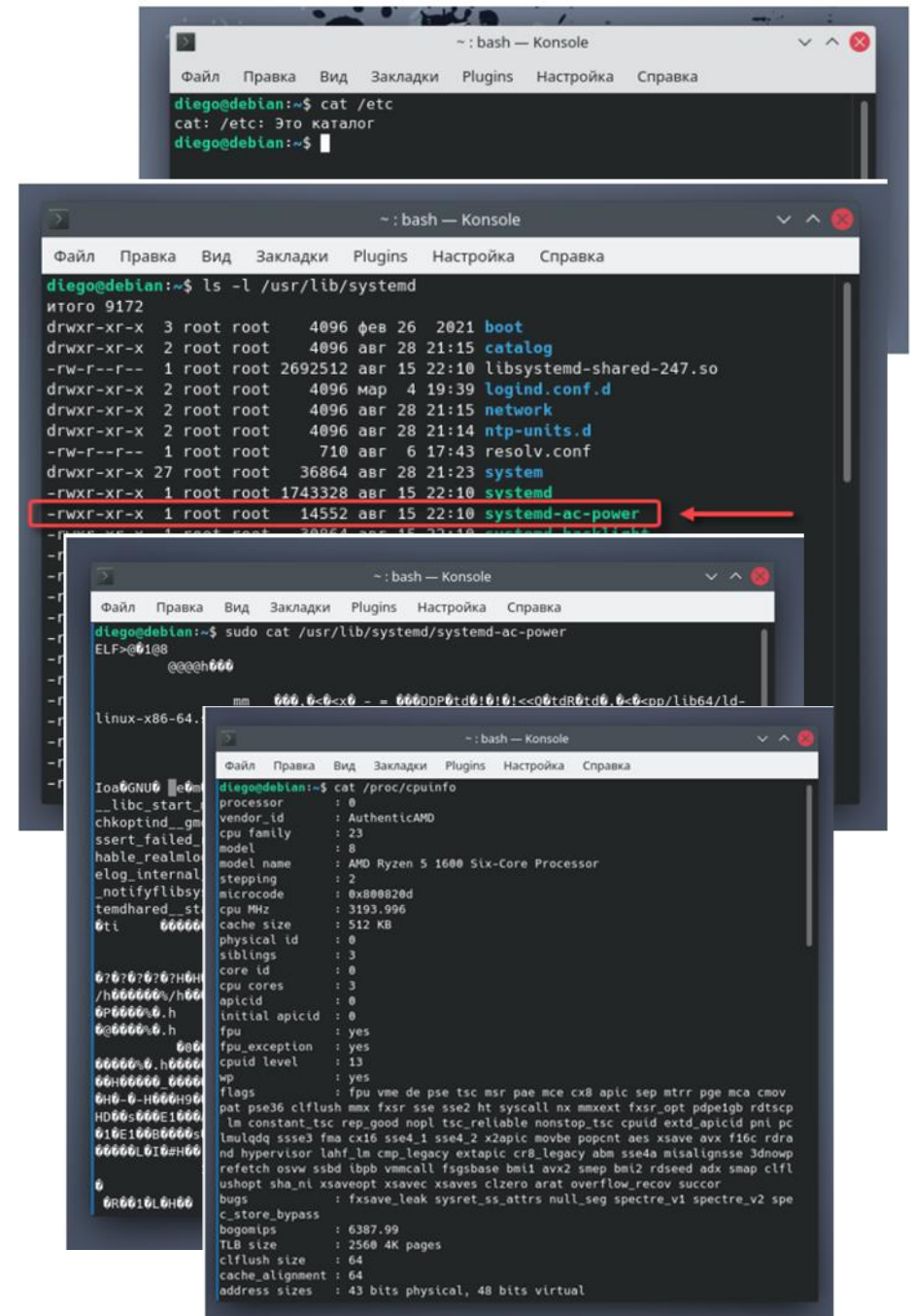
«`systemd-ac-power`» является бинарным файлом. Открыть можем, посмотреть — нет.

- Найдем информацию о процессоре. Каталог `/proc` имеет специальный файл — `/proc/cpuinfo`, который содержит данную информацию. Мы можем прочитать содержимое этого файла следующей командой:

```
$ cat /proc/cpuinfo
```

Также в каталоге `/proc` содержатся и другие файлы, например:

- `/proc/uptime` — информация о времени безотказной работы (uptime) ядра Linux. Т.е. как долго наша система работала без выключения.
- `/proc/version` — информация о версии ядра Linux.
- `/proc/meminfo` — информация о свободной и используемой памяти (как физической, так и виртуальной)



Изменение доступа к файлу



- Доступные операции с файлом:
чтение (r), запись (w) или выполнение (x)
- Иерархия доступа к файлу на уровне пользователей:
владелец (u), группа (g), другие (o), все пользователи (a)
- Изменение доступа к файлу:
 - `$ chmod <права_доступа> <имя_файла/папки>`

Формирование прав доступа:

{Владелец может: писать, читать, исполнять файл} : **u=rwx**

{Все могут: писать, читать файл} : **a=rw**

{Забрать у остальных: писать, файл} : **o-w**

Переменные окружения

- Получение информации о переменных окружения:

\$ printenv

- Вывод значения конкретной переменной:

\$ echo <имя_перем>

- Установка переменной и ее значения:

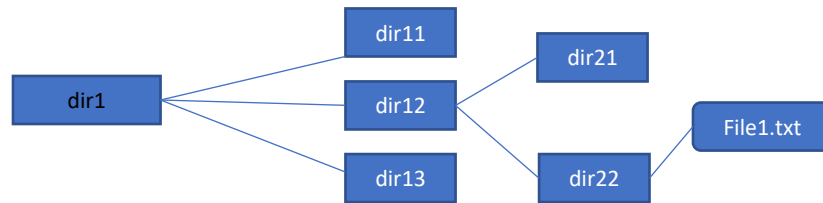
\$ ИМЯ='значение'

- Удаление переменной:

\$ unset <имя_перем>

Домашнее задание

- Изучить работу команд: `rmdir`, `mv`, `pwd`, `cat`, `find`, `shutdown`, `reboot`, `ip`, `ps`, `free`, `useradd`, `userdel`, `passwd`, `usermod`.
- Изучить восьмеричный формат для изменения прав доступа к файлу и как его применять.
- Создать структуру каталогов и файлов. (*Продвинутые пробуют писать скрипт.)



- Создать 30 директорий и поместить в каждую по 20 файлов.
- Создать пользовательские переменные окружения и установить им значения:

DIR_TEST='/home/test'

MY_BIRTHDAY=20-05-1995