

# Linux and more.

## Files, pipes

Artem Trunov for Ozon Masters



# Запуск команд

- Запуск команд - в чем разница? `ls -al`
  - `/tmp/hello_masters`
  - `Hello_masters`
- Почему в одном случае запускаем с полным путём?

# Запуск команд

- `echo $PATH` #Вывод переменной
- `which ls` #Показать полный путь к данной команде
- Команда `ls` находится в одной из директорий из списка в переменной среды `$PATH`
- *Если исполняемая программа не в папке из \$PATH, требуется полный или относительный путь*

```
ubuntu@linux1:~$ which ls
/usr/bin/ls
ubuntu@linux1:~$ echo $PATH
/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
:/usr/games:/usr/local/games
```

# Запуск программ с интерпретатором

- Запуск программ с интерпретатором
  - `cd /tmp`
  - `python3 hello.py`
  - `/usr/bin/python3 hello.py`
- Сделайте первой строчкой: `#!/usr/bin/python3`
  - `cd /tmp`
  - `./hello1.py`
  - (Не забывайте, когда вы сами создаете: `chmod +x hello1.py`)
  - А также: `#!/usr/bin/env python3`

# Немного о структуре директорий

**/bin** - утилиты общего пользования

**/sbin** - утилиты для администратора

**/boot** - ядро и загрузочный образ системы

**/dev** - псевдо фс для устройств (диски, звук, и.т.д.)

**/etc** - конфигурация приложений

**/home** - домашние директории пользователей

**/lib**, **/lib64** - разделяемые библиотеки

**/proc** - псевдо фс для интерфейса к процессам и ядру

**/root** - домашняя директория пользователя root

**/run** - где приложения могут хранить нужные им файлы (pid-file)

**/sys** - псевдо фс, интерфейс к ядру и модулям

**/tmp** - пространство для временных файлов (очищается при перезагрузке)

**/usr** - для установленных для пользователей программ (не системные)

**/var** - для записи установленными для пользователей программ (логи етс)

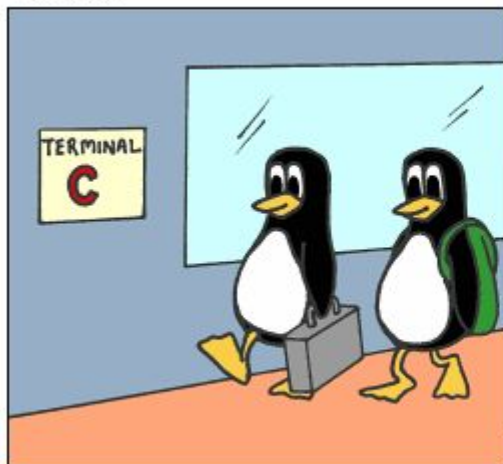
# Работа с файлами

## Просмотр файлов

- **cat** - просто вывести содержимое
- **head** - вывести начало файла
  - `head -5`
- **tail** - вывести конец файла
  - `tail -f`
- **less** - интерактивный просмотрщик (выход - q)
  - пробел, b - страницу вперед, назад
  - стрелки - на строку вперед, назад
  - `/что-то` - найти что-то (регулярное выражение) - поиск вперед
  - `?что-то` - поиск назад

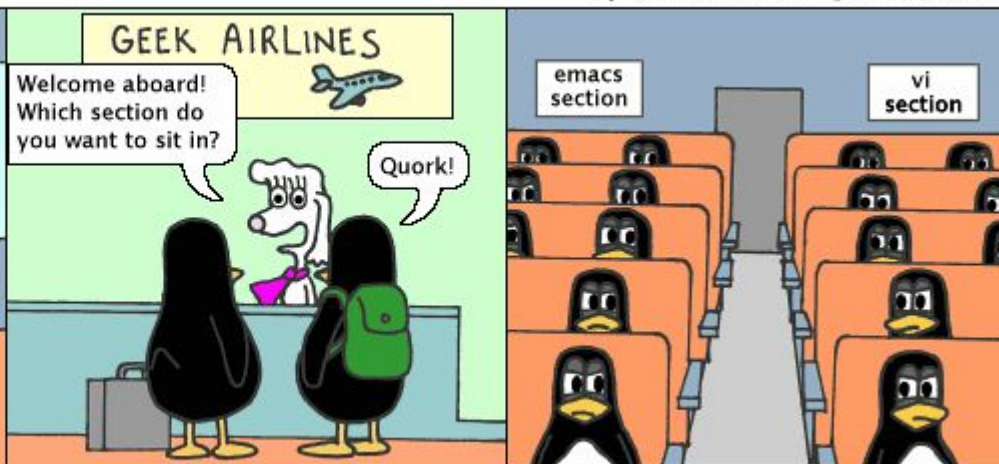
# Работа с файлами. Редактирование

Hackles



<http://hackles.org>

By Drake Emko & Jen Brodzik



Copyright © 2003 Drake Emko & Jen Brodzik

# Vi vs. emacs vs. ...

- Это просто инструменты для редактирования.
- Используйте то, что удобнее для вас лично.
- То, что помогает в вашей работе.



# Vi - краткая инструкция

- Esc а или i - начать ввод текста. Стрелки, del, ins и т п работают как надо
- Esc : w - сохранение
- Esc : x - выход с сохранением
- Esc : q - выход (q! - без сохранения)
- Esc dd - удалить строку
- Esc u - undo
- Esc yy - скопировать строку в буфер
- Esc p - скопировать из буфера после текущей строки
- Esc : %s/replace\_pattern/replace\_with/g - замена
-

# Emacs тоже установил

- Запуск: emacs
- Выход Ctrl-x Ctrl-c

# Другие инструменты

- **nano** - хорошая простая альтернатива “титанам”
  - Да еще с подсказками команд.
  - Он же pico
- Jupyter
  - Режим редактирования файлов
- IDE
  - Pycharm
  - VSCode

# Работа с файлами. Поиск в файлах

- **wc** - word count
  - `wc -l`
- **grep** - поиск подстрок и шаблонов в тексте
- **awk** - манипуляции с содержимым
- **sort** - сортировка строк
- ...

# grep

- `wget -O Alice.txt http://www.gutenberg.org/files/11/11-0.txt`
- `grep Alice Alice.txt`
- `grep -c Alice Alice.txt`
- `grep -o Alice Alice.txt`
- `grep -Eo "\w+d Alice" Alice.txt`
  - -E - extended regexp, -F, -P
- `grep -Eo -A 1 "\w+d Alice" Alice.txt`
  - -B, -C
- `grep -i Alice Alice.txt`
- `grep -v Alice Alice.txt`
- `grep Alice *`
- `grep -r Alice *`

# Лирическое отступление - Философия Unix

- Write programs that do one thing and do it well.
- Write programs to work together.
- Write programs to handle text streams, because that is a universal interface.



# Как программы работают вместе? Pipes

- Выход одной программы может подаваться на вход другой программы
- `cat Alice.txt | wc`
- `grep Alice Alice.txt | wc`
- `cat Alice.txt | grep Alice | wc`
- `cat Alice.txt | grep -Eo "\w+d Alice" | grep -iv and | sort | uniq -c | sort -nr`

# Другие команды-фильтры

- awk
- sed
- tr
- comm
- join

man coreutils



# Потоки ввода-вывода. Перенаправление.

- У каждой программы есть
  - Stdin - поток ввода (0)
  - Stdout - поток стандартного вывода (1)
  - Stderr - поток ошибок. (2)
- Их можно перенаправлять
- `grep AAA /etc/ssh/ssh_host*`
- `grep AAA /etc/ssh/ssh_host* | wc`
- `grep AAA /etc/ssh/ssh_host* 2>/dev/null | wc`
- `grep AAA /etc/ssh/ssh_host* 2>&1 | wc`

# Потоки ввода-вывода. Перенаправление.

- `grep AAA /etc/ssh/ssh_host* > pub_keys`
- `grep AAA /etc/ssh/ssh_host* > pub_keys 2>errors`
- `grep AAA /etc/ssh/ssh_host* &>out`
- `cat ~/.ssh/id_rsa.pub >> pub_keys 2>>errors #или &>>out`
- `cat ~/.ssh/id_rsa.pub | tee -a pub_keys`
- `cat >> from_keyboard.txt`
  - a
  - b
  - c
  - ctrl-d*
-

# Запуск команд в фоновом режиме

`command >out 2>err &`

*Запускаем интерактивную команду фоном*

`command`

*Ctrl-z*

`bg`

*Возвращаем в интерактив*

`fg`

*Мониторим лог-файл*

`tail -f out`