

# Работа в терминале

## лекция 1

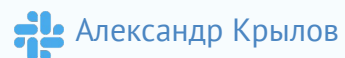


Александр  
Крылов



## **Александр Крылов**

Lead DevOps services в ПАО СК Росгосстрах



---

# План модуля

1. Работа в терминале, лекция 1
2. Работа в терминале, лекция 2
3. Операционные системы, лекция 1
4. Операционные системы, лекция 2
5. Файловые системы
6. Компьютерные сети, лекция 1
7. Компьютерные сети, лекция 2
8. Компьютерные сети, лекция 3
9. Элементы безопасности информационных систем

---

# Классическая лекция 101 – история \*nix, наша первая лекция – основы работы в терминале


Участники курса:

- пользователи Windows, даже в современном Linux многие остаются в GUI
- неопытным пользователям будет трудно воспринять дальнейшие лекции без основ



# План занятия

1. [Цели и план модуля "Основы системного администрирования"](#)
2. [Немного о структуре директорий Linux](#)
3. [Первое открытие терминала](#)
4. [Встроенные функции, ключевые слова и программы](#)
5. [Документация, как искать сведения о командах](#)
6. [Управление задачами](#)
7. [Изучайте инструменты, которые вы используете](#)
8. [Итоги](#)
9. [Домашнее задание](#)



# **Цели и план модуля “Основы системного администрирования”**

---

## Общие цели модуля

- Научиться работать в командной строке Linux – на серверах графической оболочки не будет :)
- Понять как подходить к возникающим проблемам.
- Углубить знания о ресурсах операционной системы Linux и оценке их потребления.
- Узнать, понимать сетевые протоколы разных уровней, получить навыки взаимодействия с ними на уровне администратора.


# Для чего основы администрирования полезны DevOps в теории...

- **Подготовиться к самостоятельному развертыванию сервисов при необходимости.** Нередко функциональности облачных managed-сервисов не хватает, или из-за особенностей ПО их нельзя использовать в нужном режиме. Пример: в GCP до сих пор нет MySQL 8, Huawei Cloud застрял на Kubernetes годовалой давности без cri-containerd (недоступен gVisor и т.п.)
- **Научиться экономить денежные средства при самостоятельном развертывании по сравнению с managed-сервисами.** Часто с разумными затратами усилий можно сделать инфраструктуру существенно дешевле и не хуже (а может быть – даже и лучше, см. SLA на переключение в отказоустойчивых [Google Cloud SQL](#) HA инстансах), чем предлагают облачные провайдеры.
- **Научиться поддерживать самостоятельно развернутые сервисы.** Очень часть DevOps – про облака. Однако много компаний используют собственное железо или имеют гибридную инфраструктуру, а значит спрятаться только лишь за абстракции terraform'a не получится.



## ... и на практике

- **Научиться разбираться с ПО, которое работает не так, как заявлено в документации.**
- **Научиться управляться со внутренними разработками.** Авторы давно уволились, времени/знаний разбираться в исходниках нет, надо срочно починить, на stackoverflow ответов по понятным причинам нет.
- **Научиться строить системы с пониманием компромиссов.** Часто необходимо создать оптимизированную по заданным параметрам архитектуру: отказоустойчивость/стоимость/производительность. Не бывает все и сразу.
- **Научиться оценивать производительность и влиять на нее с пониманием.** Разобраться в недостаточной производительности в продакшене того, что работает нормально в других средах, а не просто “увеличить инстанс в Амазоне”.



# Немного о структуре директорий Linux

---

# FHS

**FHS** (filesystem hierarchy standard). Некоторые характеристики:

- единый корень всей файловой системы , ‘/’, forward slash
- все директории в дереве внутри относительно корня
- если вы из мира Windows – большое отличие в том, что разделы также находятся внутри относительно единого корня
- стандарт Filesystem Hierarchy Standard
  - /bin, /sbin, /etc...
- многие директория называются понятно:
  - /home для домашних каталогов пользователей
  - /root для домашнего каталога аккаунта администратора
  - /tmp для временных, temporary, файлов



# Первое открытие терминала

---

# Что мы видим при открытии терминала?

Один из командных интерпретаторов:

- sh, bash, zsh, fish, и т.д.

Стартует сессия, элементы которой мы сразу наблюдаем:

- имя пользователя
- имя хоста
- тильда – сокращенное представление домашнего каталога, home directory
- приглашение ко вводу

Как еще посмотреть имя пользователя?

`whoami`

# В каком формате shell ожидает ввод?

Базовая структура интерактивного взаимодействия – команда, аргументы этой команды. Начинаем знакомиться с некоторыми командами и действиями в терминале Linux:

- `echo Hello world`
- `echo $USER`
- `$SHELL --version`
- `new_variable=new_text`
- `read`
- `history`
- `clear, reset (aafire)`
- управление с клавиатуры: Ctrl + L, Up/Down, Ctrl + R
- автодополнение по Tab: bash autocompletion (`\``)
- `#` и пробел
- `env` – переменные окружения (environment)

---

# Команды просмотра директорий и переходов между ними

- `pwd` – текущая директория (present working directory)
- `cd` – изменить директорию (change directory)
  - `./ ..`
  - `-`
- `ls` – показать содержимое (list)
  - `-l`
  - `-a`
  - `-la`, сочетание опций
- скрытые объекты с точкой

# alias

Удобный инструмент для сокращения длины вводимых команд – alias.

- `ll`, алиас для `ls -aLF`
- `alias my_new_shiny_alis="ll -t"`

Посмотреть список всех присутствующих alias,

- `alias` без параметров

Не только сокращения: один из способов подмены команд (`alias docker='podman'`)

Как сделать изменения алиасов постоянными (персистентными)?

- `~/.bashrc`
- редактирование файлов в `nano`
- символ `^` как `Ctrl`



# Команды для просмотра файлов

... и поиска содержимого в них.

Простейший просмотр содержимого:

- `cat` – conCATenate
- `head / tail` – начало и конец файла
  - `-n`
  - `<число>, -/+<число>`

Просмотр содержимого в режиме пейджера (постранично):

- `less` vs. `more`
  - `-i`
  - `-N`

Некоторые ключи `grep` для фильтрации (поиска) содержимого файлов:

- `-c`
- `-i`
- `-v`
- `-f (diff, diff -u + sha256sum)`

---

# Команды для создания, перемещения и удаления директорий и файлов

- `mkdir` – make directory
  - `-p`
- `touch`
- `cp` – copy
  - `-v`
  - `-r`
  - `*(strace git add *)`
- `mv` – move
- `rm` – remove
  - `-d`
  - `-r`



# **Встроенные функции, ключевые слова и программы**

# Почему не просто команды

- **type echo:** echo is a shell *builtin*
- **type whoami:** whoami is */usr/bin/whoami*
- **type {:** { is a shell *keyword*

type type :)


type -a pwd

Разные shell – разные команды.

Как shell понимает, что вы от него можете хотеть? Что за **/usr/bin/whoami**?

- echo \$PATH; **export \$PATH** (\$GREP\_OPTIONS)
- which – порядок в \$PATH
- env pwd
- compgen -k, compgen -b

Документация по bash – 4000 строк по состоянию на апрель 2020 (огромное количество встроенных ключевых слов и возможностей, которые стоит изучать).



# **Документация, как искать сведения о командах**

# Как искать сведения о командах

## Не просто google it!

- разные среды исполнения и дистрибутивы
- разные версии программ даже в рамках одного дистрибутива
- саморазвитие

Section	Description
1	General <a href="#">commands</a>
2	<a href="#">System calls</a>
3	<a href="#">Library</a> functions, covering in particular the <a href="#">C standard library</a>
4	<a href="#">Special files</a> (usually devices, those found in /dev) and <a href="#">drivers</a>
5	<a href="#">File formats</a> and conventions
6	<a href="#">Games</a> and <a href="#">screensavers</a>
7	Miscellanea
8	System administration <a href="#">commands</a> and <a href="#">daemons</a>

## Различные варианты поиска по встроенной документации

- `man -k <keyword>`
- `man -P <pager>`
- секции `man` – открывает первую найденную, `man printf = man 1 printf`,  
`man 3 printf`
- `-h / --help`
- `info`
- `help`



# Изучайте GNU Coreutils!

Один из базовых пакетов с набором обсуждаемых в данной лекции программ – coreutils, или “базовые” утилиты. Их невозможно рассмотреть все в рамках лекций, так как состав пакета обширный: [русская Wikipedia по Coreutils](#).

Как минимум стоит знать о составе пакета и почитать help на команды, которыми вы пользуетесь ежедневно.

---

# Средства управления задачами bash

По умолчанию процессы запускаются в активном режиме, “на переднем плане”.

- Ctrl + C: прервать
- Ctrl + D: завершить
- Ctrl + Z: отправить в фон, background, bg
- Ctrl + S, Ctrl + Q: приостановить и продолжить

Управление фоновыми задачами (“заднего плана”):

- jobs: посмотреть список
- fg %<id>

Планирование выполнения задач в будущем:

- at / atq / atrm



---

# Изучайте инструменты, которые используете

- Размеры/шрифты/темы
- `$PS1`
- Базовое поведение можно менять – `stty -ixon` и `Ctrl + S`
- Горячие клавиши: `Esc + .`, `!!`
- Emacs-режим vs vi-режим
- `shopts (dotglob)`

---

# Итоги

- Познакомились с терминалом и интерактивным режимом командной строки,
- узнали о некоторых особенностях стандарта FHS,
- научились перемещаться по директориям, создавать, перемещать и удалять их; делать аналогичные действия с файлами,
- просматривать файлы разными способами,
- искать документацию встроенными средствами ОС,
- изменять рабочее окружение для удобства,
- управлять выполняемыми задачами и планировать выполнение в будущем.



# Домашнее задание

Давайте посмотрим ваше [домашнее задание](#).

- Вопросы по домашней работе задавайте **в чате** мессенджера Slack.
- Задачи можно сдавать **по частям**.
- Зачёт по домашней работе проставляется после того, как **приняты все задачи**.

**Задавайте вопросы и  
пишите отзыв о лекции!**

**Александр Крылов**