

Семинар по C++ №1

Работа в терминале

Что делать, если ваша ОС:

- Linux: всё супер
- MacOS: чуть хуже, но тоже хорошо
- Windows: беда

Варианты:

1. WSL (Windows Subsystem Linux)
2. Dual Boot
3. VirtualBox

ОСНОВЫ

- Ctrl + Alt + T – запустить терминал
- Tab - автодополнение
- \$ sudo [command] – запуск с правами суперпользователя (substitute **u**ser and **d**o)
- \$ sudo apt install [name] – скачать что-нибудь

Например:

\$ sudo apt install tree

MacOS moment: \$ brew install [name]

- \$ man [command] – открыть справочную информацию по команде (**m**anual)
 - \$ whoami – Узнать текущего пользователя (username)
 - \$ su [username] – Сменить пользователя
-
- **N.B.!** “\$” - начало строки в терминале, его писать самому не нужно

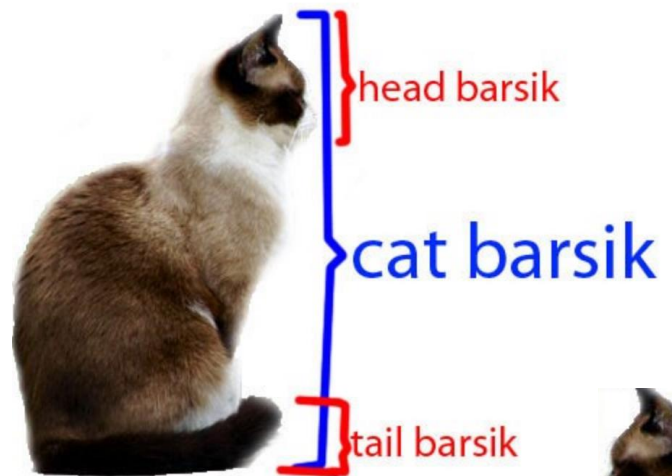
Удобства

- Alt + tab (*Cmd* + *tab*) – перемещение между активными окнами
- Стрелки вниз/вверх – перемещаться по истории команд
- Ctrl + R – искать по истории команд
 - Ещё раз Ctrl + R – следующая команда из истории
 - Enter – применить текущий вариант
- Ctrl + W – стереть слово
- Ctrl + U – стереть строку до курсора
- Ctrl + K – стереть строку после курсора
- Ctrl + A – перейти к началу строки
- Ctrl + E – перейти к концу строки
- Ctrl + Shift + C (*Cmd* + C) – копировать в терминале
- Ctrl + Shift + V (*Cmd* + V) – вставить в терминале

Вывод в консоль

- `$ more [filename]` – отобразить файл с возможностью пролистывания вверх
- `$ less [filename]` – как `more`, только лучше
- `$ grep [pattern] [file]` – найти подстроку в файле/папке
 - `grep -r` – найти рекурсивно в директории (**r**ecursive)
- `$ ack [substring] [file]` – как `grep`, только рекурсивный и более красивый
- `$ cat [filename]` – отобразить содержимое файла в консоль
 - `$ tac` – отобразить строки файла в обратном порядке
- `$ head [filename] / tail [filename]` – вывести начало/конец файла в консоль
 - `$ head -n 10 script.sh` – вывести первые 10 строк файла
- `|` перенаправление вывода одной команды во ввод другой
 - `$ ls | grep '.cpp'` – вывести все файлы из текущей директории, в названии которых есть `'.cpp'`

Наглядно



tac barsik



cat barsik | rev

Vim для самых маленьких

- `$ vim [filename]` – открыть файл в vim-е
- Esc – перейти в режим команд
- `:q` – выйти
- `:q!` – выйти без сохранения
- `:x` – выйти с сохранением
- `:w` – сохранить
- `:wq` – сохранить и выйти
- i – режим ввода (insert)



Навигация

- `$ pwd` – посмотреть, где мы сейчас находимся
- `$ tree` – посмотреть дерево файловой системы
- Скрытые файлы: `.gitignore`, `.git`, `.ssh`, `.`, `..`, ...
- `.` – текущая директория
- `..` – директория на уровень выше
- Флаги:
 - Короткие: `ls -a`
 - Длинные: `ls --all`
 - Комбинации флагов: `ls -l -a` или `ls -la`
 - `-h`, `--help` – часто используется для справки по команде
 - `-v`, `--version` – узнать версию программы (и вообще проверить, что она установлена)

ls и cd

- \$ ls [directory] – показать файлы в данном каталоге
 - \$ ls -l посмотреть директорию подробнее
 - \$ ls -R – вывести рекурсивно
 - \$ ls -a показать все файлы, включая скрытые
-
- \$ cd [directory] – перейти в директорию (**c**hange **d**irectory)
 - \$ cd .. – подняться на уровень вверх
 - \$ cd - – вернуться в предыдущую директорию
 - \$ cd / cd ~ / cd ~username – перейти в домашнюю директорию

Работа с файловой системой

- \$ mkdir [directory] – создать директорию (**make directory**)
- \$ touch [filename] – создать файл
- \$ mv [from] [to] – переместить файл
- \$ find [directory] -name [query] – найти файл в папке
- \$ chmod [options] [filename] - поменять (+/-) права (x/r/w) к файлу (**change mode**)
 - \$ chmod +x my_script.sh – добавить скрипту права на исполнение
- \$ cp [from] [to] - скопировать файл (**copy**)
 - \$ cp -r [from] [to] – скопировать рекурсивно
- \$ file [filename] – узнать про содержимое файла
- \$ rm [filename] – удалить файл (**remove**)
 - \$ rm -r [directory] – удалить рекурсивно
 - \$ rm -f [filename] – удалить принудительно

Ещё чуть-чуть напоследок

- Исполнение команды внутри другой команды:
 - `$ cat `cat file_names`` – прочитать файлы, имена которых лежат в файле `file_names`
 - `$ cat $(cat file_names)` – то же самое
 - `$ cat file_names | xargs cat` – команда `xargs` разбивает вывод команды (в данном случае команды «`cat file_names`») по пробелам и переносам строк и передаёт в качестве аргументов другой команде (в данном случае команда «`cat`»)

```
kostya@Kostya:~/seminars_C++/sem_1/test_cmd$ cat file_names
file_hello.txt
file_world.txt
kostya@Kostya:~/seminars_C++/sem_1/test_cmd$ cat file_hello.txt
hello
kostya@Kostya:~/seminars_C++/sem_1/test_cmd$ cat file_world.txt
world
kostya@Kostya:~/seminars_C++/sem_1/test_cmd$ cat `cat file_names`
hello
world
kostya@Kostya:~/seminars_C++/sem_1/test_cmd$ cat $(cat file_names)
hello
world
kostya@Kostya:~/seminars_C++/sem_1/test_cmd$ cat file_names | xargs cat
hello
world
```

Компиляция и запуск

- `$ g++ main.cpp` – компиляция программы
- `$ g++ main.cpp -o program` – скомпилировать программу и назвать исполняемый файл (по умолчанию `a.out`)
- `$./program` – запустить программу

`$ python3` – запустить интерпретатор питона в интерактивном режиме

Подключение к серверу

- `ip a` – узнать свой `ip`
- `ifconfig` – тоже посмотреть информацию про сетевые интерфейсы
- `ssh user@192.168.0.1` – подключиться к пользователю `user` по указанному `ip`
- `scp [from] [to]` – всё то же самое, что и с `cp`, но по `ssh`
 - `scp ./dir/file.txt user@192.168.0.1:/home/user_dir/new_file.txt` – скопировать файл на удалённый компьютер
 - `scp -r user@192.168.0.1:/home/user_dir/new_file.txt ./dir/file.txt` – скопировать папку к себе с удалённого компьютера

Задание:

1. Создайте локально (на своём компьютере) папку, где вы будете работать

2. Подключитесь по ssh к удалённому компьютеру:

```
ssh task1@192.168.43.82
```

```
password: task1
```

3. Скопируйте к себе папку с заданием

4. Выполните задание

5. Зациклить данную инструкцию, инкрементировав номер задания

- Можно скопировать папку и не подключаясь к серверу, локально используя команду `scp`

Процессы

- \$ ps -aux – Список запущенных процессов
- \$ pstree – Процессы в виде дерева
- \$ htop – Показать текущие запущенные процессы с графиками + плюшки

- Ctrl + C – Убить процесс
- Ctrl + Z – Остановить процесс
- Ctrl + D – Послать EOF (**E**nd **O**f **F**ile)

- N.B. Демо: inf_loop

Где заботать

- [bandit](#)
- Читать ман-ы
- Другие ресурсы