# Цели работы

* Изучить основы программирования в оболочке ОС UNIX/Linux.
* Научиться писать более сложные командные файлы.

# Задание

* Ознакомиться с теоретическим материалом.
* Написать три скрипта из задания.

# Теоретическое введение

Для написания скрипта 1 необходимо выяснить, что такое семафоры. [[1]](https://habr.com/ru/company/otus/blog/522418/)

Семафор – это механизм, который позволяет конкурирующим процессам и потокам работать с общими ресурсами и помогает в решении различных проблем синхронизации таких как гонки, дедлоки (взаимные блокировки) и неправильное поведение потоков.

Для решения этих проблем в ядре присутствуют такие средства как мьютексы, семафоры, сигналы и барьеры.

Есть три вида семафоров:

* Бинарные семафоры (binary semaphore)
* Семафоры-счетчики (counting semaphore)
* Массивы семафоров (semaphore set)

Краткий справочник команд Linux [[2]](https://hpc.icc.ru/documentation/cmnds.pdf)

# Выполнение лабораторной работы

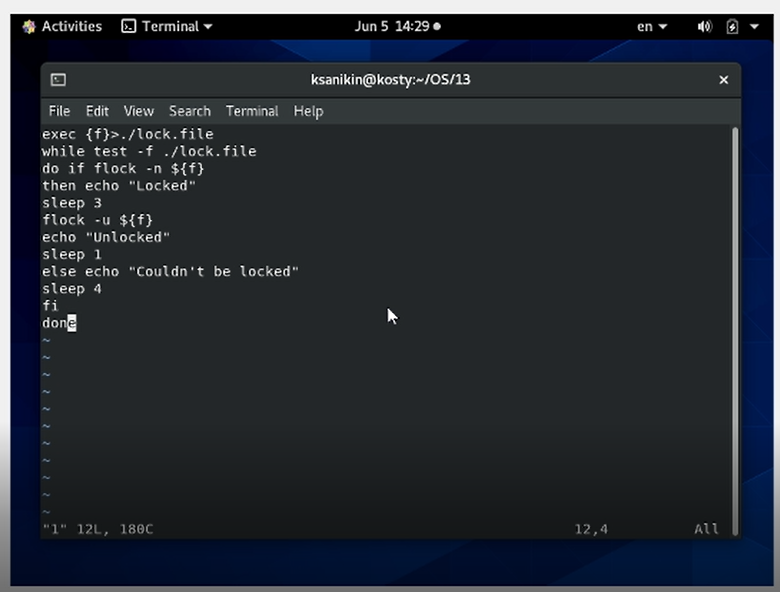
## Скрипт 1

Непосредственно сам скрипт. (рис. 1)

Пока файл открыт, проверяем его на используемость. Если файл используется, выводим Locked,

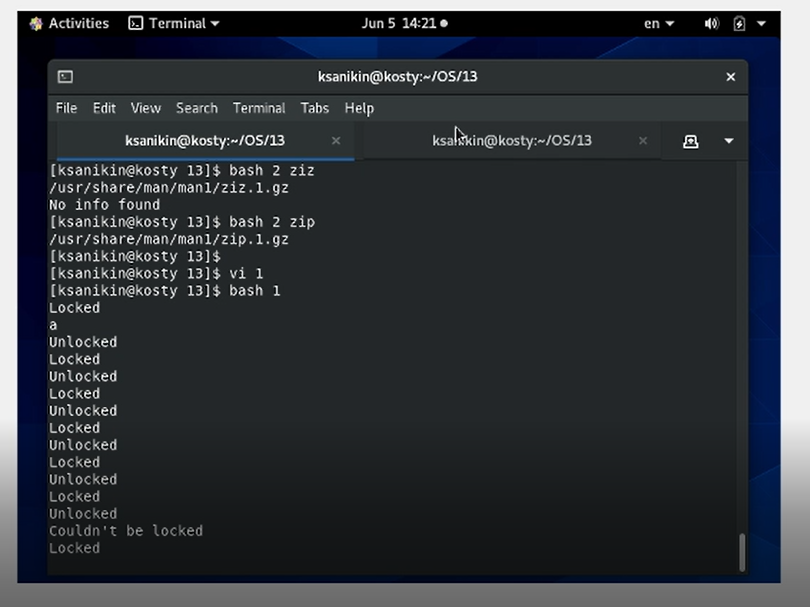
ждём 3 секунды и перестаём использовать файл. Если файл уже используется, выводим Couldn't lock,

ждём 4 секунды и пробуем открыть файл снова.



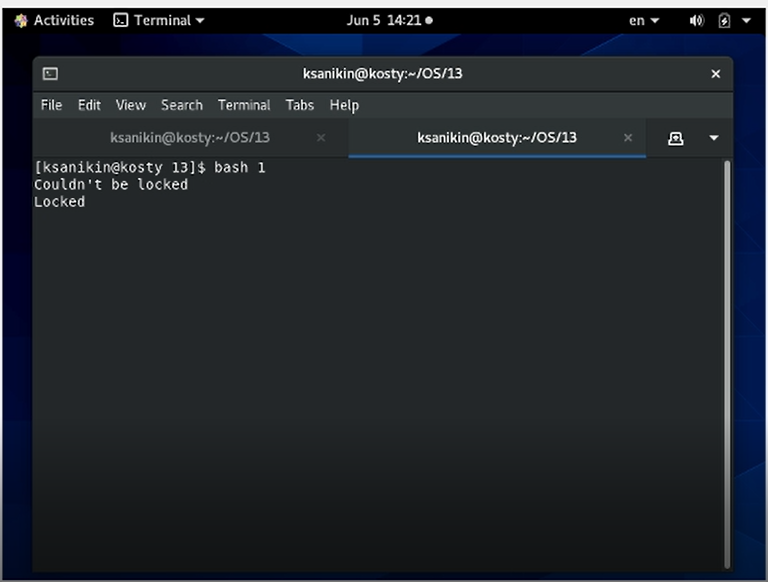
*Рис. 1: Скрипт 1*

Результаты работы скрипта 1, первый терминал (рис. 2)



*Рис. 2: Результаты работы скрипта 1*

Результаты работы скрипта 1, второй терминал (рис. 3)



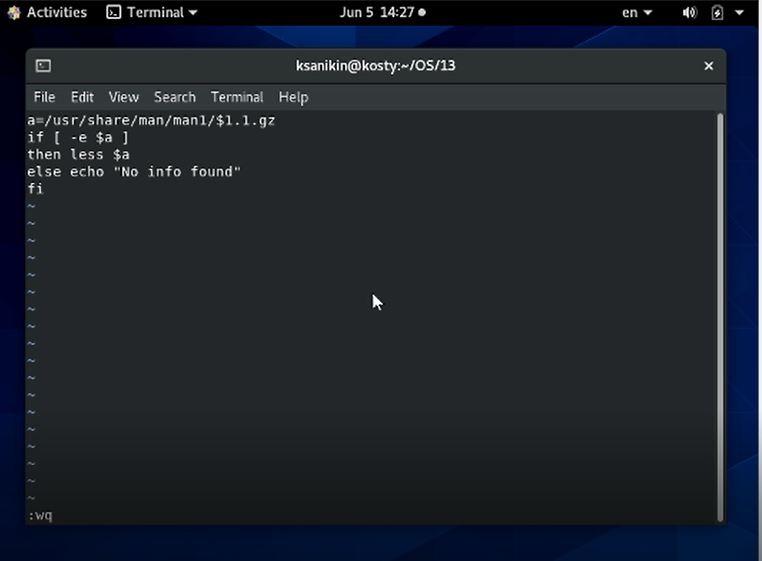
*Рис. 3: Результаты работы скрипта 1*

## Скрипт 2

Непосредственно сам скрипт. (рис. 4)

Определяем строчку, переданную аргументом. Проверяем, существует ли файл.

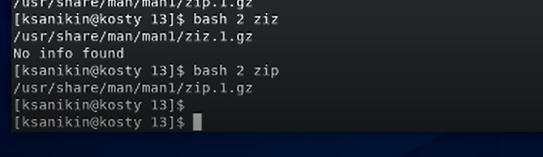
Если да, командой less выводим справку о нём. Иначе выводим "No info found".



*Рис. 4: Скрипт 2*

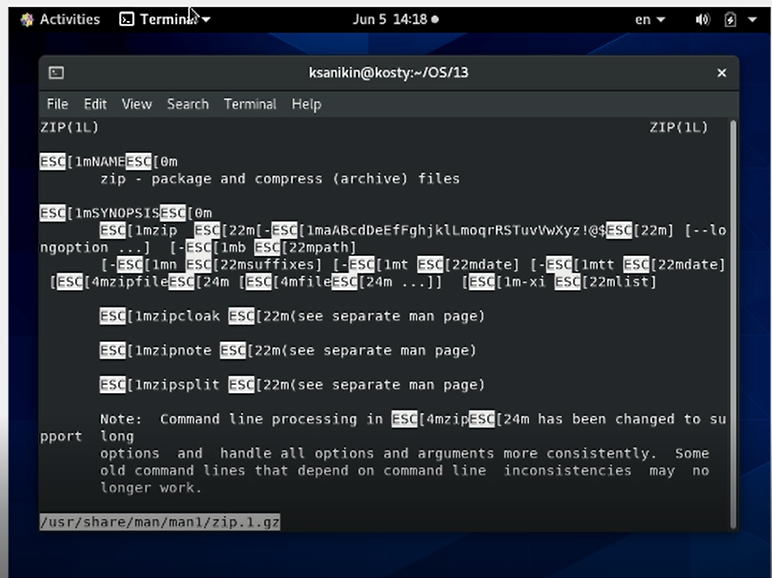
Результаты работы скрипта 2 с несуществующей справкой (рис. 5)

(выведение пути к файлу - остаток от дебага кода, в финальной версии путь не выводится)



*Рис. 5: Результаты работы скрипта 2*

Результаты работы скрипта 2 с существующей справкой (рис. 6)



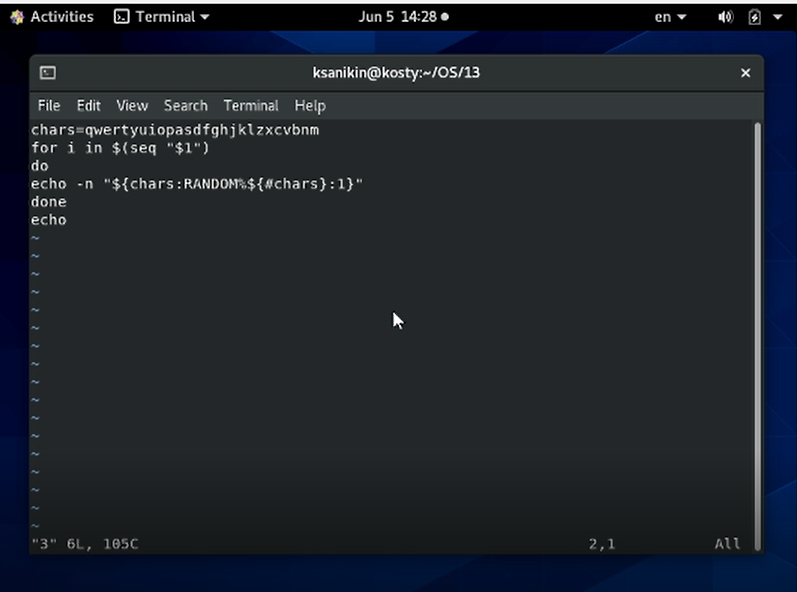
*Рис. 6: Результаты работы скрипта 2*

## Скрипт 3

Непосредственно сам скрипт. (рис. 7)

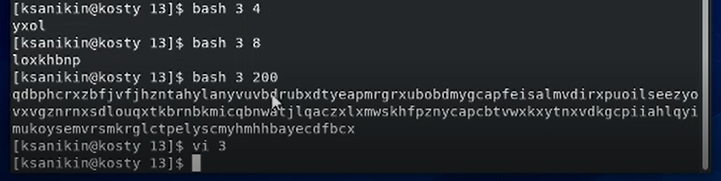
Создаём массив символов, после чего в цикле указанное аргументом число раз

выбираем случайный элемент массива и выводим его на экран



*Рис. 7: Скрипт 3*

Результаты работы скрипта 3 (рис. 8)



*Рис. 8: Результаты работы скрипта 3*

# Выводы

Скрипты созданы и работают. Работа выполнена полностью.

## Библиографический список

[1]: [Знакомимся с семафорами в Linux](https://habr.com/ru/company/otus/blog/522418/)

[2]: [Краткий справочник команд Linux](https://hpc.icc.ru/documentation/cmnds.pdf)