

# Лабораторная работа №14

## Презентация

---

Филиппьева К.Д.

11 мая 2024

Российский университет дружбы народов, Москва, Россия

# Информация

---

- Филиппева Ксения Дмитриевна
- Студент
- Российский университет дружбы народов
- 1132230795@pfur.ru

Изучить основы программирования в оболочке ОС UNIX. Научиться писать более сложные командные файлы с использованием логических управляющих конструкций и циклов.

## Задачи

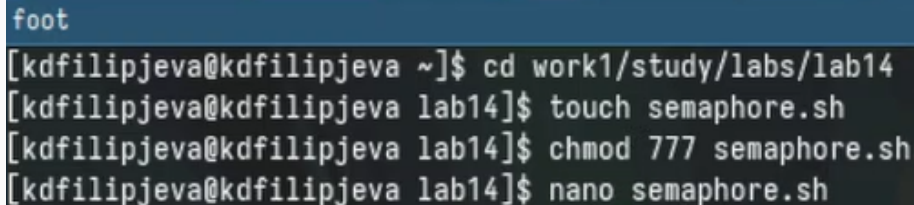
1. Написать командный файл, реализующий упрощённый механизм семафоров. Командный файл должен в течение некоторого времени  $t_1$  дожидаться освобождения ресурса, выдавая об этом сообщение, а дождавшись его освобождения, использовать его в течение некоторого времени  $t_2 < t_1$ , также выдавая информацию о том, что ресурс используется соответствующим командным файлом (процессом). Запустить командный файл в одном виртуальном терминале в фоновом режиме, перенаправив его вывод в другой ( $> /dev/tty\#$ , где  $\#$  — номер терминала куда перенаправляется вывод), в котором также запущен этот файл, но не фоновом, а в привилегированном режиме. Доработать программу так, чтобы имела возможность взаимодействия трёх и более процессов.

2. Реализовать команду `man` с помощью командного файла. Изучите содержимое каталога `/usr/share/man/man1`. В нем находятся архивы текстовых файлов, содержащих справку по большинству установленных в системе программ и команд. Каждый архив можно открыть командой `less` сразу же просмотрев содержимое справки. Командный файл должен получать в виде аргумента командной строки название команды и в виде результата выдавать справку об этой команде или сообщение об отсутствии справки, если соответствующего файла нет в каталоге `man1`.

- Используя встроенную переменную `$RANDOM`, напишите командный файл, генерирующий случайную последовательность букв латинского алфавита. Учтите, что `$RANDOM` выдаёт псевдослучайные числа в диапазоне от 0 до 32767.

## Создание файла

Создадим файл для первого задания, выдадим ему все права и войдем в него, чтобы вписать необходимый код

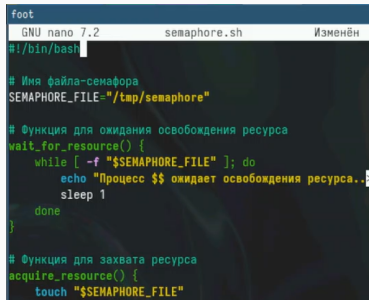
A terminal window with a dark background and a blue title bar labeled 'foot'. The terminal shows a series of commands being executed by a user named 'kdfilipjeva'. The commands are: 'cd work1/study/labs/lab14', 'touch semaphore.sh', 'chmod 777 semaphore.sh', and 'nano semaphore.sh'. The prompt changes from '~' to 'lab14' after the first command.

```
foot
[kdfilipjeva@kdfilipjeva ~]$ cd work1/study/labs/lab14
[kdfilipjeva@kdfilipjeva lab14]$ touch semaphore.sh
[kdfilipjeva@kdfilipjeva lab14]$ chmod 777 semaphore.sh
[kdfilipjeva@kdfilipjeva lab14]$ nano semaphore.sh
```

Рис. 1: 1 картинка



## Код для первого задания



```
foot
GNU nano 7.2      semaphore.sh      Изменён
#!/bin/bash

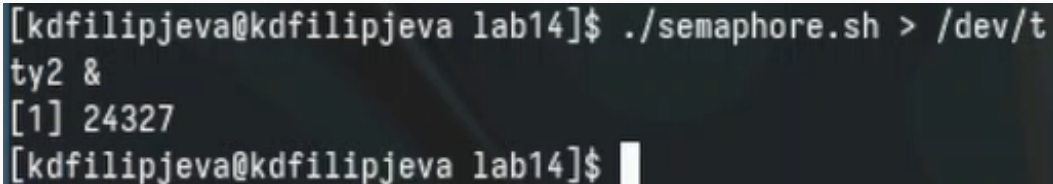
# Имя файла-семафора
SEMAPHORE_FILE="/tmp/semaphore"

# Функция для ожидания освобождения ресурса
wait_for_resource() {
    while [ -f "$SEMAPHORE_FILE" ]; do
        echo "Процесс $$ ожидает освобождения ресурса.."
        sleep 1
    done
}

# Функция для захвата ресурса
acquire_resource() {
    touch "$SEMAPHORE_FILE"
}
```

Рис. 2: 2 картинка

Активация кода и захват процесса

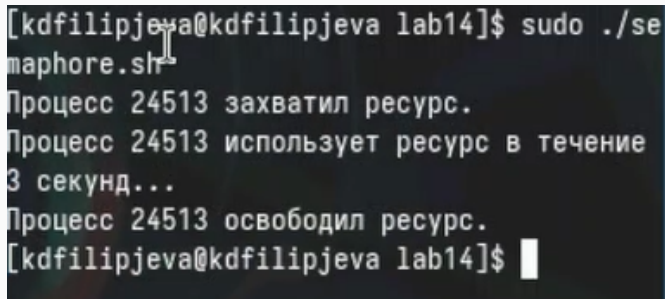


```
[kdfilipjeva@kdfilipjeva lab14]$ ./semaphore.sh > /dev/ty2 &  
[1] 24327  
[kdfilipjeva@kdfilipjeva lab14]$
```

Рис. 3: 3 картинка

## Работоспособность кода

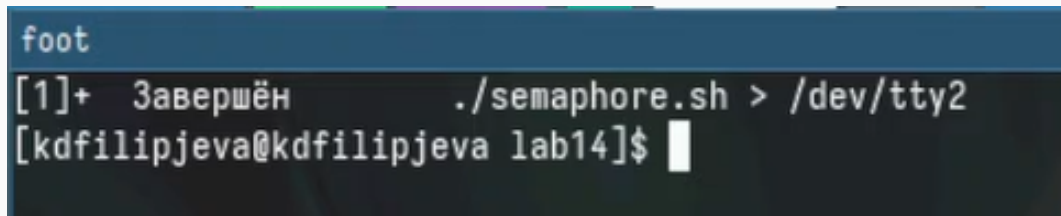
Активация кода во второй консоли, где он захватывается повторно, используется и высвобождается

A terminal window with a dark background and light-colored text. The prompt is [kdfilipjeva@kdfilipjeva lab14]\$. The user enters 'sudo ./semaphore.sh'. The output shows a process (24513) acquiring a resource, using it for 3 seconds, and then releasing it. The prompt returns to [kdfilipjeva@kdfilipjeva lab14]\$.

```
[kdfilipjeva@kdfilipjeva lab14]$ sudo ./semaphore.sh
Процесс 24513 захватил ресурс.
Процесс 24513 использует ресурс в течение
3 секунд...
Процесс 24513 освободил ресурс.
[kdfilipjeva@kdfilipjeva lab14]$
```

Рис. 4: 4 картинка

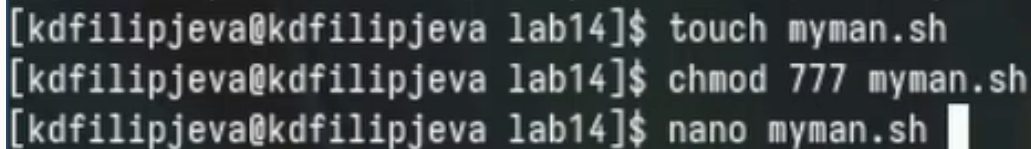
Автоматическое завершение процесса при очистке терминала

A terminal window with a dark background and light blue header. The header contains the text 'foot'. The main area shows a process '[1]+' with the status 'Завершён' (Completed) and the command './semaphore.sh > /dev/tty2'. Below this, the prompt '[kdfilipjeva@kdfilipjeva lab14]\$' is followed by a white cursor block.

```
foot
[1]+  Завершён      ./semaphore.sh > /dev/tty2
[kdfilipjeva@kdfilipjeva lab14]$
```

Рис. 5: 5 картинка

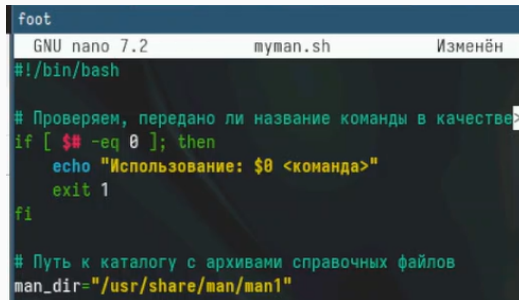
Создадим файл для второго задания и выдадим ему права

A terminal window with a dark background and light gray text. It shows three lines of commands being executed in a shell. The first line creates a file named 'myman.sh' using the 'touch' command. The second line sets permissions of '777' for 'myman.sh' using the 'chmod' command. The third line opens 'myman.sh' in the 'nano' text editor, with a white cursor visible at the end of the command.

```
[kdfilipjeva@kdfilipjeva lab14]$ touch myman.sh  
[kdfilipjeva@kdfilipjeva lab14]$ chmod 777 myman.sh  
[kdfilipjeva@kdfilipjeva lab14]$ nano myman.sh
```

**Рис. 6:** 6 картинка

Впишем необходимый код



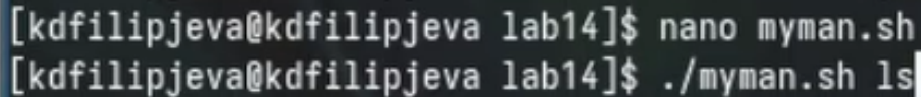
```
foot
GNU nano 7.2      myman.sh      Изменён
#!/bin/bash

# Проверяем, передано ли название команды в качестве>
if [ $# -eq 0 ]; then
    echo "Использование: $0 <команда>"
    exit 1
fi

# Путь к каталогу с архивами справочных файлов
man_dir="/usr/share/man/man1"
```

Рис. 7: 7 картинка

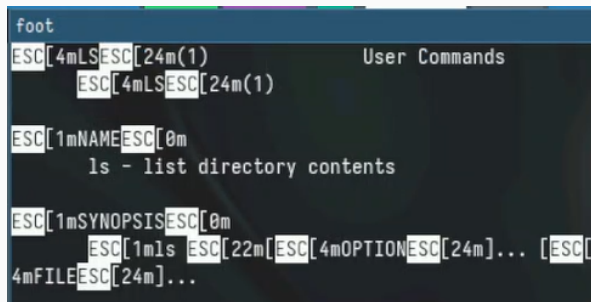
Активируем код для команды *ls*

A terminal window with a dark background and light gray text. The prompt is [kdfilipjeva@kdfilipjeva lab14]\$. The first command is nano myman.sh. The second command is ./myman.sh ls, followed by a white cursor block.

```
[kdfilipjeva@kdfilipjeva lab14]$ nano myman.sh  
[kdfilipjeva@kdfilipjeva lab14]$ ./myman.sh ls
```

Рис. 8: 8 картинка

Вывод активированного кода для команды *ls*



```
foot
ESC[4mLSESC[24m(1) User Commands
  ESC[4mLSESC[24m(1)

ESC[1mNAMEESC[0m
  ls - list directory contents

ESC[1mSYNOPSISESC[0m
  ESC[1mls ESC[22m[ESC[4mOPTIONESC[24m]... [ESC[
4mFILEESC[24m]...
```

Рис. 9: 9 картинка



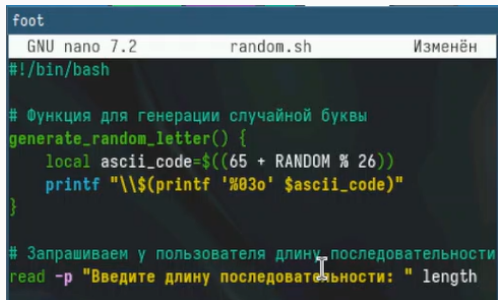
## Создание файла

Создадим файл для третьего задания

```
foot
[kdfilipjeva@kdfilipjeva lab14]$ touch random.sh
[kdfilipjeva@kdfilipjeva lab14]$ chmod 777 r
random.sh report/
[kdfilipjeva@kdfilipjeva lab14]$ chmod 777 random.sh
[kdfilipjeva@kdfilipjeva lab14]$ nano random.sh
```

Рис. 10: 10 картинка

Впишем необходимый нам код



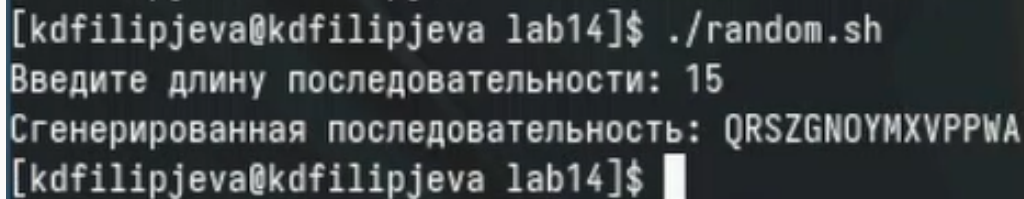
```
foot
GNU nano 7.2      random.sh      Изменён
#!/bin/bash

# Функция для генерации случайной буквы
generate_random_letter() {
    local ascii_code=$((65 + RANDOM % 26))
    printf "\\$(printf '%03o' $ascii_code)"
}

# Запрашиваем у пользователя длину последовательности
read -p "Введите длину последовательности: " length
```

Рис. 11: 11 картинка

Проверка кода

A terminal window with a dark background and light-colored text. The prompt is [kdfilipjeva@kdfilipjeva lab14]\$. The user enters ./random.sh. The output is Введите длину последовательности: 15. The user enters 15. The output is Сгенерированная последовательность: QRSZGNOYMXVPPWA. The prompt returns to [kdfilipjeva@kdfilipjeva lab14]\$.

```
[kdfilipjeva@kdfilipjeva lab14]$ ./random.sh
Введите длину последовательности: 15
Сгенерированная последовательность: QRSZGNOYMXVPPWA
[kdfilipjeva@kdfilipjeva lab14]$
```

Рис. 12: 12 картинка

Мы получили новые и отработали уже имеющиеся навыки программирования в оболочке ОС Linux.