

# **ОТЧЕТ О ВЫПОЛНЕНИИ ЛАБОРАТОРНОЙ РАБОТЫ №12**

*дисциплина: Операционные системы*

Лихтенштейн Алина Алексеевна

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>4</b>
<b>2</b>	<b>Задачи</b>	<b>5</b>
<b>3</b>	<b>Выполнение лабораторной работы</b>	<b>7</b>
3.1	Задание №1 . . . . .	7
3.2	Задание №2 . . . . .	8
<b>4</b>	<b>Выводы</b>	<b>12</b>

## Список иллюстраций

3.1	Исходный код скрипта №1 . . . . .	7
3.2	Результат выполнения скрипта №1 . . . . .	8
3.3	Исходный код скрипта №2 . . . . .	9
3.4	Исходный код скрипта №3 . . . . .	10
3.5	Результат выполнения скрипта №3 . . . . .	11

# 1 Цель работы

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

## 2 Задачи

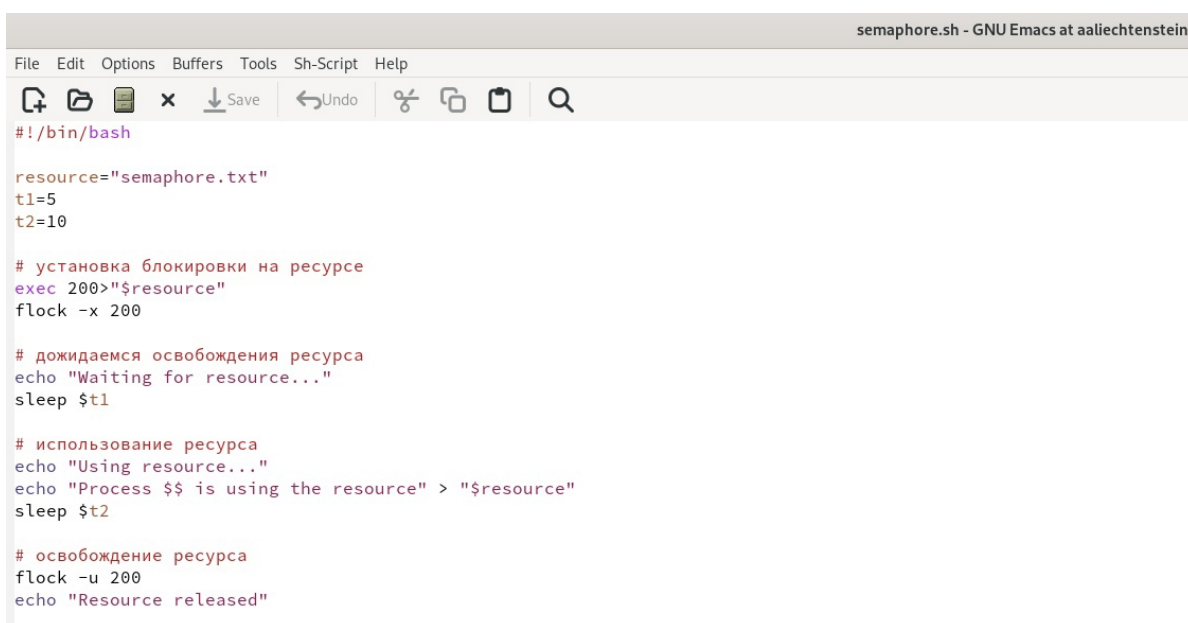
1. Написать командный файл, реализующий упрощённый механизм семафоров. Командный файл должен в течение некоторого времени  $t_1$  дожидаться освобождения ресурса, выдавая об этом сообщение, а дождавшись его освобождения, использовать его в течение некоторого времени  $t_2 < t_1$ , также выдавая информацию 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 имелась 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`.
3. 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.

## 3 Выполнение лабораторной работы

### 3.1 Задание №1

1. Данный скрипт реализует упрощенный механизм семафоров на основе файловой блокировки. Он дожидается освобождения ресурса, выдавая об этом сообщение, затем использует его, выдавая информацию о том, что ресурс используется соответствующим процессом. Ресурс защищен блокировкой на файле semaphore.txt. Скрипт можно запустить в фоновом режиме и перенаправить его вывод в другой терминал. (рис. [3.1], [3.2])

A screenshot of a terminal window titled "semaphore.sh - GNU Emacs at aaliechtenstein". The window shows the source code of a shell script. The script sets a resource to "semaphore.txt", defines two time intervals (t1=5, t2=10), and implements a semaphore mechanism using flock. It includes comments in Russian for each step: setting the lock, waiting for the resource, using the resource, and releasing it. The script uses 'exec 200>"\$resource"' to open the file for locking, 'flock -x 200' to acquire the lock, 'sleep \$t1' to wait, 'echo "Waiting for resource..."' to show status, 'sleep \$t2' to use the resource, 'flock -u 200' to release the lock, and 'echo "Resource released"' to show status.

```
#!/bin/bash

resource="semaphore.txt"
t1=5
t2=10

# установка блокировки на ресурсе
exec 200>"$resource"
flock -x 200

# ожидаем освобождения ресурса
echo "Waiting for resource..."
sleep $t1

# использование ресурса
echo "Using resource..."
echo "Process $$ is using the resource" > "$resource"
sleep $t2

# освобождение ресурса
flock -u 200
echo "Resource released"
```

Рис. 3.1: Исходный код скрипта №1

```
aaliechtenstein@aaliechtenstein:~/scripts/script1 — tmux
Запуск скрипта в терминале /dev/pts/27
Запуск скрипта в терминале /dev/pts/28
Запуск скрипта в терминале /dev/pts/3
Запуск скрипта в терминале /dev/pts/4
Запуск скрипта в терминале /dev/pts/5
Запуск скрипта в терминале /dev/pts/6
Запуск скрипта в терминале /dev/pts/7
Запуск скрипта в терминале /dev/pts/8
Запуск скрипта в терминале /dev/pts/9
[aaliechtenstein@aaliechtenstein script1]
[aaliechtenstein@aaliechtenstein script1]
[aaliechtenstein@aaliechtenstein script1]$ tty
/dev/pts/27
[aaliechtenstein@aaliechtenstein script1]
[aaliechtenstein@aaliechtenstein script1]$
[aaliechtenstein@aaliechtenstein script1]$ tty
/dev/pts/26
[aaliechtenstein@aaliechtenstein script1]$
[aaliechtenstein@aaliechtenstein script1]$ tty
/dev/pts/28
[aaliechtenstein@aaliechtenstein script1]$
```

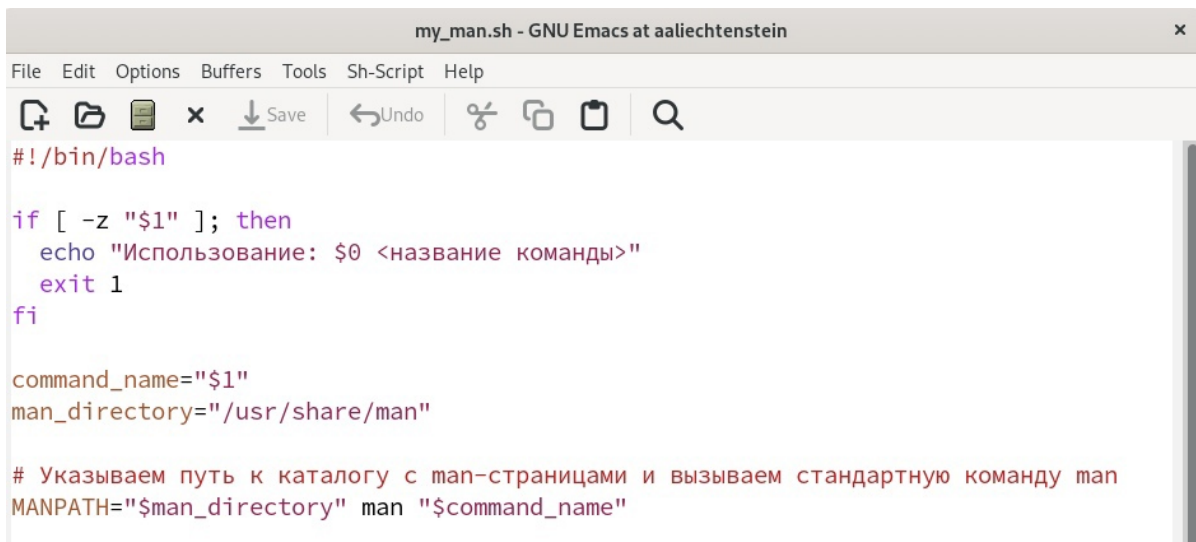
Рис. 3.2: Результат выполнения скрипта №1

## 3.2 Задание №2

2. Этот скрипт предоставляет пользователю возможность получить справку для команды, указанной в качестве аргумента. Он использует стандартную команду `man` с указанием пути к каталогу с `man`-страницами через переменную окружения `MANPATH`.

Скрипт проверяет, указано ли имя команды в качестве аргумента. Если аргумент не указан, выводит справочное сообщение и завершает работу. Запоминает указанное имя команды и путь к каталогу `man`-страниц. Устанавливает значение переменной окружения `MANPATH` равным пути к каталогу `man`-страниц. Вызывает стандартную команду `man` с именем команды в качестве аргумента. (рис. [3.3], [??])





```
my_man.sh - GNU Emacs at aaliechtenstein
File Edit Options Buffers Tools Sh-Script Help
[Icons: Open, Save, Undo, Redo, Copy, Paste, Find]

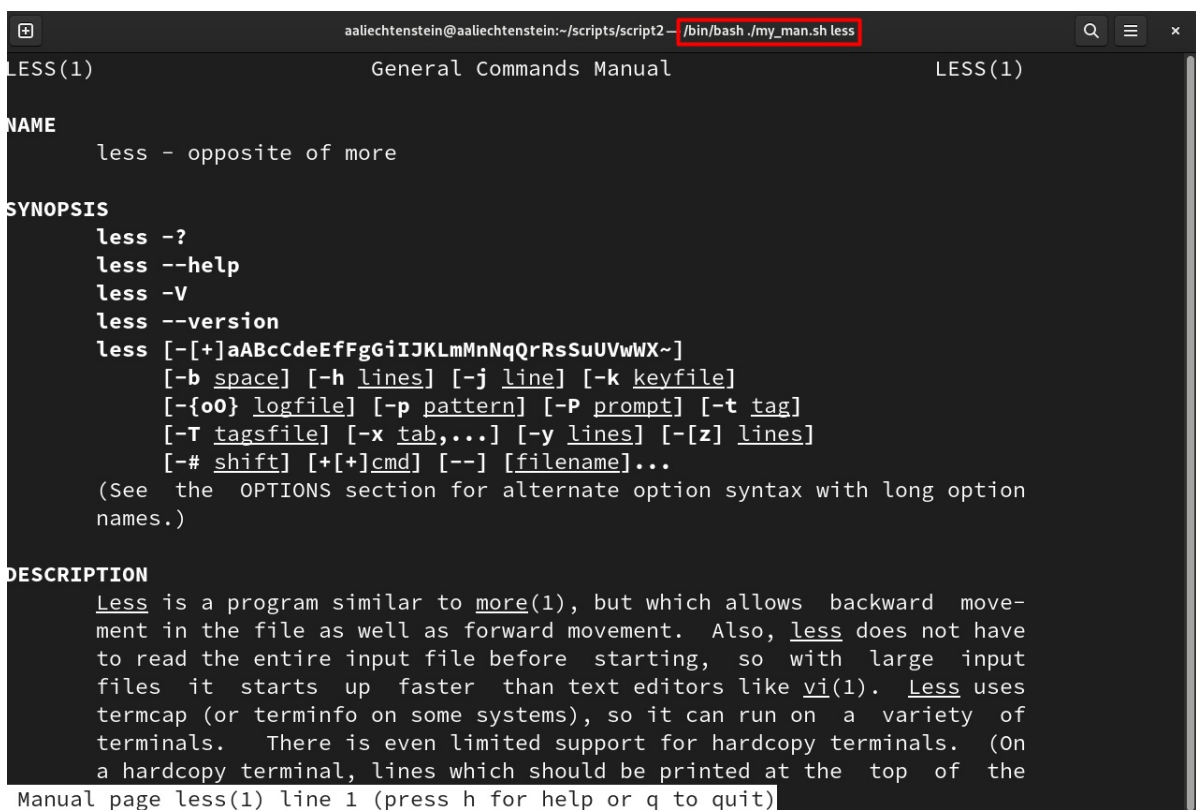
#!/bin/bash

if [ -z "$1" ]; then
    echo "Использование: $0 <название команды>"
    exit 1
fi

command_name="$1"
man_directory="/usr/share/man"

# Указываем путь к каталогу с man-страницами и вызываем стандартную команду man
MANPATH="$man_directory" man "$command_name"
```

Рис. 3.3: Исходный код скрипта №2



```
aaliechtenstein@aaliechtenstein: ~/scripts/script2 - /bin/bash ./my_man.sh less
LESS(1) General Commands Manual LESS(1)

NAME
    less - opposite of more

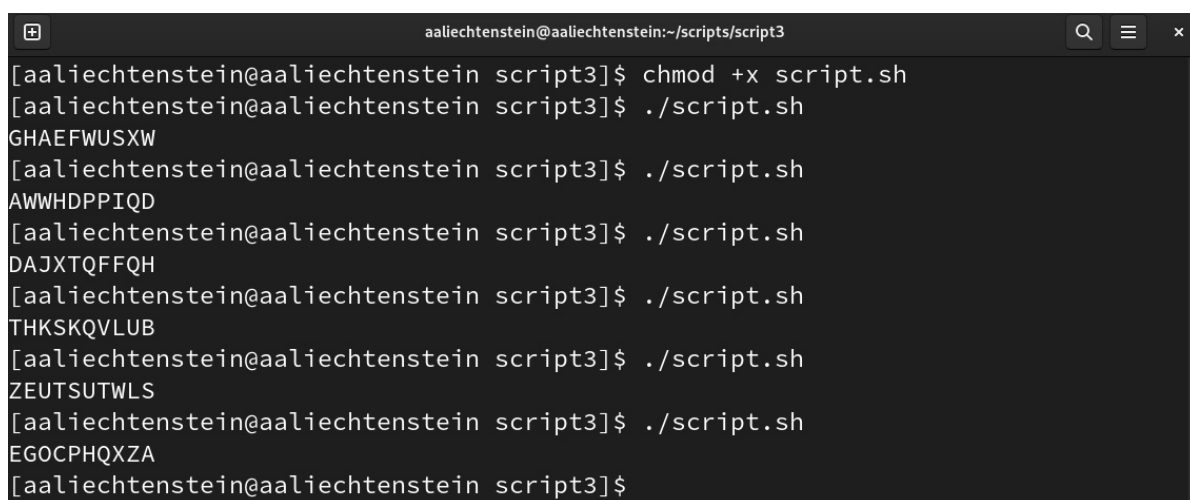
SYNOPSIS
    less -?
    less --help
    less -v
    less --version
    less [-[+]aABcCdeEfFgGiIJKLmMnNqQrRsSuUVwWx~]
        [-b space] [-h lines] [-j line] [-k keyfile]
        [-{oo} logfile] [-p pattern] [-P prompt] [-t tag]
        [-T tagsfile] [-x tab,...] [-y lines] [-[z] lines]
        [-# shift] [+ [+]cmd] [--] [filename]...
    (See the OPTIONS section for alternate option syntax with long option
    names.)

DESCRIPTION
    Less is a program similar to more(1), but which allows backward move-
    ment in the file as well as forward movement. Also, less does not have
    to read the entire input file before starting, so with large input
    files it starts up faster than text editors like vi(1). Less uses
    termcap (or terminfo on some systems), so it can run on a variety of
    terminals. There is even limited support for hardcopy terminals. (On
    a hardcopy terminal, lines which should be printed at the top of the
    Manual page less(1) line 1 (press h for help or q to quit)
```

## Задание №3 3. Скрипт генерирует случайную последовательность латинских букв заданной длины. Он использует переменную \$RANDOM для получения

псевдослучайных чисел, преобразует их в соответствующие буквы и объединяет их в строку. Затем выводит эту строку на экран. (рис. [3.4], [3.5])

Рис. 3.4: Исходный код скрипта №3



```
aaliechtenstein@aaliechtenstein:~/scripts/script3
[aaliechtenstein@aaliechtenstein script3]$ chmod +x script.sh
[aaliechtenstein@aaliechtenstein script3]$ ./script.sh
GHAEFWUSXW
[aaliechtenstein@aaliechtenstein script3]$ ./script.sh
AWWHDPPIQD
[aaliechtenstein@aaliechtenstein script3]$ ./script.sh
DAJXTQFFQH
[aaliechtenstein@aaliechtenstein script3]$ ./script.sh
THKSKQVLUB
[aaliechtenstein@aaliechtenstein script3]$ ./script.sh
ZEUTSUTWLS
[aaliechtenstein@aaliechtenstein script3]$ ./script.sh
EGOCPHQXZA
[aaliechtenstein@aaliechtenstein script3]$
```

Рис. 3.5: Результат выполнения скрипта №3

## 4 Выводы

Были получены практические навыки написания более сложные командных файлов с использованием логических управляющих конструкций и циклов.