

Лабораторная работа №14

Операционные системы

Башиянц Александра Кареновна

Содержание

1	Цель работы	4
2	Задание	5
3	Выполнение лабораторной работы	6
4	Выводы	9

Список иллюстраций

3.1	файлы sh	6
3.2	ex1.sh	6
3.3	bash ex1.sh	6
3.4	ex2.sh	7
3.5	bash ex2.sh	7
3.6	ex3.sh	8
3.7	bash ex3.sh	8

1 Цель работы

Цель данной работы — изучить основы программирования в оболочке ОС UNIX. Научиться писать более сложные командные файлы с использованием логических управляющих конструкций и циклов.

2 Задание

В этой лабораторной работе необходимо изучить работу bash-скриптов.

Необходимо научиться:

- Использовать сложные логические управляющие конструкции
- Использовать циклы

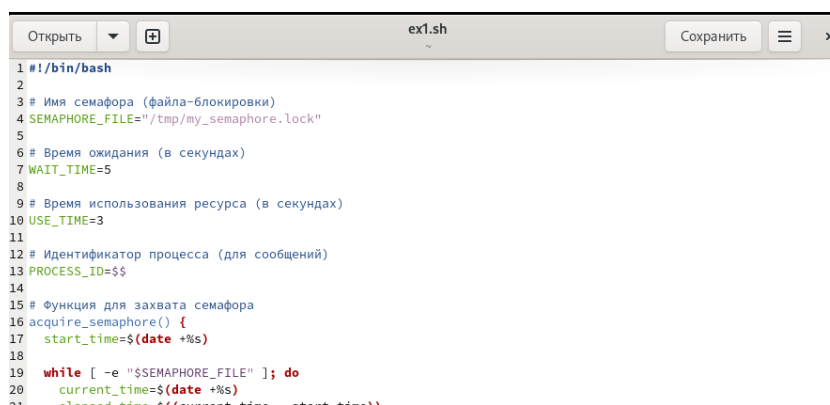
3 Выполнение лабораторной работы

Создадим файлы ex1.sh-ex4.sh для выполнения работы (рис. 3.1).

```
akbashiyanc@akbashiyanc:~$ touch ex1.sh
```

Рис. 3.1: файлы sh

Напишем скрипт для задания 1 (рис. 3.2).

The image shows a text editor window titled 'ex1.sh'. The script content is as follows:

```
1 #!/bin/bash
2
3 # Имя семафора (файла-блокировки)
4 SEMAPHORE_FILE="/tmp/my_semaphore.lock"
5
6 # Время ожидания (в секундах)
7 WAIT_TIME=5
8
9 # Время использования ресурса (в секундах)
10 USE_TIME=3
11
12 # Идентификатор процесса (для сообщений)
13 PROCESS_ID=$$
14
15 # Функция для захвата семафора
16 acquire_semaphore() {
17     start_time=$(date +%s)
18
19     while [ -e "$SEMAPHORE_FILE" ]; do
20         current_time=$(date +%s)
21         elapsed_time=$((current_time - start_time))
```

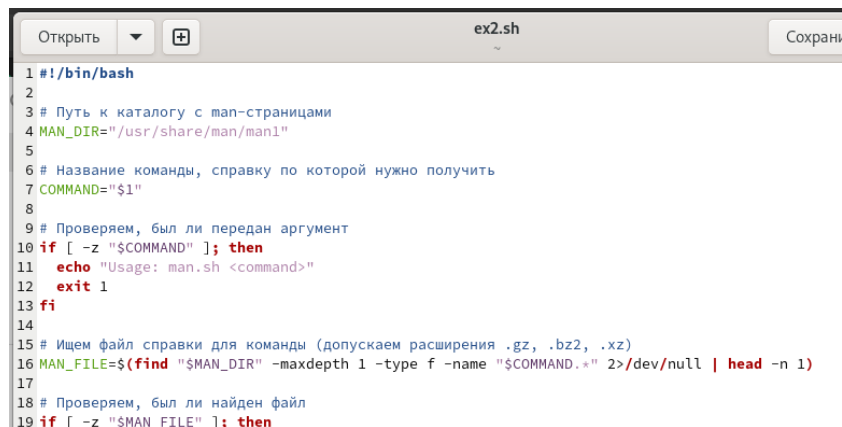
Рис. 3.2: ex1.sh

Выполним файл ex1.sh и проверим корректность выполнения (рис. 3.3).

```
akbashiyanc@akbashiyanc:~$ ./ex1.sh /work
Process 3524: Acquired semaphore.
Process 3524: Using resource for 3 seconds...
Process 3524: Released semaphore.
akbashiyanc@akbashiyanc:~$
```

Рис. 3.3: bash ex1.sh

Напишем скрипт для задания 2 (рис. 3.4).



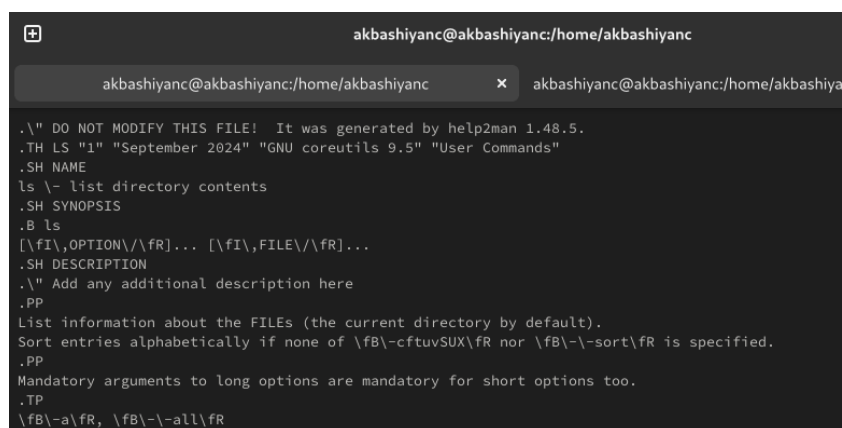
```

1 #!/bin/bash
2
3 # Путь к каталогу с man-страницами
4 MAN_DIR="/usr/share/man/man1"
5
6 # Название команды, справку по которой нужно получить
7 COMMAND="$1"
8
9 # Проверяем, был ли передан аргумент
10 if [ -z "$COMMAND" ]; then
11     echo "Usage: man.sh <command>"
12     exit 1
13 fi
14
15 # Ищем файл справки для команды (допускаем расширения .gz, .bz2, .xz)
16 MAN_FILE=$(find "$MAN_DIR" -maxdepth 1 -type f -name "$COMMAND.*" 2>/dev/null | head -n 1)
17
18 # Проверяем, был ли найден файл
19 if [ -z "$MAN_FILE" ]; then

```

Рис. 3.4: ex2.sh

Выполним файл ex2.sh и проверим корректность выполнения (рис. 3.5).



```

akbashiyanc@akbashiyanc:/home/akbashiyanc
akbashiyanc@akbashiyanc:/home/akbashiyanc x akbashiyanc@akbashiyanc:/home/akbashiyanc

.\" DO NOT MODIFY THIS FILE! It was generated by help2man 1.48.5.
.TH LS "1" "September 2024" "GNU coreutils 9.5" "User Commands"
.SH NAME
ls \- list directory contents
.SH SYNOPSIS
.B ls
[\fI\,OPTION\|\fR]... [\fI\,FILE\|\fR]...
.SH DESCRIPTION
.\" Add any additional description here
.PP
List information about the FILES (the current directory by default).
Sort entries alphabetically if none of \fB\--cftuvSUX\|fR nor \fB\--sort\|fR is specified.
.PP
Mandatory arguments to long options are mandatory for short options too.
.TP
\fB\-a\|fR, \fB\--all\|fR

```

Рис. 3.5: bash ex2.sh

Напишем скрипт для задания 3 (рис. 3.6).



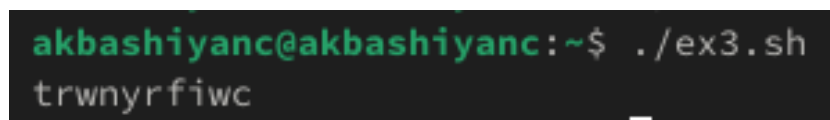
```

1 #!/bin/bash
2
3 # Длина последовательности
4 LENGTH=10
5
6 # Функция для генерации случайной буквы
7 generate_random_letter() {
8     random_index=$((RANDOM % 26))
9     ascii_code=$((random_index + 97))
10
11     # Выводим букву
12     printf "%b" "\\$(printf '%03o' "$ascii_code")"
13 }
14
15 # Генерируем последовательность букв
16 for i in $(seq 1 "$LENGTH"); do
17     generate_random_letter
18 done
19

```

Рис. 3.6: ex3.sh

Выполним файл ex3.sh и проверим корректность выполнения (рис. 3.7).



```

akbashiyanc@akbashiyanc:~$ ./ex3.sh
trwnyrfiwc

```

Рис. 3.7: bash ex3.sh

4 Выводы

В этой лабораторной работе мы изучили углубленную работу bash-скриптов.