

# **Отчет по лабораторной работе №13**

**Дисциплина: Операционные системы**

**Чекалова Лилия Руслановна, ст.б. 1032201654**

# Содержание

1	Цель работы	4
2	Выполнение лабораторной работы	5
3	Выводы	12
4	Библиография	13

# List of Figures

2.1	Реализация метода семафоров . . . . .	6
2.2	Запуск командного файла . . . . .	7
2.3	Модификация командного файла . . . . .	8
2.4	Работа программы на двух терминалах . . . . .	9
2.5	Содержимое каталога . . . . .	9
2.6	Реализация команды map . . . . .	10
2.7	Запуск справки . . . . .	10
2.8	Результат работы справки . . . . .	10
2.9	Реализация получения случайной последовательности . . . . .	11
2.10	Вывод случайной последовательности . . . . .	11

# 1 Цель работы

Изучение основ программирования в оболочке UNIX/Linux, приобретение навыков написания более сложных командных файлов с использованием логических управляющих конструкций и циклов.

## 2 Выполнение лабораторной работы

Пишу командный файл, получающий два значения времени в переменные `t1` и `t2`. Завожу переменные-счетчики времени и организую два цикла `while`. Первый выполняется в течение периода времени `t1`, на экран командой `echo` выводится сообщение “waiting”. Второй цикл выполняется в течение промежутка времени `t2`, на экран выводится сообщение “using”. Перед началом второго цикла переменным-счетчикам времени присваиваются новые значения (рис. 2.1)

```
#!/bin/bash
t1=$1
t2=$2
s1=$(date +%S)
s2=$(date +%S)
((t=$s2-$s1))
while ((t < t1))
do
    echo "waiting"
    sleep 1
    s2=$(date +%S)
    ((t=$s2-$s1))
done
s1=$(date +%S)
s2=$(date +%S)
((t=$s2-$s1))
while ((t < t2))
do
    echo "using"
    sleep 1
    s2=$(date +%S)
    ((t=$s2-$s1))
done
```

Figure 2.1: Реализация метода семафоров

Передаю файлу права на исполнение командой `chmod +x` и запускаю его (рис. 2.2)

```
lrchekalova@dk8n72 ~ $ chmod +x n_1.sh
lrchekalova@dk8n72 ~ $ ./n_1.sh
lrchekalova@dk8n72 ~ $ ./n_1.sh 3 4
waiting
waiting
waiting
using
using
using
using
lrchekalova@dk8n72 ~ $
```

Figure 2.2: Запуск командного файла

Модифицирую командный файл, переношу выполнение предыдущих циклов в функции `waiting` и `using`. Теперь программа принимает на вход три значения: два временных отрезка и значение операции. Если переданная операция равна `wait` или `use` выполняется функция `waiting` или `using` соответственно. Когда программа получает значение операции `exit`, она завершает работу (рис. 2.3)

```
#!/bin/bash
function waiting
{
    s1=$(date +%S)
    s2=$(date +%S)
    ((t=s2-s1))
    while ((t < t1))
    do
        echo "waiting"
        sleep 1
        s2=$(date +%S)
        ((t=s2-s1))
    done
}
function using
{
    s1=$(date +%S)
    s2=$(date +%S)
    ((t=s2-s1))
    while ((t < t2))
    do
        echo "using"
        sleep 1
        s2=$(date +%S)
        ((t=s2-s1))
    done
}
t1=$1
t2=$2
operation=$3
while true
do
    if [ "$operation" == "exit" ]
    then
        echo "exiting"
        exit 0
    fi
    if [ "$operation" == "wait" ]
    then waiting
    fi
    if [ "$operation" == "use" ]
    then using
    fi
    echo "next action: "
    read operation
done
```

Figure 2.3: Модификация командного файла



Запускаю программу, передавав ей значение операции и направляя результат на четвертый терминал (рис. 2.4)

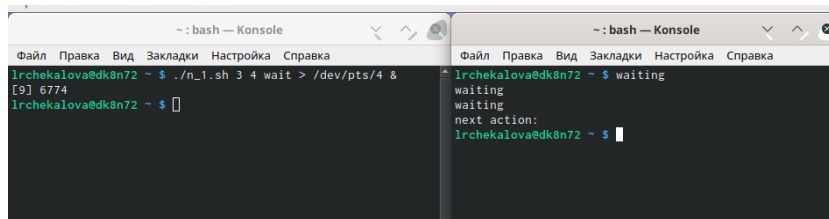


Figure 2.4: Работа программы на двух терминалах

Просматриваю содержимое каталога /usr/share/man/man1 командой ls (рис. 2.5)

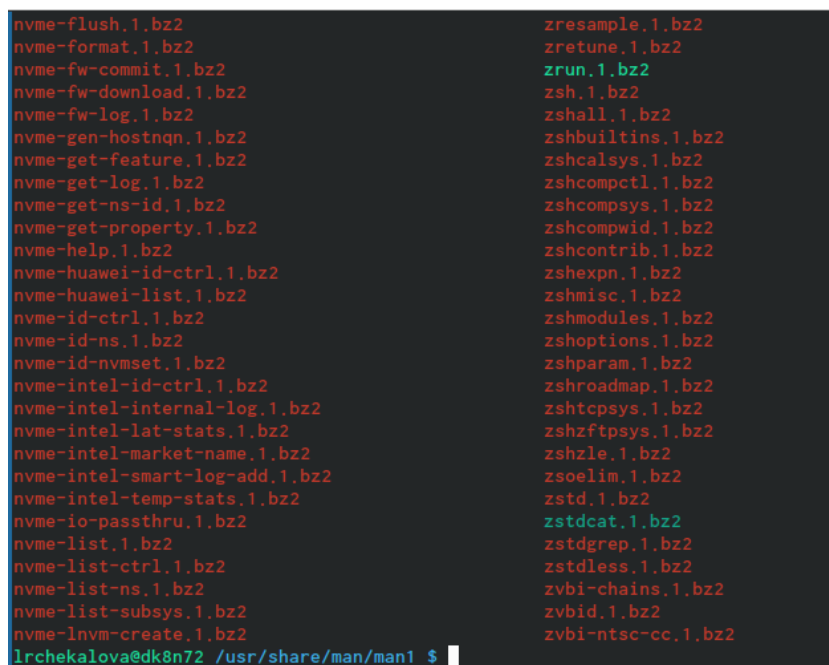


Figure 2.5: Содержимое каталога

Пишу командный файл, принимающий на вход имя команды. Если архив или файл с таким именем найден, его содержимое выводится на экран командой less, в противном случае выводится сообщение об ошибке (рис. 2.6)

```
#!/bin/bash
name=$1
if [ -a /usr/share/man/man1/$name.1.bz2 ]
then
    less /usr/share/man/man1/$name.1.bz2
elif [ -a /usr/share/man/man1/$name.1 ]
then less /usr/share/man/man1/$name.1
else
    echo "no manual found"
fi
```

Figure 2.6: Реализация команды man

Запускаю программу, запрашивая справку о команде cat (рис. 2.7)

```
nekalova@dk8n72 ~ $ bash n_2.sh cat
```

Figure 2.7: Запуск справки

На экран выводится справка по команде cat, находящаяся в архиве cat.1.bz2 (рис. 2.8)

```
(1) User Commands CAT(1)
cat - concatenate files and print on the standard output
SYNOPSIS
cat [OPTION]... [FILE]...
DESCRIPTION
Concatenate FILE(s) to standard output.
With no FILE, or when FILE is -, read standard input.
-A, --show-all
    equivalent to -vET
-b, --number-nonblank
    number nonempty output lines, overrides -n
-e
    equivalent to -vE
-E, --show-ends
    display $ at end of each line
-n, --number
    number all output lines
-s, --squeeze-blank
    suppress one line of blank lines
./share/man/man1/cat.1.bz2 lines 1-28
```

Figure 2.8: Результат работы справки

Пишу командный файл, получающий на вход количество букв последовательности. В цикле for передаю переменной letter значение \$RANDOM в диапазоне

от 1 до 26 и вывожу на экран соответствующую номеру букву в конструкции case (рис. 2.9)

```
#!/bin/bash
k=$1
for (( i=0; i<$k; i++ ))
do
  ((letter=$RANDOM%26+1 ))
  case $letter in
    1) echo -n a;; 2) echo -n b;; 3) echo -n c;; 4) echo -n d;; 5) echo -n e;;
    6) echo -n f;; 7) echo -n g;; 8) echo -n h;; 9) echo -n i;; 10) echo -n j;;
    11) echo -n k;; 12) echo -n l;; 13) echo -n m;; 14) echo -n n;; 15) echo -n o;;
    16) echo -n p;; 17) echo -n q;; 18) echo -n r;; 19) echo -n s;; 20) echo -n t;;
    21) echo -n u;; 22) echo -n v;; 23) echo -n w;; 24) echo -n x;; 25) echo -n y;;
    26) echo -n z;;
  esac
done
echo
```

Figure 2.9: Реализация получения случайной последовательности

Запускаю программу, чтобы получить последовательность из 7 букв латинского алфавита (рис. 2.10)

```
lrchekalova@dk8n72 ~ $ bash n_3.sh 7
zbixron
lrchekalova@dk8n72 ~ $
```

Figure 2.10: Вывод случайной последовательности

## 3 Выводы

После выполнения данной лабораторной работы я научилась писать более сложные командные файлы с использованием условных конструкций if-elif-else и case, циклов for и while и функций.

## 4 Библиография

1. Machtelt Garrels. Bash Guide For Beginners, 2008
2. Mendel Cooper. Advanced Bash-Scripting Guide, 2004
3. Теоретические материалы к лабораторной работе: [https://esystem.rudn.ru/pluginfile.php/1111111/mod\\_resource/content/1/lab\\_shell\\_prog\\_1.pdf](https://esystem.rudn.ru/pluginfile.php/1111111/mod_resource/content/1/lab_shell_prog_1.pdf)