

# **Лабораторная работа 12**

Сафин Андрей Алексеевич

# Содержание

|          |                                       |           |
|----------|---------------------------------------|-----------|
| <b>1</b> | <b>Цель работы</b>                    | <b>5</b>  |
| <b>2</b> | <b>Задание</b>                        | <b>6</b>  |
| <b>3</b> | <b>Теоретическое введение</b>         | <b>7</b>  |
| <b>4</b> | <b>Выполнение лабораторной работы</b> | <b>8</b>  |
| <b>5</b> | <b>Выводы</b>                         | <b>12</b> |

## Список иллюстраций

|     |  |    |
|-----|--|----|
| 4.1 | Первая программа . . . . .             | 8  |
| 4.2 | Выполнение первой программы . . . . .  | 9  |
| 4.3 | Выполнение первой программы . . . . .  | 9  |
| 4.4 | Вторая программа . . . . .             | 9  |
| 4.5 | Выполнение второй программы . . . . .  | 10 |
| 4.6 | Третья программа . . . . .             | 10 |
| 4.7 | Выполнение третьей программы . . . . . | 11 |

## **Список таблиц**

# 1 Цель работы

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

## **2 Задание**

Написать программы, описанные в лабораторной работе.

### 3 Теоретическое введение

bash (Bourne Again Shell) - это интерпретатор команд, позволяющий пользователю взаимодействовать с операционной системой через терминал. Bash способен считывать не только команды, вводимые в интерактивном режиме, но и файлы с командами. Эту функцию можно использовать для написания командных файлов.

## 4 Выполнение лабораторной работы

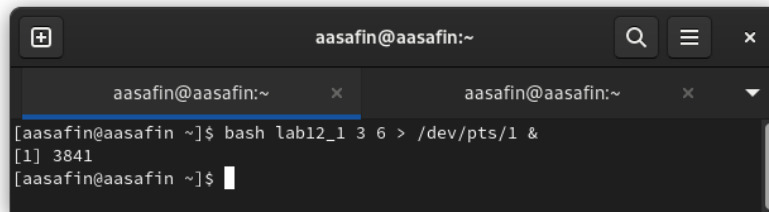
1. Создана программа, выполняющая упрощённый механизм семафоров (рис. 4.1). Программа выполнена (рис. 4.2-4.3).



```
t1=$1
t2=$2
t=0
if ((t<t1))
then echo 'Ожидание'
fi
while ((t<t1))
do
    sleep 1
    let t+=1
done
while ((t<t2))
do
    echo 'Выполнение'
    sleep 1
    let t+=1
done
```

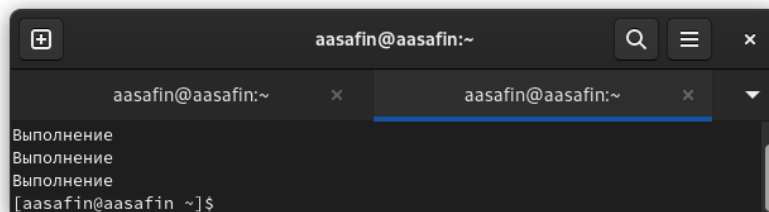
Рис. 4.1: Первая программа





```
aasafin@aasafin:~  
[aasafin@aasafin ~]$ bash lab12_1 3 6 > /dev/pts/1 &  
[1] 3841  
[aasafin@aasafin ~]$
```

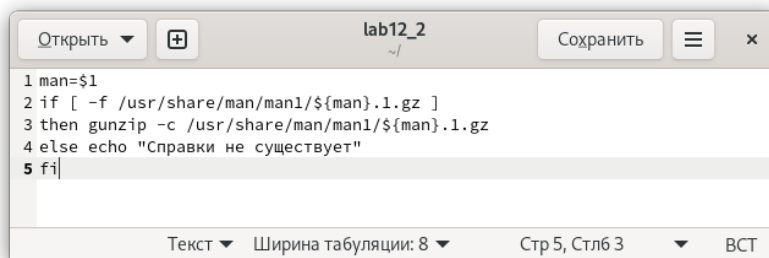
Рис. 4.2: Выполнение первой программы



```
aasafin@aasafin:~  
Выполнение  
Выполнение  
Выполнение  
[aasafin@aasafin ~]$
```

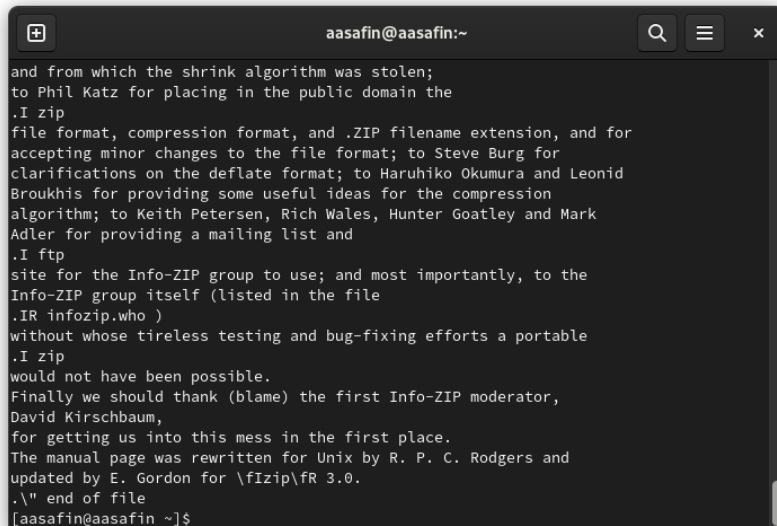
Рис. 4.3: Выполнение первой программы

2. Написана программа, аналогичная команде `man` (рис. 4.4). Программа работает корректно (рис. 4.5).



```
lab12_2  
1 man=$1  
2 if [ -f /usr/share/man/man1/${man}.1.gz ]  
3 then gunzip -c /usr/share/man/man1/${man}.1.gz  
4 else echo "Справки не существует"  
5 fi
```

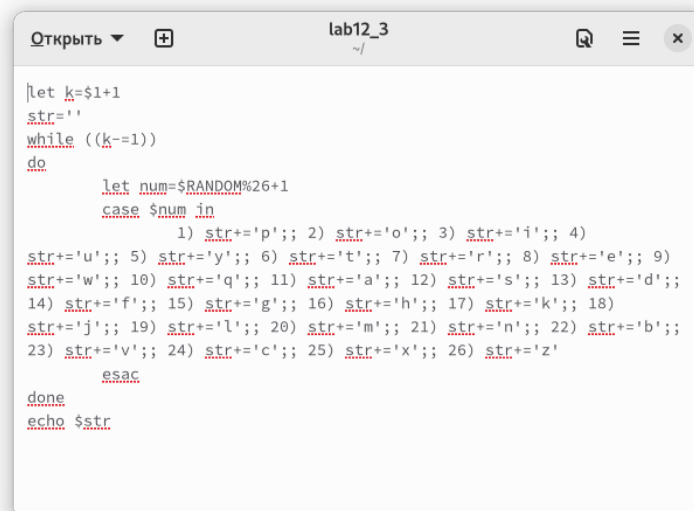
Рис. 4.4: Вторая программа



```
aasafin@aasafin:~  
and from which the shrink algorithm was stolen;  
to Phil Katz for placing in the public domain the  
.I zip  
file format, compression format, and .ZIP filename extension, and for  
accepting minor changes to the file format; to Steve Burg for  
clarifications on the deflate format; to Haruhiko Okumura and Leonid  
Broukhis for providing some useful ideas for the compression  
algorithm; to Keith Petersen, Rich Wales, Hunter Goatley and Mark  
Adler for providing a mailing list and  
.I ftp  
site for the Info-ZIP group to use; and most importantly, to the  
Info-ZIP group itself (listed in the file  
.IR infozip.who )  
without whose tireless testing and bug-fixing efforts a portable  
.I zip  
would not have been possible.  
Finally we should thank (blame) the first Info-ZIP moderator,  
David Kirschbaum,  
for getting us into this mess in the first place.  
The manual page was rewritten for Unix by R. P. C. Rodgers and  
updated by E. Gordon for \fIzip\fR 3.0.  
.\" end of file  
[aasafin@aasafin ~]$
```

Рис. 4.5: Выполнение второй программы

3. Создана программа, создающая случайную последовательность из заданного количества символов (рис. 4.6). Программа выполнена (рис. 4.7).



```
lab12_3  
~/  
Открыть  
let k=$1+1  
str=''   
while ((k-=1))  
do  
    let num=$((RANDOM%26+1))  
    case $num in  
        1) str+='p';; 2) str+='o';; 3) str+='i';; 4)  
str+='u';; 5) str+='y';; 6) str+='t';; 7) str+='r';; 8) str+='e';; 9)  
str+='w';; 10) str+='q';; 11) str+='a';; 12) str+='s';; 13) str+='d';;  
14) str+='f';; 15) str+='g';; 16) str+='h';; 17) str+='k';; 18)  
str+='j';; 19) str+='l';; 20) str+='m';; 21) str+='n';; 22) str+='b';;  
23) str+='v';; 24) str+='c';; 25) str+='x';; 26) str+='z'  
    esac  
done  
echo $str
```

Рис. 4.6: Третья программа

A terminal window with a dark background. The title bar shows 'aasafin@aasafin:~' and standard window controls. The terminal content shows a user prompt '[aasafin@aasafin ~]\$' followed by the command 'bash lab12\_3 10'. The output is 'twmkascuap'. The prompt returns to '[aasafin@aasafin ~]\$'.

```
aasafin@aasafin:~  
[aasafin@aasafin ~]$ bash lab12_3 10  
twmkascuap  
[aasafin@aasafin ~]$
```

Рис. 4.7: Выполнение третьей программы

## 5 Выводы

Описанные командные файлы созданы, навык работы с bash получен.