

# **Отчёт по лабораторной работе №13**

Серегин Денис Алексеевич

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>5</b>
<b>2</b>	<b>Задания</b>	<b>6</b>
<b>3</b>	<b>Выполнение лабораторной работы</b>	<b>8</b>
<b>4</b>	<b>Вывод</b>	<b>13</b>
<b>5</b>	<b>Использованные ресурсы</b>	<b>14</b>

# List of Tables

## List of Figures

3.1	Код файл . . . . .	9
3.2	Вывод в первом терминале . . . . .	10
3.3	Вывод во втором терминале . . . . .	10
3.4	Код файла . . . . .	11
3.5	Результат работы . . . . .	11
3.6	Справка о команде mkdir . . . . .	11
3.7	Код файла . . . . .	12
3.8	Результат работы исполняемого файла . . . . .	12

# 1 Цель работы

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

## 2 Задания

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

1. Написал 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.

(рис 1. 3.1)



```
#!/bin/bash
function func1
{
touch lr13.txt
ls>lr13.txt
for((i=0;i<5;i++))
do
echo "скрипт работает"
sleep 2
done
rm lr13.txt
}
function func2
{
while test -f lr13.txt
do
echo "занято другим процессом"
sleep 2
done
}
func2
func1
```

Figure 3.1: Код файл

(рис 2. 3.2)

```

[root@localhost daseregim]# tty
/dev/pts/1
[root@localhost daseregim]# ./lab13.sh
скрипт работает
занято другим процессом
скрипт работает
занято другим процессом
скрипт работает
занято другим процессом
скрипт работает
занято другим процессом
скрипт работает
занято другим процессом
[root@localhost daseregim]# скрипт работает
скрипт работает
скрипт работает

```

Figure 3.2: Вывод в первом терминале

(рис 3. 3.3)

```

[root@localhost daseregim]# ./lab13.sh>/dev/pts/1 &
[1] 5416
[root@localhost daseregim]# █

```

Figure 3.3: Вывод во втором терминале

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`.

(рис 4. 3.4)

```

daseregin@localhost:/home/daseregin
Файл  Правка  Вид  Поиск  Терминал  Справка
#!/bin/bash
cd /usr/share/man/man1
if test -f $1.1.gz
then less $1.1.gz
else echo "отсутствует информация о команде" $1
fi

```

Figure 3.4: Код файла

(рис 5. 3.5)

```

[root@localhost daseregin]# ./lab13_2.sh mkdir
[root@localhost daseregin]# ./lab13_2.sh mkdirrrr
отсутствует информация о команде mkdirrrr
[root@localhost daseregin]# █

```

Figure 3.5: Результат работы

(рис 6. 3.6)

```

MKDIR(1)                                User Commands                                MKDIR(1)

ESC[1mNAMEESC[0m
mkdir - make directories

ESC[1mSYNOPSISESC[0m
ESC[1mmkdir ESC[22m[ESC[4mOPTIONESC[24m]... ESC[4mDIRECTORYESC[24m]...

ESC[1mDESCRIPTIONESC[0m
Create the DIRECTORY(ies), if they do not already exist.

Mandatory arguments to long options are mandatory for short options
too.

ESC[1m-mESC[22m, ESC[1m--modeESC[22m=ESC[4mMODEESC[0m
set file mode (as in chmod), not a=rwx - umask

ESC[1m-pESC[22m, ESC[1m--parentsESC[0m
no error if existing, make parent directories as needed

ESC[1m-vESC[22m, ESC[1m--verboseESC[0m
print a message for each created directory

ESC[1m-Z ESC[22mset SELinux security context of each created directory to the
default type

ESC[1m--contextESC[22m=ESC[4mCTXESC[24m]
like ESC[1m-ZESC[22m, or if CTX is specified then set the SELinux or SMACK
security context to CTX

ESC[1m--help ESC[22mdisplay this help and exit

ESC[1m--versionESC[0m
output version information and exit

GNU coreutils online help: <http://www.gnu.org/software/coreutils/>
Report mkdir translation bugs to <http://translationproject.org/team/>

ESC[1mAUTHORESC[0m
Written by David MacKenzie.

mkdir.1.gz

```

Figure 3.6: Справка о команде mkdir

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. В файл lr13.txt записал все латинские буквы.

(рис 7. 3.7)

```
#!/bin/bash
for ((i=0;i<$1;i++))
do
let N=$RANDOM%52+1
cat ./lr13.txt|tail -n +$N|head -n 1|tr -d '\n'
done
echo
```

Figure 3.7: Код файла

(рис 8. 3.8)

```
[root@localhost daseregin]# touch lr13.txt
[root@localhost daseregin]# vi lr13.txt
[root@localhost daseregin]# ./lab13_3.sh 1

[root@localhost daseregin]# ./lab13_3.sh 33

[root@localhost daseregin]# vi lab13_3.sh
[root@localhost daseregin]# chmod +x ./lab13_3.sh
[root@localhost daseregin]# ./lab13_3.sh 33
QWERTYUIOPASDFGHJKLZXCVBNMqwertyuiopasdfghjklzxcvbnm
[root@localhost daseregin]# ./lab13_3.sh 1

[root@localhost daseregin]# ./lab13_3.sh 88
QWERTYUIOPASDFGHJKLZXCVBNMqwertyuiopasdfghjklzxcvbnm
```

Figure 3.8: Результат работы исполняемого файла

## **4 Вывод**

В ходе лабораторной работы мне удалось изучить основы программирования в оболочке ОС UNIX, а также научиться писать более сложные командные файлы с использованием логических управляющих конструкций и циклов.

## **5 Используемые ресурсы**

<https://www.celitel.info/klad/nhelp/helpbat.php?dcmd=mkdir>