

Отчёт по лабораторной работе № 13

Перелыгин Сергей Викторович

Содержание

1 Цель работы	4
2 Выполнение лабораторной работы	5
3 Ответы на контрольные вопросы	14
4 Выводы	17
5 Библиография	18

Список иллюстраций

2.1 Скрипт 1	6
2.2 Проверка скрипта 1	6
2.3 Скрипт 2	7
2.4 Скрипт 2	8
2.5 Проверка скрипта 2	8
2.6 Проверка скрипта 2	9
2.7 Проверка скрипта 2	9
2.8 Переход в каталог /usr/share/man/man1	10
2.9 Изучаем содержимое каталога	10
2.10 Скрипт 3	11
2.11 Проверка скрипта 3	11
2.12 Проверка скрипта 3	12
2.13 Скрипт 4	13
2.14 Скрипт 4	13

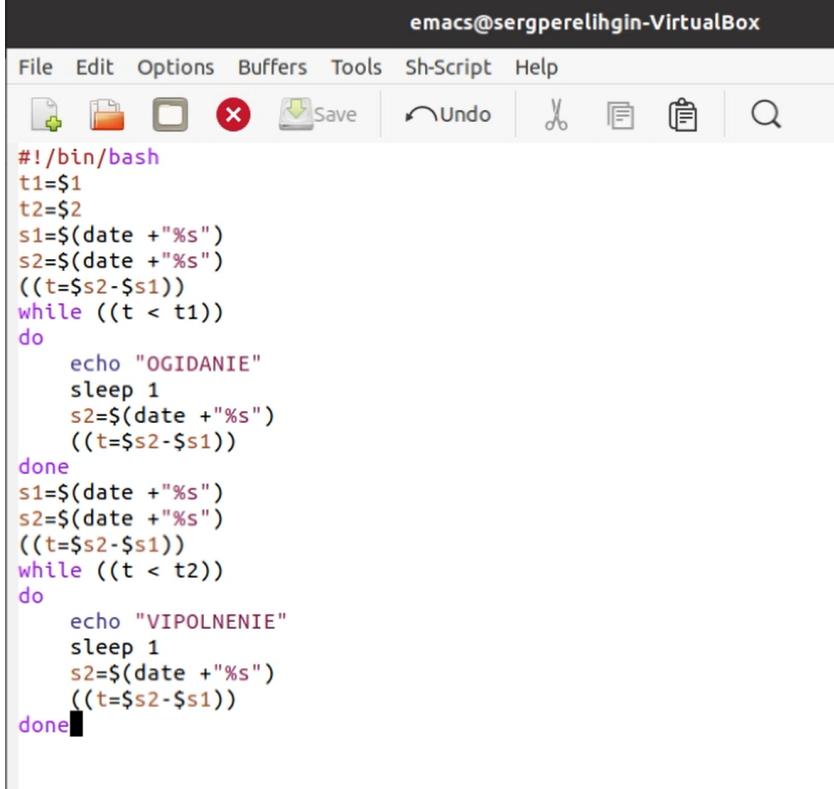
1 Цель работы

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

2 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). Для данной задачи я создал файл: ex1.sh (Рисунок 1) и написал соответствующий скрипт.



```
emacs@sergperelihgin-VirtualBox
```

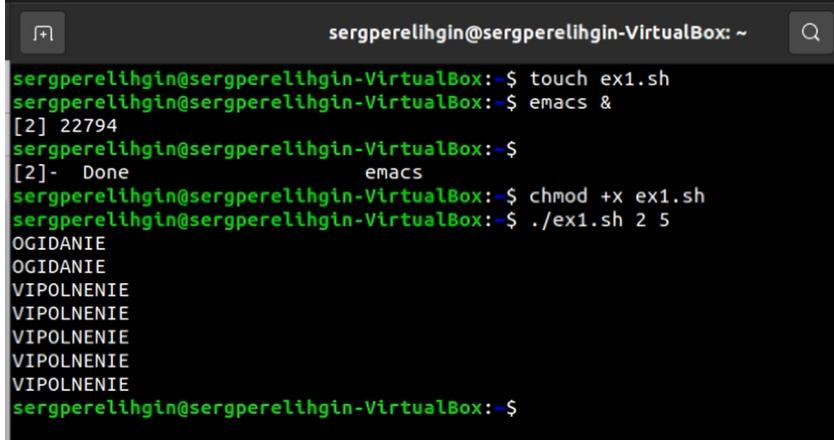
File Edit Options Buffers Tools Sh-Script Help

Save Undo

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

Рис. 2.1: Скрипт 1

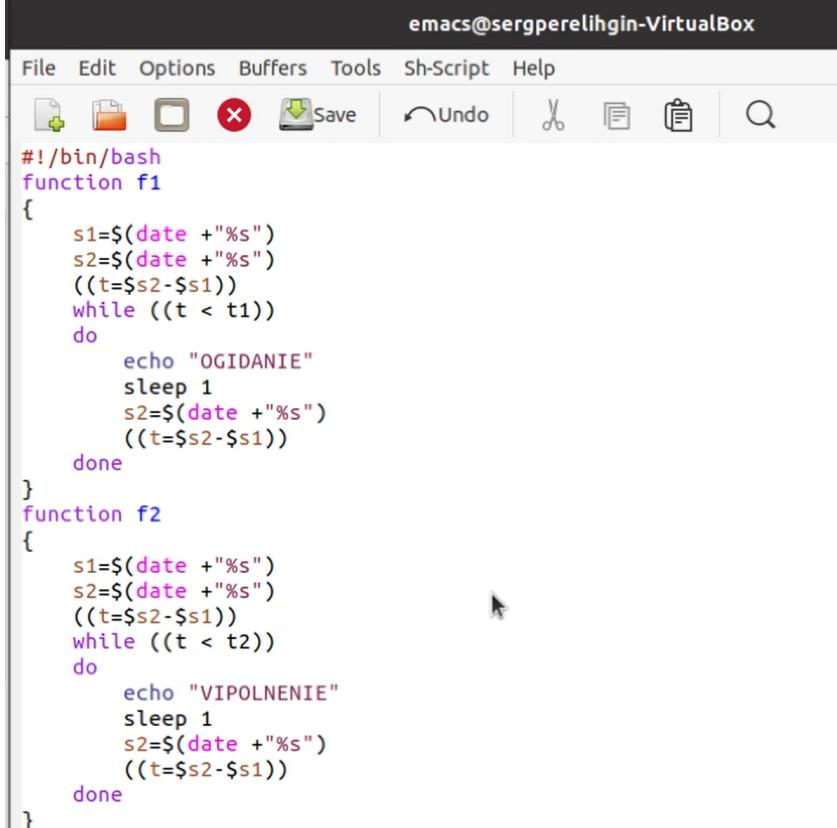
Далее я проверил работу написанного скрипта (команда «./ex1.sh 2 5»), предварительно добавив право на исполнение файла (команда «chmod +x ex1.sh») (Рисунок 2). Скрипт работает корректно.



```
sergperelihgin@sergperelihgin-VirtualBox:~$ touch ex1.sh
sergperelihgin@sergperelihgin-VirtualBox:~$ emacs &
[2] 22794
sergperelihgin@sergperelihgin-VirtualBox:~$ 
[2]- Done                  emacs
sergperelihgin@sergperelihgin-VirtualBox:~$ chmod +x ex1.sh
sergperelihgin@sergperelihgin-VirtualBox:~$ ./ex1.sh 2 5
OGIDANIE
OGIDANIE
VIPOLNENIE
VIPOLNENIE
VIPOLNENIE
VIPOLNENIE
VIPOLNENIE
sergperelihgin@sergperelihgin-VirtualBox:~$
```

Рис. 2.2: Проверка скрипта 1

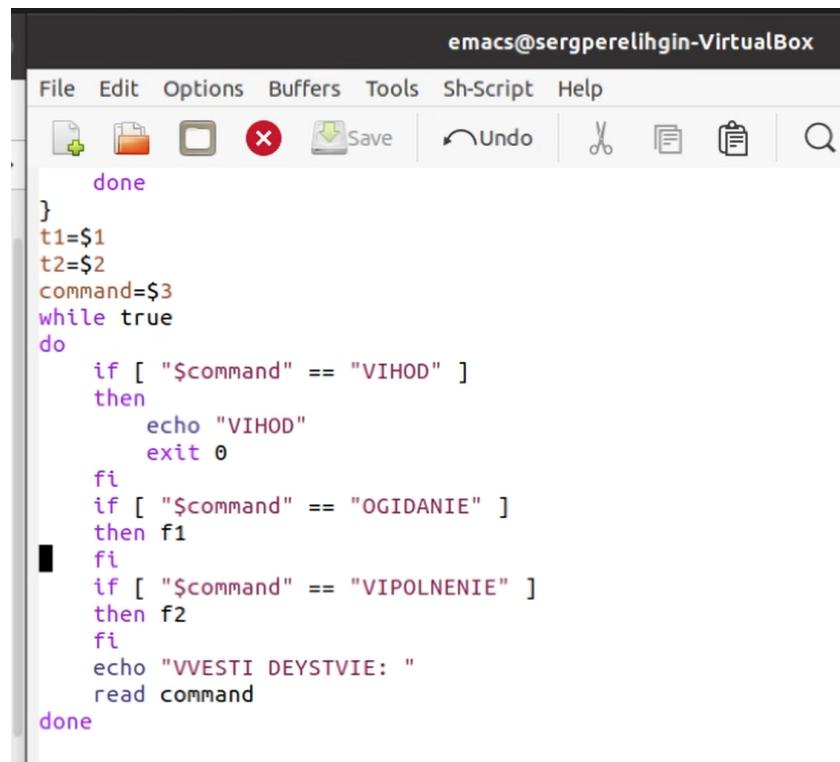
После этого я изменил скрипт так, чтобы его можно было выполнять в нескольких терминалах и проверил его работу (Рисунки 3, 4, 5, 6, 7). К сожалению при вводе любых команд, не получилось достичь желаемого результата, выводится сообщение “Permission denied”, то есть отказ в доступе.



The screenshot shows an Emacs window titled "emacs@sergperelihgin-VirtualBox". The buffer contains a shell script with syntax highlighting. The script defines two functions, f1 and f2, which calculate the time difference between two dates and then loop until that time has passed, printing either "OGIDANIE" or "VIPOLNENIE" to the console. The script uses the date command to get the current timestamp and sleep to wait for one second at a time.

```
#!/bin/bash
function f1
{
    s1=$(date +%s)
    s2=$(date +%s)
    ((t=$s2-$s1))
    while ((t < t1))
    do
        echo "OGIDANIE"
        sleep 1
        s2=$(date +%s)
        ((t=$s2-$s1))
    done
}
function f2
{
    s1=$(date +%s)
    s2=$(date +%s)
    ((t=$s2-$s1))
    while ((t < t2))
    do
        echo "VIPOLNENIE"
        sleep 1
        s2=$(date +%s)
        ((t=$s2-$s1))
    done
}
```

Рис. 2.3: Скрипт 2

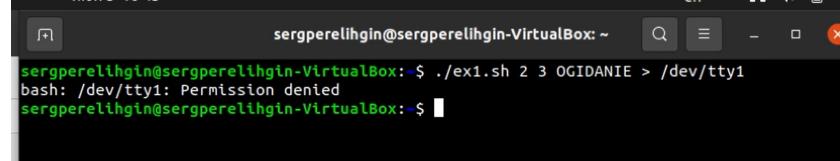


emacs@sergperelihgin-VirtualBox

File Edit Options Buffers Tools Sh-Script Help

done
}
t1=\$1
t2=\$2
command=\$3
while true
do
 if ["\$command" == "VIHOD"]
 then
 echo "VIHOD"
 exit 0
 fi
 if ["\$command" == "OGIDANIE"]
 then f1
 fi
 if ["\$command" == "VIPOLNENIE"]
 then f2
 fi
 echo "VVESTI DEYSTVIE: "
 read command
done

Рис. 2.4: Скрипт 2



sergperelihgin@sergperelihgin-VirtualBox: ~

```
sergperelihgin@sergperelihgin-VirtualBox: $ ./ex1.sh 2 3 OGIDANIE > /dev/tty1
bash: /dev/tty1: Permission denied
sergperelihgin@sergperelihgin-VirtualBox: ~$
```

Рис. 2.5: Проверка скрипта 2

```
sergperelihgin@sergperelihgin-VirtualBox: ~
sergperelihgin@sergperelihgin-VirtualBox: $ ./ex1.sh 2 3 OGIDANIE > /dev/tty1
bash: /dev/tty1: Permission denied
sergperelihgin@sergperelihgin-VirtualBox: ~$ ./ex1.sh 2 3 OGIDANIE | sudo tee -a /dev/tt
y1
bash: ./ex1.sh: No such file or directory
[sudo] password for sergperelihgin:
sergperelihgin@sergperelihgin-VirtualBox: ~
sergperelihgin@sergperelihgin-VirtualBox: ~$ ./ex1.sh 2 3 OGIDANIE | sudo tee -a /dev/tt
y1
OGIDANIE
OGIDANIE
VVESTI DEYSTVIE:
VIHOD
VIHOD
sergperelihgin@sergperelihgin-VirtualBox: ~$ ./ex1.sh 2 3 OGIDANIE | sudo tee -a /dev/tt
y1
OGIDANIE
OGIDANIE
VVESTI DEYSTVIE:
OGIDANIE
OGIDANIE
OGIDANIE
OGIDANIE
VVESTI DEYSTVIE:
VIPOLNENIE
VIPOLNENIE
VIPOLNENIE
VIPOLNENIE
VVESTI DEYSTVIE:
VIHOD
VIHOD
sergperelihgin@sergperelihgin-VirtualBox: ~$
```

Рис. 2.6: Проверка скрипта 2

```
[1]+ 42992* bash: /dev/pts/1: Permission denied
\_
[1]+ Exit 1 ./ex1.sh 2 3 OGIDANIE > /dev/pts/1
sergperelihgin@sergperelihgin-VirtualBox: ~$ ./ex1.sh 2 3 OGIDANIE | sudo tee -a /dev/pts/1 &
[1] 23057
sergperelihgin@sergperelihgin-VirtualBox: ~$ VIHOD
VIHOD: command not found

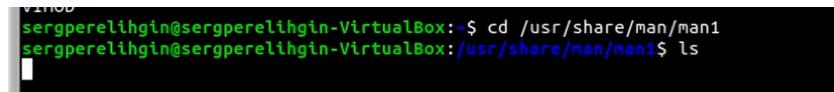
[2]+ Stopped ./ex1.sh 2 3 OGIDANIE | sudo tee -a /dev/pts/2
sergperelihgin@sergperelihgin-VirtualBox: ~$ ./ex1.sh 2 3 OGIDANIE | sudo tee -a /dev/pts/2 &
[2] 23062
sergperelihgin@sergperelihgin-VirtualBox: ~$ OGIDANIE
OGIDANIE: command not found

[3]+ Stopped ./ex1.sh 2 3 | sudo tee -a /dev/pts/2
sergperelihgin@sergperelihgin-VirtualBox: ~$
```

Рис. 2.7: Проверка скрипта 2

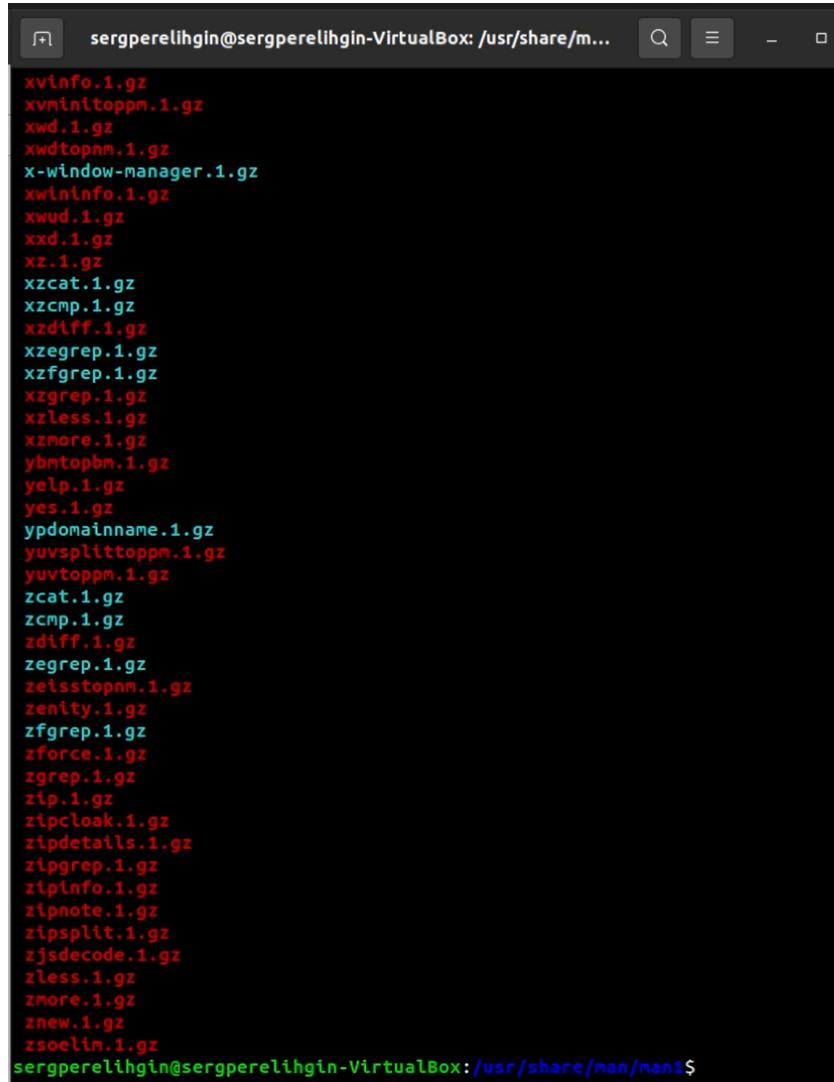
2. Реализовал команду man с помощью командного файла. Изучил содержимое каталога /usr/share/man/man1 (Рисунок 8). В нем находятся архивы текстовых файлов, содержащих справку по большинству установленных в системе программ и команд. Каждый архив можно открыть командой less сразу же просмотрев содержимое 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.



```
VINOD
sergperelihgin@sergperelihgin-VirtualBox:~$ cd /usr/share/man/man1
sergperelihgin@sergperelihgin-VirtualBox:/usr/share/man/man1$ ls
```

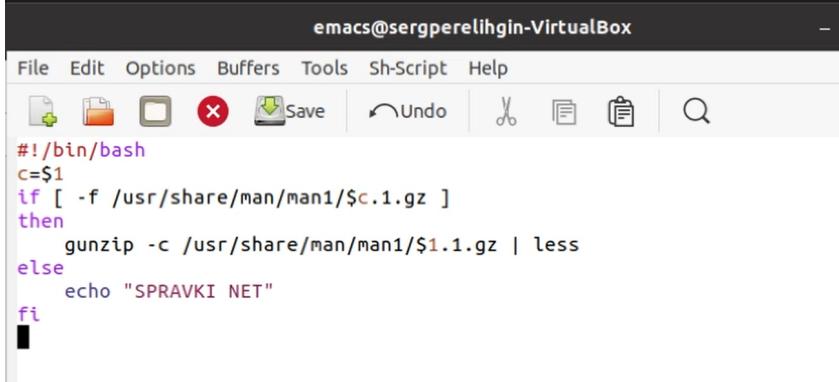
Рис. 2.8: Переход в каталог /usr/share/man/man1



```
VINOD
sergperelihgin@sergperelihgin-VirtualBox:/usr/share/m... Q - □
xvinfo.1.gz
xvminitoppm.1.gz
xd.1.gz
xwdtopnm.1.gz
x-window-manager.1.gz
xwininfo.1.gz
xxd.1.gz
xxd.1.gz
xzcat.1.gz
xzcmp.1.gz
xxdiff.1.gz
xzgrep.1.gz
xzfgrep.1.gz
xzgrep.1.gz
xzless.1.gz
xzmore.1.gz
ybmtopbm.1.gz
yelp.1.gz
yes.1.gz
ypdomainname.1.gz
yuvsplitppm.1.gz
yuvtoppm.1.gz
zcat.1.gz
zcmp.1.gz
zdiff.1.gz
zegrep.1.gz
zelsstopnm.1.gz
zenity.1.gz
zfgrep.1.gz
zforce.1.gz
zgrep.1.gz
zip.1.gz
zipcloak.1.gz
zipdetails.1.gz
zipgrep.1.gz
zipinfo.1.gz
zipnote.1.gz
zipsplit.1.gz
zjsdecode.1.gz
zless.1.gz
zmore.1.gz
znew.1.gz
zsoelim.1.gz
sergperelihgin@sergperelihgin-VirtualBox:/usr/share/man/man1$
```

Рис. 2.9: Изучаем содержимое каталога

Для данной задачи я создал файл: ex2.sh (Рисунок 10) и написал соответствующий скрипт.



```
emacs@sergperelihgın-VirtualBox
```

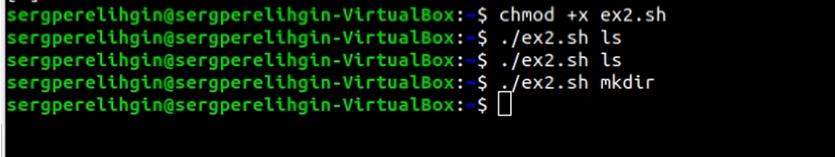
File Edit Options Buffers Tools Sh-Script Help

Save Undo Cut Copy Find

```
#!/bin/bash
c=$1
if [ -f /usr/share/man/man1/$c.1.gz ]
then
    gunzip -c /usr/share/man/man1/$c.1.gz | less
else
    echo "SPRAVKI NET"
fi
```

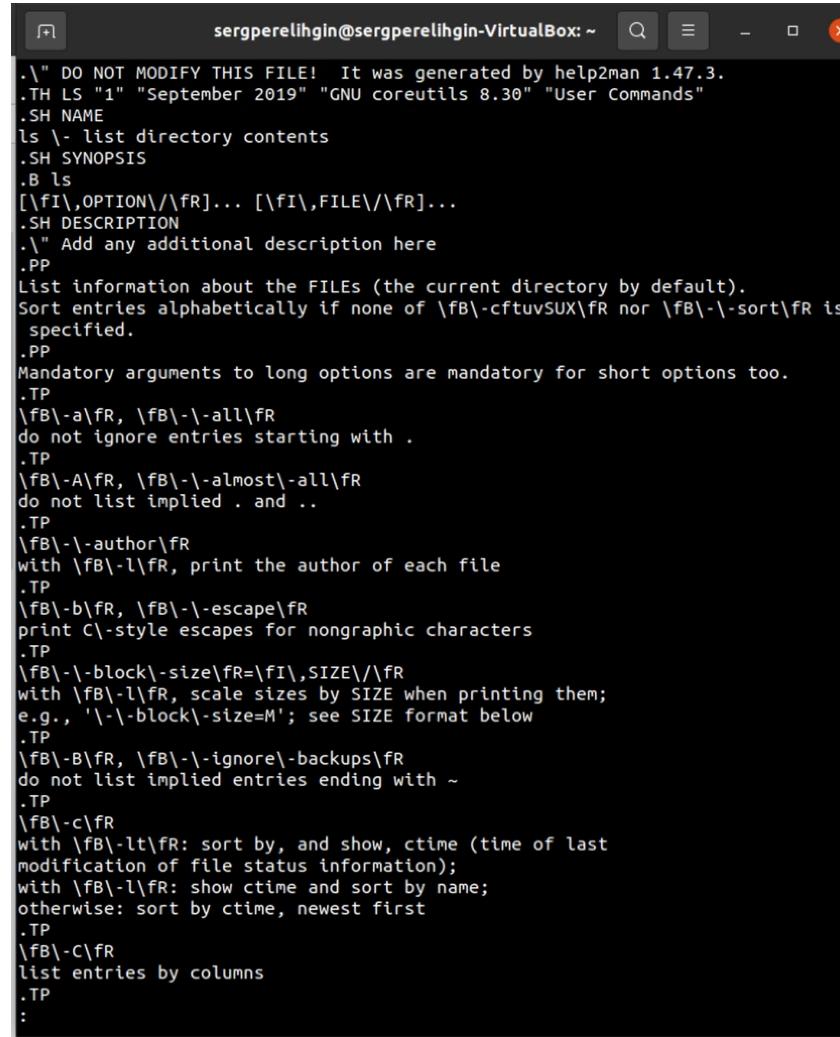
Рис. 2.10: Скрипт 3

Далее я проверил работу написанного скрипта (команды «./ex2.sh ls» и «./ex2.sh mkdir»), предварительно добавив право на исполнение файла (команда «chmod +x ex2.sh») (Рисунки 11, 12). Скрипт работает корректно.



```
sergperehgin@sergperelihgın-VirtualBox:~$ chmod +x ex2.sh
sergperehgin@sergperelihgın-VirtualBox:~$ ./ex2.sh ls
sergperehgin@sergperelihgın-VirtualBox:~$ ./ex2.sh ls
sergperehgin@sergperelihgın-VirtualBox:~$ ./ex2.sh mkdir
sergperehgin@sergperelihgın-VirtualBox:~$ 
```

Рис. 2.11: Проверка скрипта 3

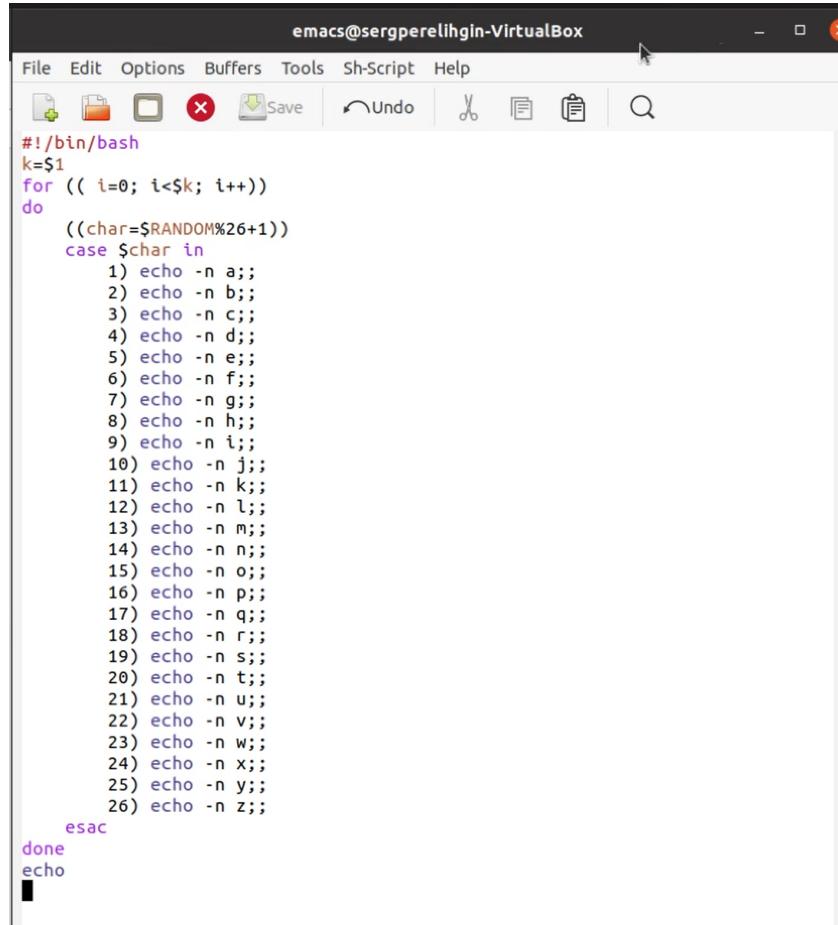


The screenshot shows a terminal window with the title bar "sergperelihgin@sergperelihgin-VirtualBox: ~". The window contains the man page for the "ls" command. The text is as follows:

```
." DO NOT MODIFY THIS FILE! It was generated by help2man 1.47.3.
.TH LS "1" "September 2019" "GNU coreutils 8.30" "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
do not ignore entries starting with .
.TP
\fB\-\A\fR, \fB\-\-\almost\-\all\fR
do not list implied . and ..
.TP
\fB\-\-\author\fR
with \fB\-\l\fR, print the author of each file
.TP
\fB\-\b\fR, \fB\-\-escape\fR
print C\ -style escapes for nongraphic characters
.TP
\fB\-\-\block\fR=\fI\,SIZE\fR
with \fB\-\l\fR, scale sizes by SIZE when printing them;
e.g., '\-\block=10'; see SIZE format below
.TP
\fB\-\B\fR, \fB\-\-\ignore\-\backups\fR
do not list implied entries ending with ~
.TP
\fB\-\c\fR
with \fB\-\lt\fR: sort by, and show, ctime (time of last
modification of file status information);
with \fB\-\l\fR: show ctime and sort by name;
otherwise: sort by ctime, newest first
.TP
\fB\-\C\fR
list entries by columns
.TP
:
```

Рис. 2.12: Проверка скрипта 3

3. Используя встроенную переменную \$RANDOM, написал командный файл, генерирующий случайную последовательность букв латинского алфавита. Для данной задачи я создал файл: ex3.sh (Рисунок 13) и написал соответствующий скрипт.



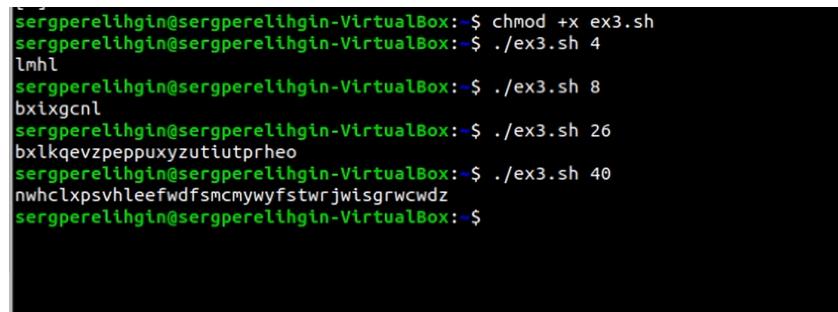
```
emacs@sergperelihgin-VirtualBox ~
```

```
File Edit Options Buffers Tools Sh-Script Help
```

```
#! /bin/bash
k=$1
for (( i=0; i<$k; i++))
do
    ((char=$RANDOM%26+1))
    case $char 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
```

Рис. 2.13: Скрипт 4

Далее я проверил работу написанного скрипта, предварительно добавив право на исполнение файла (Рисунок 14). Скрипт работает корректно.



```
sergperelihgin@sergperelihgin-VirtualBox:~$ chmod +x ex3.sh
sergperelihgin@sergperelihgin-VirtualBox:~$ ./ex3.sh 8
lmhl
sergperelihgin@sergperelihgin-VirtualBox:~$ ./ex3.sh 26
bxixgcnl
sergperelihgin@sergperelihgin-VirtualBox:~$ ./ex3.sh 40
bxlkqevzpeppuxyzututprheo
sergperelihgin@sergperelihgin-VirtualBox:~$ ./ex3.sh 40
nwhclxpsvhleefwdfsmcmwyfstwrjwigrwcwdz
sergperelihgin@sergperelihgin-VirtualBox:~$
```

Рис. 2.14: Скрипт 4

3 Ответы на контрольные вопросы

1) while [\$1 != "exit"]

В данной строчке допущены следующие ошибки:

- не хватает пробелов после первой скобки [и перед второй скобкой]
- выражение \$1 необходимо взять в "", потому что эта переменная может содержать пробелы.

Таким образом, правильный вариант должен выглядеть так: while ["\$1"!= "exit"]

2) Чтобы объединить несколько строк в одну, можно воспользоваться некоторыми способами:

- Первый:

```
VAR1="Hello,  
"VAR2=" World"  
VAR3="VAR1VAR2"  
echo "$VAR3"
```

Результат: Hello, World

- Второй:

```
VAR1="Hello,"  
VAR1+=" World"  
echo "$VAR1"
```

Результат: Hello, World

3) Команда seq в Linux используется для генерации чисел от ПЕРВОГО до ПОСЛЕДНЕГО шага INCREMENT.

Параметры:

- seq LAST: если задан только один аргумент, он создает числа от 1 до LAST с шагом шага, равным 1. Если LAST меньше 1, значение is не выдает.
 - seq FIRST LAST: когда заданы два аргумента, он генерирует числа от FIRST до LAST с шагом 1, равным 1. Если LAST меньше FIRST, он не выдает никаких выходных данных.
 - seq FIRST INCREMENT LAST: когда заданы три аргумента, он генерирует числа от FIRST до LAST на шаге INCREMENT . Если LAST меньше, чем FIRST, он не производит вывод.
 - seq -f «FORMAT» FIRST INCREMENT LAST: эта команда используется для генерации последовательности в форматированном виде. FIRST и INCREMENT являются необязательными.
 - seq -s «STRING» ПЕРВЫЙ ВКЛЮЧЕНО: Эта команда используется для STRING для разделения чисел. По умолчанию это значение равно /n. FIRST и INCREMENT являются необязательными.
 - seq -w FIRST INCREMENT LAST: эта команда используется для выравнивания ширины путем заполнения начальными нулями. FIRST и INCREMENT являются необязательными.
- 4) Результатом данного выражения $\$(10/3)$ будет 3, потому что это целочисленное деление без остатка.
- 5) Отличия командной оболочки zshot bash:
- В zsh более быстрое автодополнение для cdc помощью Tab

- В zsh существует калькулятор zcalc, способный выполнять вычисления внутри терминала
 - В zsh поддерживаются числа с плавающей запятой
 - В zsh поддерживаются структуры данных «хэш»
 - В zsh поддерживается раскрытие полного пути на основе неполных данных
 - В zsh поддерживается замена частичного пути
 - В zsh есть возможность отображать разделенный экран, такой же как разделенный экран vim
- 6) `for((a=1; a<= LIMIT; a++))` синтаксис данной конструкции верен, потому что, используя двойные круглые скобки, можно не писать \$ перед переменными () .

7) Преимущества скриптового языка bash:

- Один из самых распространенных и ставится по умолчанию в большинстве дистрибутивах Linux, MacOS
- Удобное перенаправление ввода/вывода
- Большое количество команд для работы с файловыми системами Linux
- Можно писать собственные скрипты, упрощающие работу в Linux

Недостатки скриптового языка bash:

- Дополнительные библиотеки других языков позволяют выполнить больше действий
- Bash не является языком общего назначения
- Утилиты, при выполнении скрипта, запускают свои процессы, которые, в свою очередь, отражаются на быстроте выполнения этого скрипта
- Скрипты, написанные на bash, нельзя запустить на других операционных системах без дополнительных действий.

4 Выводы

В ходе выполнения данной лабораторной работы я изучил основы программирования в оболочке ОС UNIX, а также научился писать более сложные командные файлы с использованием логических управляющих конструкций и циклов.

5 Библиография

- Кулябов Д.С. Операционные системы: лабораторные работы: учебное пособие / Д.С. Кулябов, М.Н. Геворкян, А.В. Королькова, А.В. Демидова. — М. : Изд-во РУДН, 2016. — 117 с. — ISBN 978-5-209-07626-1 : 139.13; То же [Электронный ресурс]. — URL: <http://lib.rudn.ru/MegaPro2/Download/MObject/6118>.
- Робачевский А.М. Операционная система UNIX [текст] : Учебное пособие / А.М. Робачевский, С.А. Немлюгин, О.Л. Стесик. — 2-е изд., перераб. и доп. — СПб. : БХВ-Петербург, 2005, 2010. — 656 с. : ил. — ISBN 5-94157-538-6 : 164.56. (ET 60)
- Таненбаум Эндрю. Современные операционные системы [Текст] / Э. Таненбаум. — 2-е изд. — СПб. : Питер, 2006. — 1038 с. : ил. — (Классика Computer Science). — ISBN 5-318-00299-4 : 446.05. (ET 50)
- Ван Стеен М., Эндрю Таненбаум Распределенные системы. Принципы и парадигмы [Текст] / Э. Таненбаум, в.М. Стеен. — СПб. : Питер, 2003. — 877 с. : ил. — (Классика Computer science). — ISBN 5-272-00053-6 : 377.52. (ET 50)
- Сафонов, В.О. Основы современных операционных систем : учебное пособие / В.О. Сафонов. — Москва : Интернет-Университет Информационных Технологий, 2011. — 584 с. — (Основы информационных технологий). — ISBN 978-5-9963-0495-0 ; То же [Электронный ресурс]. — URL: <http://biblioclub.ru/index.php?page=book&id=233210>.
- Немет Эви. UNIX — руководство системного администратора [Текст] / Э. Немет, Г. Снайдер, С. Сибасс; Э.Немет, Г.Снайдер, С.Сибасс, Х.Р.Трент. — 3-е изд. — СПб. : Питер, 2004. — 925 с. : ил. — (Для профессионалов). — ISBN

0-13-020601-6. — ISBN 5-318-00754-6 : 280.00. (ET 30)

- Бек Л. Введение в системное программирование [Текст] / Л. Бек; Пер. с англ. Н.А.Богомолова, В.М.Вязовского и С.Е.Морковина; Под ред. Л.Н.Королева. — М. : Мир, 1988. — 448 с. : ил. — ISBN 5-03-000011-9 : 2.60. (ET 3)
- Дьяконов Владимир Юрьевич. Системное программирование [Текст] : Учебное пособие для втузов / В.Ю. Дьяконов, В.А. Китов, И.А. Калинчев; Под ред. А.Л.Горелика. — М. : Высшая школа, 1990. — 221 с. : ил. — ISBN 5-06-000732-4 : 0.55.