

# Laboratory №13

---

Kulikov Maksim

MAY-2021

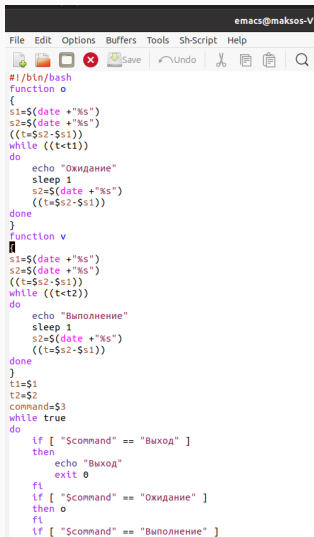
RUDN University, Moscow, Russian Federation

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

1. Ознакомиться с теоретическим материалом.
2. Выполнить работу.

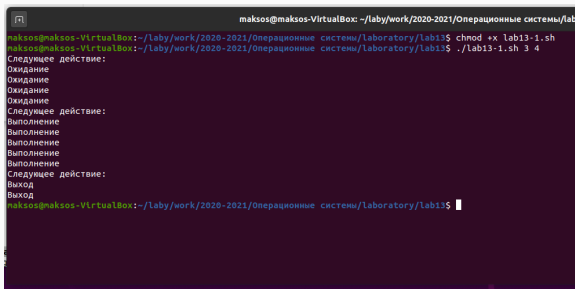
# Выполнение лабораторной работы

1. Создаю файл с расширением “sh”, в котором пишу скрипт, который является упрощенным механизмом семафоров (рис. -fig. 1)



```
emacs@maksos-V
File Edit Options Buffers Tools Sh-Script Help
Save Undo
#!/bin/bash
function o
{
s1=$(date +%s")
s2=$(date +%s")
((t=s2-s1))
while ((t<t1))
do
echo "Ожидание"
sleep 1
s2=$(date +%s")
((t=s2-s1))
done
}
function v
{
s1=$(date +%s")
s2=$(date +%s")
((t=s2-s1))
while ((t<t2))
do
echo "Выполнение"
sleep 1
s2=$(date +%s")
((t=s2-s1))
done
}
t1=$1
t2=$2
command=$3
while true
do
if [ "$command" == "Выход" ]
then
echo "Выход"
exit 0
fi
if [ "$command" == "Ожидание" ]
then
o
fi
if [ "$command" == "Выполнение" ]
```

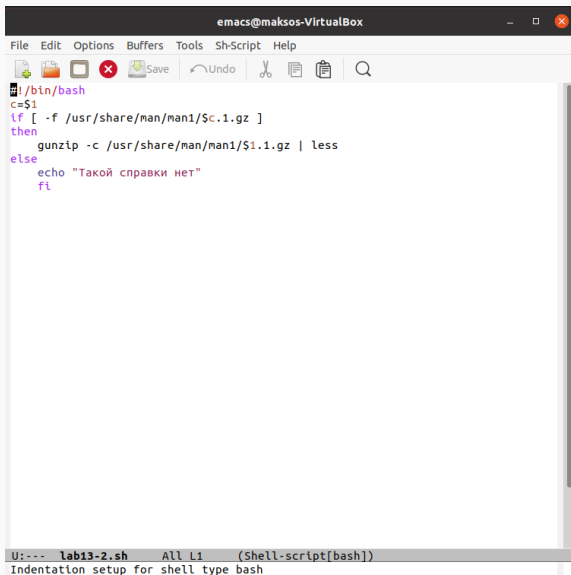
## 2. Тест скрипта. Работает исправно (рис. -fig. 2)



```
maksos@maksos-VirtualBox: ~/lab/work/2020-2021/Операционные системы/lab
maksos@maksos-VirtualBox:~/lab/work/2020-2021/Операционные системы/laboratory/lab13$ chmod +x lab13-1.sh
maksos@maksos-VirtualBox:~/lab/work/2020-2021/Операционные системы/laboratory/lab13$ ./lab13-1.sh 3 4
Следующее действие:
Ожидание
Ожидание
Ожидание
Ожидание
Следующее действие:
Выполнение
Выполнение
Выполнение
Выполнение
Следующее действие:
Выход
Выход
maksos@maksos-VirtualBox:~/lab/work/2020-2021/Операционные системы/laboratory/lab13$
```

Рис. 2: Тест №1

3. Создаю файл с расширением “sh”, в котором пишу скрипт, который реализовывает команду man (рис. -fig. 3)

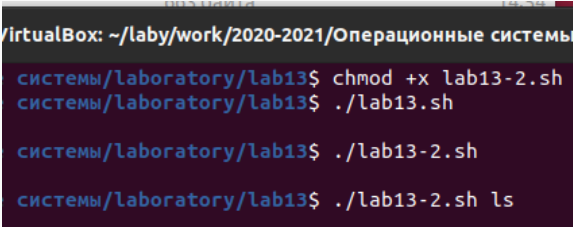


```
emacs@maksos-VirtualBox
File Edit Options Buffers Tools Sh-Script Help
Save Undo Cut Copy Paste Find
# /bin/bash
c=$1
if [ -f /usr/share/man/man1/$c.1.gz ]
then
  gunzip -c /usr/share/man/man1/$1.1.gz | less
else
  echo "Такой справки нет"
fi

U:--- lab13-2.sh All L1 (Shell-script[bash])
Indentation setup for shell type bash
```

Рис. 3: Второй скрипт

4. Запускаю скрипт. Он должен должен выдать справку об команде “ls” (рис. -fig. 4)

A screenshot of a terminal window with a dark background and light-colored text. The prompt is 'VirtualBox: ~/laby/work/2020-2021/Операционные системы'. The user enters four commands: 'chmod +x lab13-2.sh', './lab13.sh', './lab13-2.sh', and './lab13-2.sh ls'.

```
VirtualBox: ~/laby/work/2020-2021/Операционные системы
системы/laboratory/lab13$ chmod +x lab13-2.sh
системы/laboratory/lab13$ ./lab13.sh

системы/laboratory/lab13$ ./lab13-2.sh

системы/laboratory/lab13$ ./lab13-2.sh ls
```

Рис. 4: Запуск скрипта

5. Скрипт работает исправно (рис. -fig. 5)

```

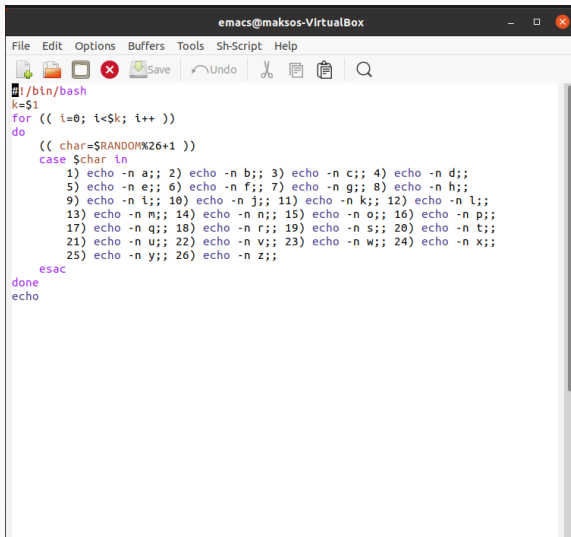
makos@maksos-VirtualBox: ~/laby/work/2020-2021/Операционные системы/lab1
.TP
\fb\z\fr, \fb\l-context\fr
print any security context of each file
.TP
\fb\l\fr
list one file per line. Avoid 'len' with \fb\q\fr or \fb\b\fr
.TP
\fb\l-help\fr
display this help and exit
.TP
\fb\l-version\fr
output version information and exit
.PP
The SIZE argument is an integer and optional unit (example: 10K is 10*1024).
Units are K,M,G,T,P,E,Z,Y (powers of 1024) or KB,MB,... (powers of 1000).
.PP
The TIME_STYLE argument can be full-iso, long-iso, iso, locale, or +FORMAT.
FORMAT is interpreted like in date(1). If FORMAT is FORMAT1<newline>FORMAT2,
then FORMAT1 applies to non-recent files and FORMAT2 to recent files.
TIME_STYLE prefixed with 'posix-' takes effect only outside the POSIX locale.
Also the TIME_STYLE environment variable sets the default style to use.
.PP
Using color to distinguish file types is disabled both by default and
with \fb\l-color\fr=\f1,never\fr. With \fb\l-color\fr=\f1,auto\fr, ls emits color codes only when
standard output is connected to a terminal. The LS_COLORS environment
variable can change the settings. Use the dircolors command to set it.
.SH "Exit status:"
.TP
0
If OK,
.TP
1
If minor problems (e.g., cannot access subdirectory),
.TP
2
If serious trouble (e.g., cannot access command-line argument).
.SH AUTHOR
Written by Richard M. Stallman and David Mackenzie.
.SH "REPORTING BUGS"
GNU coreutils online help: <https://www.gnu.org/software/coreutils/>
.br
Report ls translation bugs to <https://translationproject.org/team/>
.SH COPYRIGHT
Copyright ©(c) 2018 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <https://gnu.org/licenses/gpl.html>.
.br
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.
.SH "SEE ALSO"
Full documentation at: <https://www.gnu.org/software/coreutils/ls>
.br
or, available locally via: info \((a)g/coreutils) ls invocation\((a)g

```

Рис. 5: Тест №2



6. Создаю файл с расширением “sh”, в котором пишу скрипт, который принимает значение с клавиатуры и выводит на экран строку из случайных символов, длина которой равна числу, введённому с клавиатуры (рис. -fig. 6)



```
emacs@maksos-VirtualBox
File Edit Options Buffers Tools Sh-Script Help
[Icons: Save, Undo, Cut, Copy, Paste, Find]
~/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
```

7. Тестирую скрипт №3. Работает верно (рис. -fig. 7)

```
maksos@maksos-VirtualBox: ~/laby/work/2020-2021/Операционные системы/
[m] 3640
maksos@maksos-VirtualBox:~/laby/work/2020-2021/Операционные системы/laboratory/lab13$ emacs &
maksos@maksos-VirtualBox:~/laby/work/2020-2021/Операционные системы/laboratory/lab13$ chmod +x lab13-3.sh
maksos@maksos-VirtualBox:~/laby/work/2020-2021/Операционные системы/laboratory/lab13$ ./lab13-3.sh 7
xplnpjpl
maksos@maksos-VirtualBox:~/laby/work/2020-2021/Операционные системы/laboratory/lab13$ ./lab13-3.sh 17
qhurtcsrbVuejVlfz
maksos@maksos-VirtualBox:~/laby/work/2020-2021/Операционные системы/laboratory/lab13$
```

Рис. 7: Тест №3

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

## Контрольные вопросы

1. Найдите синтаксическую ошибку в следующей строке `while [$1 != "exit"] )$1` следует внести в кавычки(«») `)$1`
2. Как объединить (конкатенация) несколько строк в одну? С помощью знака `>|`
3. Найдите информацию об утилите `seq`. Какими иными способами можно реализовать ее функционал при программировании на `bash`? Эта утилита выводит последовательность целых чисел с заданным шагом. Также можно реализовать с помощью утилиты `jot`.
4. Какой результат даст вычисление выражения `$((10/3))`? Результат: 3.
5. Укажите кратко основные отличия командной оболочки `zsh` от `bash`. В `zsh` можно настроить отдельные сочетания клавиш так, как вам нравится. Использование истории команд в `zsh` ничем особенным не отличается от `bash`. `Zsh` очень удобен для