

# Лабораторной работы №12

Дисциплина “Операционные системы”

---

Обрезкова А.В.

29 апреля 2023

Российский университет дружбы народов, Москва, Россия

ФФМиЕН

## Информация

---

- Обрезкова Анастасия Владимировна
- студентка направления “Математика и механика”
- Российский университет дружбы народов
- 1132226505@pfur.ru

## Вводная часть

---

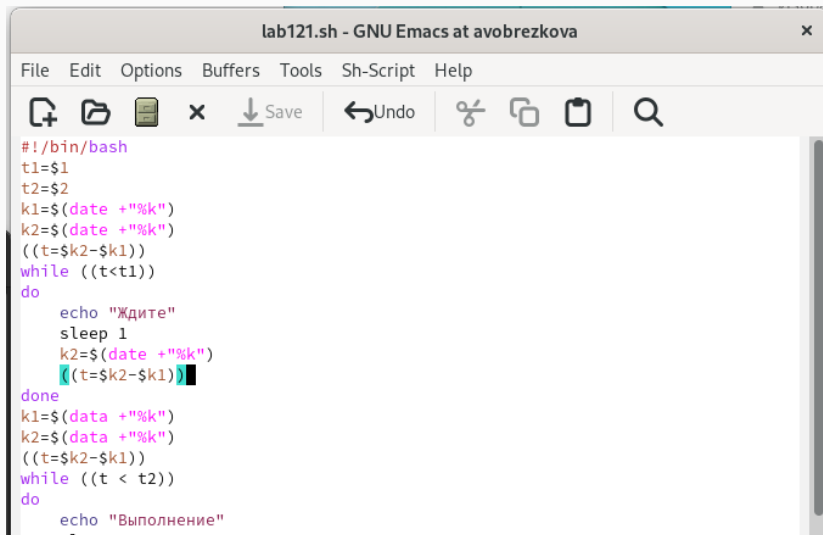
Изучить основы программирования в оболочке ОС UNIX. Научиться писать более сложные командные файлы с использованием логических управляющих конструкций и циклов.

## Основная часть

---

# Программа 1

- Для данной задачи я создала файл: lab121.sh и написала соответствующий скрипт



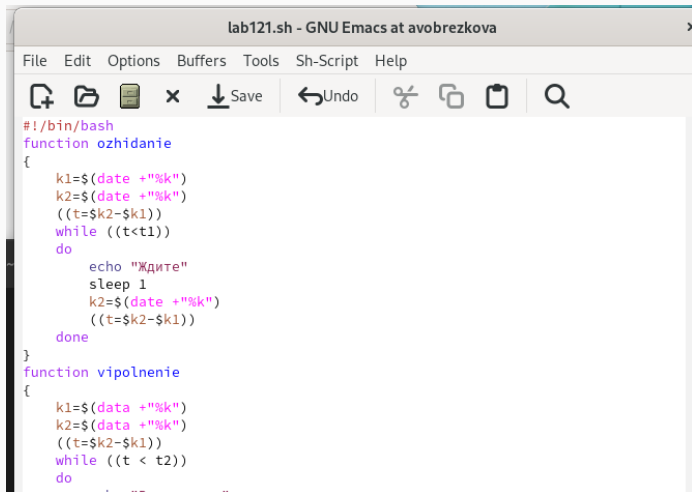
```
#!/bin/bash
t1=$1
t2=$2
k1=$(date +%k")
k2=$(date +%k")
((t=$k2-$k1))
while ((t<t1))
do
    echo "Ждите"
    sleep 1
    k2=$(date +%k")
    ((t=$k2-$k1))
done
k1=$(date +%k")
k2=$(date +%k")
((t=$k2-$k1))
while ((t < t2))
do
    echo "Выполнение"
    sleep 1
    k2=$(date +%k")
    ((t=$k2-$k1))
done
```

```
[avobrezkova@avobrezkova:~/work/study/2022-2023/Операционные системы/os-intro2/lab12/program] chmod +x lab121.sh
[avobrezkova@avobrezkova:~/work/study/2022-2023/Операционные системы/os-intro2/lab12/program] ./lab121.sh 3 5
Ждите
Ждите
Ждите
Ждите
```



# Программа 1

- После этого я изменила скрипт так, чтобы его можно было выполнять в нескольких терминалах и проверила его работу.



```
lab121.sh - GNU Emacs at avobrezkova
File Edit Options Buffers Tools Sh-Script Help
Save Undo
#!/bin/bash
function ozhidanie
{
    k1=$(date +%k")
    k2=$(date +%k")
    ((t=k2-k1))
    while ((t<t1))
    do
        echo "Ждите"
        sleep 1
        k2=$(date +%k")
        ((t=k2-k1))
    done
}
function vipolnenie
{
    k1=$(date +%k")
    k2=$(date +%k")
    ((t=k2-k1))
    while ((t < t2))
    do
```

# Программа 1

```
lab121.sh - GNU Emacs at avobrezkova
File Edit Options Buffers Tools Sh-Script Help
[Icons: New, Open, Save, Close, Undo, Cut, Copy, Paste, Find]
#!/bin/bash
function ozhidanie
{
    k1=$(date +%k)
    k2=$(date +%k)
    ((t=$k2-$k1))
    while ((t<t1))
    do
        echo "Ждите"
        sleep 1
        k2=$(date +%k)
        ((t=$k2-$k1))
    done
}
function vipolnenie
{
    k1=$(date +%k)
    k2=$(date +%k)
    ((t=$k2-$k1))
    while ((t < t2))
    do
        echo "Выполнение"
        sleep 1
        k2=$(date +%k)
        ((t=$k2-$k1))
    done
}
```

- Но ни одна команда не работала, так как мне было “Отказано в доступе”. При этом скрипт работает корректно.

```
[avobrezkova@avobrezkova:~/work/study/2022-2023/Операционные системы/os-intro2/labs/lab12/program] ./lab121.sh Ждите > /dev/pts/1 &
[1] 5395
bash: /dev/pts/1: Отказано в доступе
[1]+ Выход 1 ./lab121.sh Ждите > /dev/pts/1
[avobrezkova@avobrezkova:~/work/study/2022-2023/Операционные системы/os-intro2/labs/lab12/program] ./lab121.sh 2 4 Ждите > /dev/pts/1 &
[1] 5403
[avobrezkova@avobrezkova:~/work/study/2022-2023/Операционные системы/os-intro2/labs/lab12/program] ./lab121.sh 3 4 Ждите > /dev/pts/2 &
[1] 5413
bash: /dev/pts/2: Отказано в доступе
[1]+ Выход 1 ./lab121.sh 3 4 Ждите > /dev/pts/2
[avobrezkova@avobrezkova:~/work/study/2022-2023/Операционные системы/os-intro2/labs/lab12/program]
```

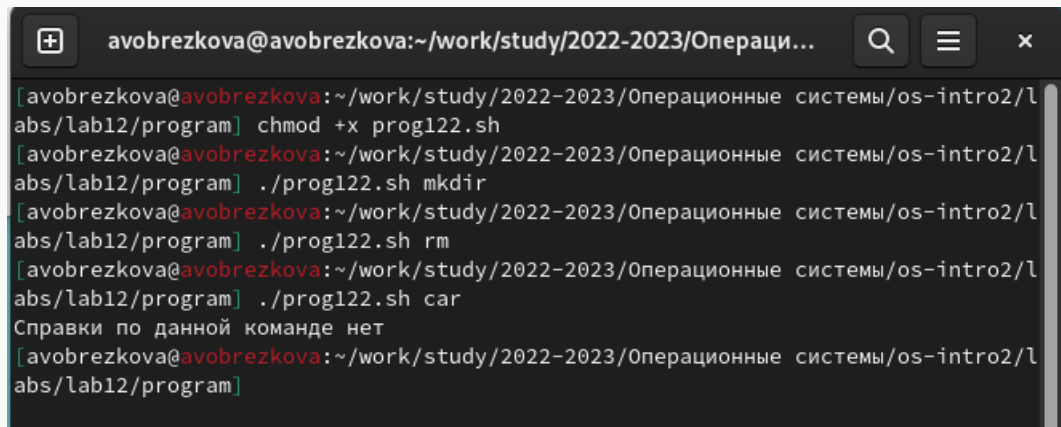
## Программа 2

```
[avobrezkova@avobrezkova:~/work/study/2022-2023/Операционные системы/os-intro2/labs/lab12/program] cd /usr/share/man/man1
[avobrezkova@avobrezkova:/usr/share/man/man1] ls
.:1.gz                                mren.1.gz
'.1.gz'                               msgattrib.1.gz
a2ping.1.gz                          msgcat.1.gz
ab.1.gz                              msgcmp.1.gz
abrt.1.gz                             msgcomm.1.gz
abrt-action-analyze-backtrace.1.gz   msgconv.1.gz
abrt-action-analyze-c.1.gz           msgen.1.gz
abrt-action-analyze-ccpp-local.1.gz  msgexec.1.gz
abrt-action-analyze-core.1.gz        msgfilter.1.gz
abrt-action-analyze-java.1.gz        msgfmt.1.gz
abrt-action-analyze-oops.1.gz        msggrep.1.gz
abrt-action-analyze-python.1.gz      msginit.1.gz
abrt-action-analyze-vmcore.1.gz      msgmerge.1.gz
abrt-action-analyze-vulnerability.1.gz msgunfmt.1.gz
abrt-action-analyze-xorg.1.gz        msguniq.1.gz
abrt-action-check-oops-for-hw-error.1.gz mshortname.1.gz
abrt-action-find-bodhi-update.1.gz   mshowfat.1.gz
abrt-action-generate-backtrace.1.gz  msxlint.1.gz
abrt-action-generate-core-backtrace.1.gz mthelp.1.gz
abrt-action-install-debuginfo.1.gz   mtools.1.gz
abrt-action-list-dsos.1.gz           mtoolstest.1.gz
abrt-action-notify.1.gz              mtrace.1.gz
abrt-action-perform-ccpp-analysis.1.gz mtz-babel.1.gz
abrt-action-save-package-data.1.gz   mtz-base.1.gz
abrt-action-trim-files.1.gz          mtz-bibtex.1.gz
abrt-applet.1.gz                    mtz-cache.1.gz
abrt-auto-reporting.1.gz             mtz-chars.1.gz
abrt-bodhi.1.gz                     mtz-check.1.gz
```

- Для данной задачи я создала файл: prod122.sh и написала соответствующий скрипт

fi" data-bbox="57 214 933 929"/>

```
prog112.sh - GNU Emacs at avobrezkova
File Edit Options Buffers Tools Sh-Script Help
[Icons: Open, Save, Undo, Redo, Copy, Paste, Search]
#!/bin/bash
a=$1
if [ -f /usr/share/man/man1/$a.1.gz ]
then
    gunzip -c /usr/share/man/man1/$1.1.gz | less
else
    echo "Справки по данной команде нет"
fi
```



```
avobrezkova@avobrezkova:~/work/study/2022-2023/Операционные системы/os-intro2/lab12/program] chmod +x prog122.sh
avobrezkova@avobrezkova:~/work/study/2022-2023/Операционные системы/os-intro2/lab12/program] ./prog122.sh mkdir
avobrezkova@avobrezkova:~/work/study/2022-2023/Операционные системы/os-intro2/lab12/program] ./prog122.sh rm
avobrezkova@avobrezkova:~/work/study/2022-2023/Операционные системы/os-intro2/lab12/program] ./prog122.sh car
Справки по данной команде нет
avobrezkova@avobrezkova:~/work/study/2022-2023/Операционные системы/os-intro2/lab12/program]
```

## Программа 2

```
avobrezkova@avobrezkova:~/work/study/2022-2023/Операци...
.\" DO NOT MODIFY THIS FILE! It was generated by help2man 1.48.5.
.TH MKDIR "1" "January 2023" "GNU coreutils 9.0" "User Commands"
.SH NAME
mkdir \- make directories
.SH SYNOPSIS
.B mkdir
[\fI\,OPTION\/\fR]... \fI\,DIRECTORY\/\fR...
.SH DESCRIPTION
.\" Add any additional description here
.PP
Create the DIRECTORY(ies), if they do not already exist.
.PP
Mandatory arguments to long options are mandatory for short options too.
.TP
\fB\-m\\fR, \fB\---mode\\fR=\fI\,MODE\/\fR
set file mode (as in chmod), not a=rwx \- umask
.TP
\fB\-p\\fR, \fB\---parents\\fR
no error if existing, make parent directories as needed,
with their file modes unaffected by any \fB\-m\\fR option.
.TP
\fB\-v\\fR, \fB\---verbose\\fR
print a message for each created directory
:
```

- Используя встроенную переменную `$RANDOM`, написала командный файл, генерирующий случайную последовательность букв латинского алфавита. Для данной задачи я создала файл: `prog123.sh` и написал соответствующий скрипт.



# Программа 3

```
prog123.sh - GNU Emacs at avobrezkova
File Edit Options Buffers Tools Sh-Script Help
[Icons: New, Open, Save, Close, Undo, Cut, Copy, Paste, Find]
#!/bin/bash
a=$1
for ((i=0; i<$a; 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
```

```
[avobrezkova@avobrezkova:~/work/study/2022-2023/Операционные системы/os-intro2/lab12/program] chmod +x prog123.sh
[avobrezkova@avobrezkova:~/work/study/2022-2023/Операционные системы/os-intro2/lab12/program] ./prog123.sh 5
brnj1
[avobrezkova@avobrezkova:~/work/study/2022-2023/Операционные системы/os-intro2/lab12/program] ./prog123.sh 9
qzfyzygxm
[avobrezkova@avobrezkova:~/work/study/2022-2023/Операционные системы/os-intro2/lab12/program] █
```

## Заключение

---

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