

# Отчет по лабораторной работе №12

---

Кашкин Иван Евгеньевич

28 мая 2022 г.

РУДН, Москва, Россия

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

Прописать 3 скрипта заданных в лабораторной работе

Сначала создаем файлы для скриптов и открываем эти файлы в emacs

```
[ivanekashkin@iekashkin lab12]$ touch prog1_semafor.sh  
[ivanekashkin@iekashkin lab12]$ emacs &
```

Рис. 1: Создание первой программы

## Первый скрипт

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



```
[ivanekashkin@iekashkin lab12]$ chmod +x *.sh
[1]+  Завершён      emacs
[ivanekashkin@iekashkin lab12]$ ./prog1_semafor.sh
[ivanekashkin@iekashkin lab12]$ ./prog1_semafor.sh 5 4
Ожидание
Ожидание
Ожидание
Ожидание
Ожидание
Выполнение
Выполнение
Выполнение
Выполнение
[ivanekashkin@iekashkin lab12]$
```

Рис. 3: Ввод программы

# Модернизированный скрипт

```
#!/bin/bash
function ozhid{
    s1=$(date +%s")
    s2=$(date +%s")
    ((t=s2-s1))
    while ((t<t1))
    do
        echo "Ожидание"
        sleep 1
        s2=$(date +%s")
        ((t=s2-s1))
    done
}
function vipoln{
    s1=$(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
com=$3
while true
do
    if [ "$com"=="Exit" ]
    then
        echo "Exit"
        exit 0
    fi
    if [ "$com"=="Ожидание"]
    then ozhid
    fi
    if [ "$com"=="Выполнение"]
    then vipoln
    fi
    echo "Следующее действие: "
    read com
done
```

```
[ivanekashkin@iekashkin lab12]$ sudo ./prog1_semafor.sh 5 4 Ожидание > /dev/pts/1 &  
[1] 13519  
bash: /dev/pts/1: Отказано в доступе  
[1]+ Выход 1          sudo ./prog1_semafor.sh 5 4 Ожидание > /dev/pts/1  
[ivanekashkin@iekashkin lab12]$ sudo ./prog1_semafor.sh 5 4 Ожидание > /dev/pts/2 &  
[1] 13554  
bash: /dev/pts/2: Отказано в доступе  
[1]+ Выход 1          sudo ./prog1_semafor.sh 5 4 Ожидание > /dev/pts/2  
[ivanekashkin@iekashkin lab12]$ ./prog1_semafor.sh 5 4 Ожидание > /dev/pts/2 &  
[1] 13571  
bash: /dev/pts/2: Отказано в доступе  
[1]+ Выход 1          ./prog1_semafor.sh 5 4 Ожидание > /dev/pts/2  
[ivanekashkin@iekashkin lab12]$
```

Рис. 5: Ввод программы



```
#!/bin/bash
a=$1
if [ -f /usr/share/man/man1/$a.1.gz ]
then
    gzip -c /usr/share/man/man1/$a.1.gz | less
else
    echo "Справки нет"
fi
□
```

Рис. 6: Программа №2

```

.\" DO NOT MODIFY THIS FILE!  It was generated by help2man 1.47.3.
.TH LS "1" "March 2022" "GNU coreutils 8.32" "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

```

Рис. 7: Ввод программы

## Третий скрипт

```
#!/bin/bash

a=$1
for ((i=0; i<$a; i++))
do
    ((char=$RANDOM%26+1))
    case $char in
        1) echo -n A;; 14) echo -n N;;
        2) echo -n B;; 15) echo -n O;;
        3) echo -n C;; 16) echo -n P;;
        4) echo -n D;; 17) echo -n Q;;
        5) echo -n E;; 18) echo -n R;;
        6) echo -n F;; 19) echo -n S;;
        7) echo -n G;; 20) echo -n T;;
        8) echo -n H;; 21) echo -n U;;
        9) echo -n I;; 22) echo -n V;;
        10) echo -n J;; 23) echo -n W;;
        11) echo -n K;; 24) echo -n X;;
        12) echo -n L;; 25) echo -n Y;;
        13) echo -n M;; 26) echo -n Z;;
    esac
done
echo
□
```

## Ввод программы

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

Спасибо за внимание!