

# Программирование в командном процессоре ОС UNIX. Расширенное программирование

Лабораторная работа №12

---

Козлова Нонна

Российский университет дружбы народов, Москва, Россия

НБИбд-04-22

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

## 1. Создаю командный файл

```
nykozlova@dk5n56 ~/work/study/2022-2023/Операционные системы/os-intro/labs/lab12
$ touch lab12.sh
nykozlova@dk5n56 ~/work/study/2022-2023/Операционные системы/os-intro/labs/lab12
$ chmod +x lab12.sh
```

Рис. 1: Пользуюсь командами touch и chmod

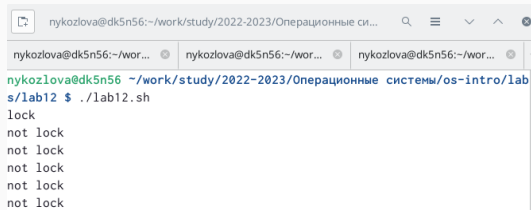
## 2. Пишу программный код, реализующий упрощённый механизм семафоров

```
File Edit Options Buffers Tools Sh-Script Help
New File Open Open Directory Close Save Undo Cut

#!/bin/bash
lockfile="./lockfile"
exec {fn}>$lockfile
echo "lock"
until flock -n ${fn}
do
    echo "not lock"
    sleep 1
    flock -n ${fn}
done
for ((i=0;i<=5; i++))
do
    sleep 1
done

--:--- lab12.sh All L1 (Shell-scr
Welcome to GNU Emacs, one component of the GNU/Linux
```

### 3. Запускаю файл

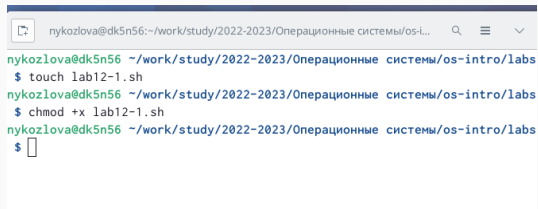


A terminal window with a title bar showing the user 'nykozlova@dk5n56' and the current directory '~/work/study/2022-2023/Операционные си...'. The terminal has three tabs, all with the same title. The active tab shows the following text:

```
nykozlova@dk5n56 ~/work/study/2022-2023/Операционные системы/os-intro/labs/lab12 $ ./lab12.sh
lock
not lock
not lock
not lock
not lock
not lock
```

Рис. 3: Работает

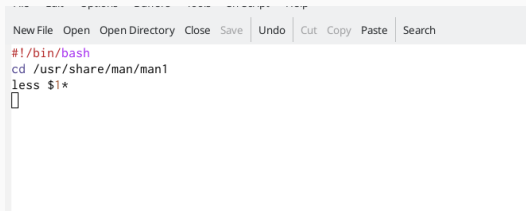
#### 4. Создаю командный файл



```
nykozlova@dk5n56:~/work/study/2022-2023/Операционные системы/os-i...
nykozlova@dk5n56 ~/work/study/2022-2023/Операционные системы/os-intro/labs
$ touch lab12-1.sh
nykozlova@dk5n56 ~/work/study/2022-2023/Операционные системы/os-intro/labs
$ chmod +x lab12-1.sh
nykozlova@dk5n56 ~/work/study/2022-2023/Операционные системы/os-intro/labs
$
```

Рис. 4: Пользуюсь командами touch и chmod

## 5. Пишу программный код, реализующий команду man



```
New File Open Open Directory Close Save Undo Cut Copy Paste Search
#!/bin/bash
cd /usr/share/man/man1
less $1*
```

Рис. 5: Код

## 6. Запускаю файл

```
nykozlova@dk5n56 ~/work/study/2022-2023/Операционные системы/os-intro/labs/lab12
$ ./lab12-1.sh less
nykozlova@dk5n56 ~/work/study/2022-2023/Операционные системы/os-intro/labs/lab12 $
```

Рис. 6: Работает



```
nykozlova@dk5n56:~/work/study/2022-202
```

NAME

less - opposite of more

SYNOPSIS

```
less ~?
less --help
less -V
less --version
less [-[+jaAbcCdeEffgGiIJKLmMnNqQrRsSuUVvWx~]
      [-b space] [-h lines] [-j line] [-k keyfile]
      [-{oO} logfile] [-p pattern] [-P prompt] [-t tag]
      [-T tagsfile] [-x tab,...] [-y lines] [-{z} lines]
      [-# shift] [+{+}cmd] [--] [filename]...
```

(See the OPTIONS section for alternate option syntax with long option names.)

DESCRIPTION

Less is a program similar to more(1), but which allows backward movement in the file as well as forward movement. Also, less does not have to read the entire input file before starting, so with large input files it starts up faster than text editors like vi(1). Less uses termcap (or terminfo on some systems), so it can run on a variety of terminals. There is even limited support for hardcopy terminals. (On a hardcopy terminal, lines which should be printed at the top of the screen are prefixed with a caret.)

Commands are based on both more and vi. Commands may be preceded by a decimal number, called N in the descriptions below. The number is used by some commands, as indicated.

COMMANDS

In the following descriptions, ^X means control-X. ESC stands for the ESCAPE key; for example ESC-v means the two character sequence "ESCAPE", then "v".

h or H Help: display a summary of these commands. If you forget all the other commands, remember this one.

SPACE or ^V or f or ^F

Scroll forward N lines, default one window (see option -z below). If N is more than the screen size, only the final screen-

less.1.bz2 lines 2-4]

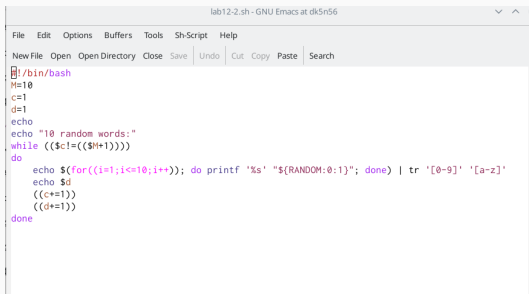
Рис. 7: Работает

## 7. Создаю командный файл .

```
nykozlova@dk5n56 ~/work/study/2022-2023/Операционные системы/os-intro/labs/lab12 $ touch lab12-2.sh
nykozlova@dk5n56 ~/work/study/2022-2023/Операционные системы/os-intro/labs/lab12 $ chmod +x lab12-2.sh
nykozlova@dk5n56 ~/work/study/2022-2023/Операционные системы/os-intro/labs/lab12 $
```

Рис. 8: Пользуюсь командами touch и chmod

9. Пишу программный код, генерирующий случайную последовательность букв латинского алфавита



```
lab12-2.sh - GNU Emacs at dk5n56
File Edit Options Buffers Tools Sh-Script Help
New File Open Open Directory Close Save Undo Cut Copy Paste Search
#!/bin/bash
M=10
c=1
d=1
echo
echo "10 random words:"
while (($c!=($M+1)))
do
    echo $(for((i=1;i<=10;i++)); do printf '%s' "${RANDOM:0:1}"; done) | tr '[:0-9]' '[:a-z]'
    echo $d
    ((c+=1))
    ((d+=1))
done
```

Рис. 9: Код

## 10. Запускаю файл

```
nykozlova@dk5n56 ~/work/study/2022-2023/Операционные системы/os-intro/labs/lab12 $ ./lab12-2.sh
10 random words:
becbbdcdfc
1 dcbccjbccf
2 cebdcbbcbc
3 hbceiccccc
4 bigjhhcbcd
5 bccccjcdbe
6 jcbccccbdc
7 bccdbdbjbj
8 cdfedcbfci
9 cccbcccbch
10
nykozlova@dk5n56 ~/work/study/2022-2023/Операционные системы/os-intro/labs/lab12 $
```

Рис. 10: Работает

В ходе лабораторной работы я изучила основы программирования в оболочке ОС UNIX. Научилась писать более сложные командные файлы с использованием логических управляющих конструкций и циклов.