

Отчет по лабораторной работе №12

Дисциплина: Компьютерные науки и технологии программирования

Ангелина Павловна Ким

Содержание

1	Цель работы	5
2	Выполнение лабораторной работы	6
3	Выводы	10
	Список литературы	11

Список иллюстраций

2.1	Первая программа	6
2.2	Выполнение первой программы	7
2.3	Вторая программа	8
2.4	Выполнение второй программы	8
2.5	Третья программа	9
2.6	Выполнение третьей программы	9

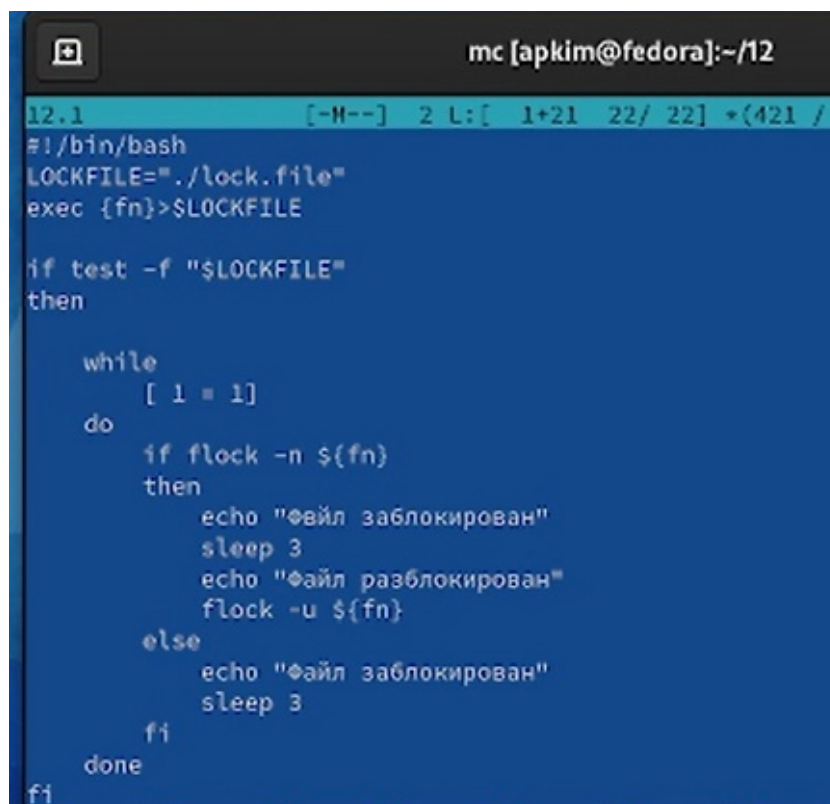
Список таблиц

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

Первая программа (рис. 2.1).

A screenshot of a terminal window with a dark background. The title bar at the top reads 'mc [apkim@fedora]:~/12'. The terminal content shows a shell script starting with '12.1' in a light blue prompt, followed by '#!/bin/bash'. The script sets 'LOCKFILE=./lock.file' and uses 'exec {fn}>\$LOCKFILE'. It then checks if the lock file exists with 'if test -f "\$LOCKFILE"'. If it does, it enters a 'while' loop with '[1 = 1]' as the condition. Inside the loop, it uses 'do' and 'fi' blocks. The 'do' block contains an 'if flock -n \${fn}' check. If successful, it echoes 'Файл заблокирован', sleeps for 3 seconds, echoes 'Файл разблокирован', and unlocks the file with 'flock -u \${fn}'. If the flock command fails, it echoes 'Файл заблокирован' and sleeps for 3 seconds. The 'done' keyword ends the while loop, and the final 'fi' ends the main if statement.

```
mc [apkim@fedora]:~/12
12.1 [ -M-- ] 2 L: [ 1+21 22/ 22 ] * (421 /
#!/bin/bash
LOCKFILE=./lock.file
exec {fn}>$LOCKFILE

if test -f "$LOCKFILE"
then
    while
        [ 1 = 1 ]
    do
        if flock -n ${fn}
        then
            echo "Файл заблокирован"
            sleep 3
            echo "Файл разблокирован"
            flock -u ${fn}
        else
            echo "Файл заблокирован"
            sleep 3
        fi
    done
fi
```

Рис. 2.1: Первая программа

Выполнение первой программы (рис. 2.2).

```

итого 4
-rwxrwxr-x. 1 apkim apkim 421 апр 25 21:27 12.1
[apkim@fedora 12]$ mc

[apkim@fedora 12]$ ./12.1
./12.1: строка 9: [: отсутствует символ «]»

[apkim@fedora 12]$ ./12.1
./12.1: строка 9: [: отсутствует символ «]»

[apkim@fedora 12]$ ./12.1
./12.1: строка 9: [: отсутствует символ «]»

[apkim@fedora 12]$ ./12.1
Файл заблокирован
Файл разблокирован
Файл заблокирован
Файл разблокирован
Файл заблокирован
Файл разблокирован
Файл заблокирован
Файл разблокирован
Файл заблокирован

```

Рис. 2.2: Выполнение первой программы

Вторая программа (рис. 2.3).

```

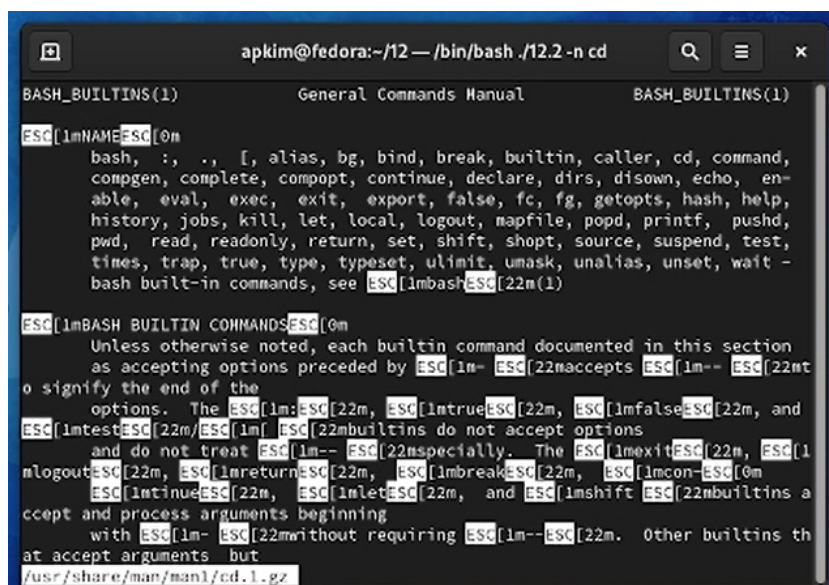
12.2 [-M--] 17 L:[ 1+ 3 4/ 15] *(4
#!/bin/bash
command=""

while getopts :n: opt
do
    case $opt in
        n) command="$OPTARG";;
        esac
    done
    ....
if test -f "/usr/share/man/man1/$command.1.gz"
then less /usr/share/man/man1/$command.1.gz
else
echo "no such command"
fi

```

Рис. 2.3: Вторая программа

Выполнение второй программы (рис. 2.4).



```

apkim@fedora:~/12 — /bin/bash ./12.2 -n cd
BASH_BUILTINS(1)      General Commands Manual      BASH_BUILTINS(1)

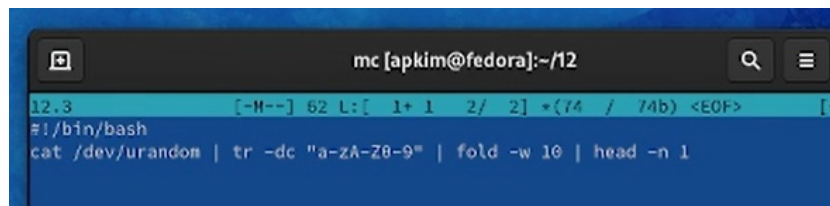
ESC[1mNAMEESC[0m
    bash, :, ., [, alias, bg, bind, break, builtin, caller, cd, command,
    compgen, complete, compopt, continue, declare, dirs, disown, echo, en-
    able, eval, exec, exit, export, false, fc, fg, getopts, hash, help,
    history, jobs, kill, let, local, logout, mapfile, popd, printf, pushd,
    pwd, read, readonly, return, set, shift, shopt, source, suspend, test,
    times, trap, true, type, typeset, ulimit, umask, unalias, unset, wait -
    bash built-in commands, see ESC[1mbashESC[22m(1)

ESC[1mBASH BUILTIN COMMANDSESC[0m
    Unless otherwise noted, each builtin command documented in this section
    as accepting options preceded by ESC[1m- ESC[22maccepts ESC[1m-- ESC[22mt
    o signify the end of the
    options. The ESC[1m-ESC[22m, ESC[1mtrueESC[22m, ESC[1mfalseESC[22m, and
    ESC[1mtestESC[22m/ESC[1mESC[22mbuiltins do not accept options
    and do not treat ESC[1m- ESC[22mspecially. The ESC[1mexitESC[22m, ESC[1l
    nlogoutESC[22m, ESC[1mreturnESC[22m, ESC[1mbreakESC[22m, ESC[1mcon-ESC[0m
    ESC[1mcontinueESC[22m, ESC[1mletESC[22m, and ESC[1mshift ESC[22mbuiltins a
    ccept and process arguments beginning
    with ESC[1m- ESC[22mwithout requiring ESC[1m--ESC[22m. Other builtins th
    at accept arguments but
    /usr/share/man/man1/cd.1.gz

```

Рис. 2.4: Выполнение второй программы

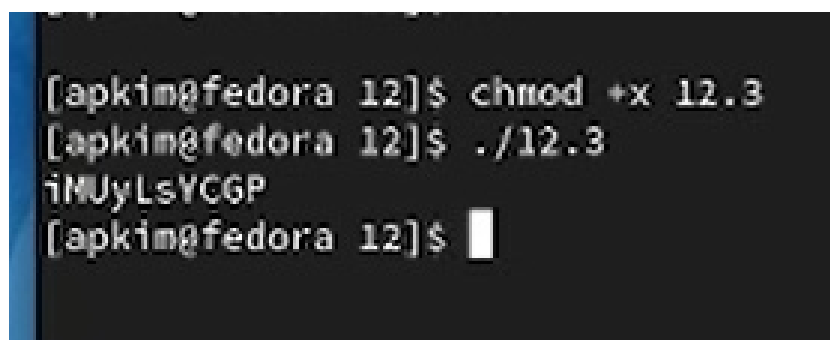
Третья программа (рис. 2.5).



```
mc [apkim@fedora]:~/12
12.3      [-M--] 62 L: [ 1+ 1  2/  2] *{74 / 74b} <EOF>
#!/bin/bash
cat /dev/urandom | tr -dc "a-zA-Z0-9" | fold -w 10 | head -n 1
```

Рис. 2.5: Третья программа

Выполнение третьей программы (рис. 2.6).



```
[apkim@fedora 12]$ chmod +x 12.3
[apkim@fedora 12]$ ./12.3
iMUyLsYCGP
[apkim@fedora 12]$
```

Рис. 2.6: Выполнение третьей программы

3 Выводы

В ходе выполнения данной лабораторной работы мы научились писать более сложные командные файлы.

Список литературы