

Лабораторная работа №12

Операционные системы

Кирилюк Светлана Алексеевна

Содержание

1	Цель работы	5
2	Выполнение лабораторной работы	6
3	Выводы	9

Список иллюстраций

2.1	Скрипт 1-ой программы	6
2.2	Выполнение 1-ой программы	7
2.3	Скрипт 2-ой программы	7
2.4	Выполнение 2-ой программы	8
2.5	Скрипт 3-й программы	8
2.6	Выполнение 3-й программы	8

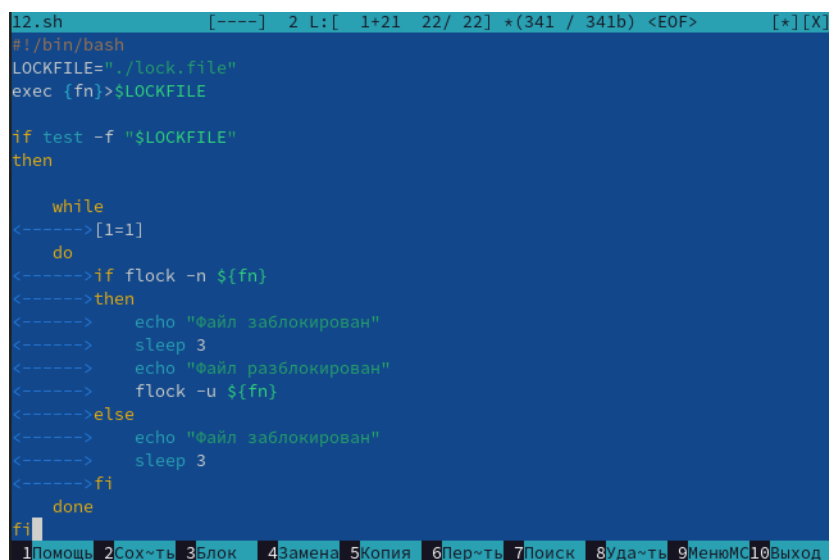
Список таблиц

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

Я создала новый файл с командой `touch` и перешла в него при помощи `tc`, после чего написала скрипт программы (рис. 2.1). Затем я написала команду, которая даёт все права пользователю на работу с файлом, и выполнила программу (рис. 2.2).



```
12.sh [----] 2 L:[ 1+21 22/ 22] *(341 / 341b) <EOF> [*][X]
#!/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
```

1Помощь 2Сох~ть 3Блок 4Замена 5Копия 6Пер~ть 7Поиск 8Уда~ть 9МенюMC10Выход

Рис. 2.1: Скрипт 1-ой программы

```
[sakirilyuk@fedora ~]$ chmod +x *.sh
[sakirilyuk@fedora ~]$ ./12.sh
./12.sh: строка 9: [1=1]: команда не найдена
[sakirilyuk@fedora ~]$ mc

[sakirilyuk@fedora ~]$ ./12.sh
Файл заблокирован
Файл разблокирован
Файл заблокирован
Файл разблокирован
Файл заблокирован
Файл разблокирован
Файл заблокирован
^C
[sakirilyuk@fedora ~]$
```

Рис. 2.2: Выполнение 1-ой программы

Вновь создав файл и открыв его в `mc`, я написала новый скрипт (рис. 2.3). И снова выполнила программу (рис. 2.4).

```
12.2.sh [-M--] 2 L:[ 1+14 15/ 15] *(237 / 237b) <EOF> [*][X]
#!/bin/bash
command=""

while getopts :in: opt
do
    case $opt in
        n) command="$OPTARG";;
        esac
    done
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: Скрипт 2-ой программы

```

BASH_BUILTINS(1)          General Commands Manual          BASH_BUILTINS(1)

ESC[1mNAMEESC[0m
    :, ., [, alias, bg, bind, break, builtin, caller, cd, command, compgen,
    complete, compopt, continue, declare, dirs, disown, echo, enable, eval,
    exec, exit, export, false, fc, fg, getopts, hash, help, history, jobs,
    kill, let, local, logout, mapfile, popd, printf, pushd, pwd, read,
    readarray, 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[1m[ ESC[22mbuiltins do not accept options
    and do not treat ESC[1m-- ESC[22mspecially. The ESC[1mexitESC[22m, ESC[1
    mlogoutESC[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 2-ой программы

Создав последний файл и зпустив его, я вновь написала скрипт (рис. 2.5) и выполнила программу (рис. 2.6).

```

12.3.sh [-M--] 33 L:[ 1+ 1 2/ 3] *(45 / 75b) 0048 0x030 [*][X]
#!/bin/bash
cat /dev/urandom | tr -dc "a-zA-Z0-9" | fold -w 10 | head -n 1

```

Рис. 2.5: Скрипт 3-й программы

```

[sakirilyuk@fedora ~]$ chmod +x 12.3.sh
[sakirilyuk@fedora ~]$ ./12.3.sh
u6tjr7MDim
[sakirilyuk@fedora ~]$

```

Рис. 2.6: Выполнение 3-й программы

3 Выводы

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.