

Лабораторная работа №12

**Программирование в командном процессоре ОС UNIX. Расширенное
программирование**

Медникова Екатерина Михайловна

Содержание

1	Цель работы	5
2	Выполнение лабораторной работы	6
3	Выводы	9
4	Контрольные вопросы	10

Список иллюстраций

2.1	Задание 1	6
2.2	Результат задания 1	6
2.3	Задание 2	7
2.4	Результат задания 2	7
2.5	Задание 3	7
2.6	Результат задания 3	8

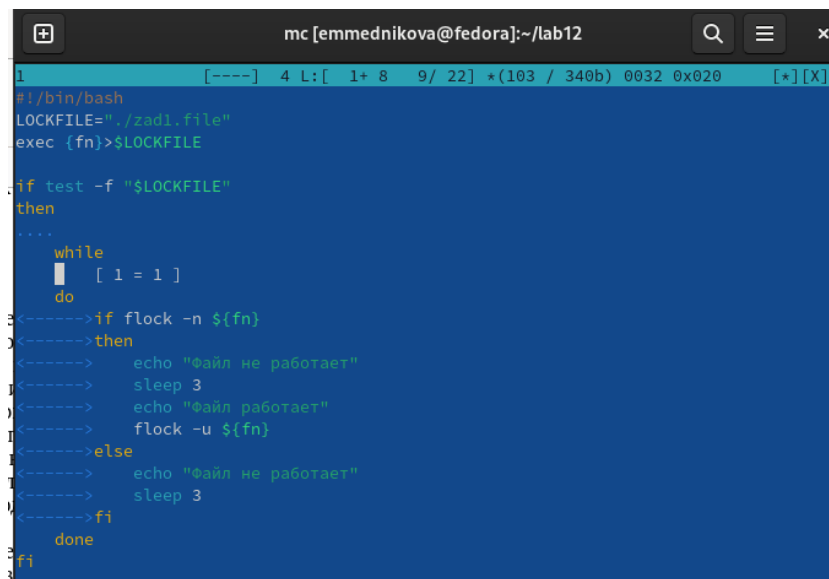
Список таблиц

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

1. Написала командный файл, реализующий упрощённый механизм семафоров.



```
1 [-----] 4 L: [ 1+ 8 9/ 22] *(103 / 340b) 0032 0x020 [*] [X]
#!/bin/bash
LOCKFILE="/zad1.file"
exec {fn}>$LOCKFILE

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

Рис. 2.1: Задание 1



```
[emmednikova@fedora lab12]$ ./1
Файл не работает
Файл работает
Файл не работает
^C
[emmednikova@fedora lab12]$
```

Рис. 2.2: Результат задания 1

2. Реализовала команду man с помощью командного файла. Изучила содержимое каталога /usr/share/man/man1.

```
mc [emmednikova@fedora]:~/lab12
2 [-----] 8 L:[ 1+ 2 3/ 3] *(43 / 43b) <EOF> [*][X]
#!/bin/bash
cd /usr/share/man/man1
less $1*
```

Рис. 2.3: Задание 2

```
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 , getopt , 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[22mto s
ignify 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[1ml
ogoutESC[22m, ESC[1mreturnESC[22m, ESC[1mbreakESC[22m, ESC[1mcon-ESC[0m
ESC[1mtrueESC[22m, ESC[1mletESC[22m, and ESC[1mshift ESC[22mbuiltins ac
cept and process arguments beginning
with ESC[1m- ESC[22mwithout requiring ESC[1m--ESC[22m. Other builtins that
accept arguments but
are not specified as accepting options interpret arguments beginning with
ESC[1m- ESC[22mas invalid options and require ESC[1m-- ESC[22mto prevent thi
s interpretation.
ESC[1m: ESC[22mESC[4margumentsESC[24m
No effect; the command does nothing beyond expanding ESC[4marguments
ESC[24m and
performing any specified redirections. The return status is zero.
```

Рис. 2.4: Результат задания 2

3. Написала командный файл, генерирующий случайную последовательность букв латинского алфавита.

```
mc [emmednikova@fedora]:~/lab12
3 [-----] 62 L:[ 1+ 1 2/ 2] *(74 / 74b) <EOF> [*][X]
#!/bin/bash
cat /dev/urandom | tr -dc "a-zA-Z0-9" | fold -w 10 | head -n 1
```

Рис. 2.5: Задание 3

```
[emmednikova@fedora lab12]$ chmod +x 3  
[emmednikova@fedora lab12]$ ./3  
GSKCJRNeJ8  
[emmednikova@fedora lab12]$
```

Рис. 2.6: Результат задания 3

3 Выводы

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

4 Контрольные вопросы

1. Найдите синтаксическую ошибку в следующей строке: `while [$1 != "exit"]`

Ответ: Ошибка заключается в `$1`. Также нет пробелов между квадратными скобками.

2. Как объединить (конкатенация) несколько строк в одну?

Ответ: Добавление переменных или литеральных строк к переменной с помощью оператора `+=`

3. Найдите информацию об утилите `seq`. Какими иными способами можно реализовать её функционал при программировании на `bash`?

Ответ: Утилит `seq` выдаёт последовательность чисел. Реализовать можно с помощью команды `for n in {1..5} do done`.

4. Какой результат даст вычисление выражения `$((10/3))`?

Ответ: 3

5. Укажите кратко основные отличия командной оболочки `zsh` от `bash`.

Ответ: Оболочка `zsh` предлагает встроенную коррекцию орфографии, улучшенное завершение командной строки, загружаемые модули, которые выступают в качестве плагинов для оболочки, глобальные псевдонимы, которые позволяют использовать псевдонимы имен файлов или чего-либо еще в командной строке вместо просто команд, и больше поддержки тем. Это похоже на `bash`, но с множеством дополнительных возможностей, дополнительных функций и настраиваемых параметров.

6. Проверьте, верен ли синтаксис данной конструкции: `for ((a=1; a <= LIMIT; a++))`

Ответ: Да, верен.

7. Сравните язык *bash* с какими-либо языками программирования. Какие преимущества у *bash* по сравнению с ними? Какие недостатки?

Bash — это командный язык, а не язык программирования общего назначения. Поэтому с усложнением кода он становится более запутанным и менее читаемым. Кроме того, Bash всё и всегда воспринимает как команду, потому что это командный язык. У Bash нет стандартного API, однако он поставляется с простыми встроенными функциями (например, со встроенной тестовой обработкой). Однако часто придется создавать процессы для обработки данных.