

Лабораторная работа №13

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

Ким Михаил Алексеевич

Содержание

1	Цель работы	3
2	Выполнение лабораторной работы.	4
3	Выводы	15
4	Термины	16

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. Написали командный файл, реализующий упрощённый механизм семафоров. Командный файл в течение некоторого времени дожидаться освобождения ресурса, выдавая об этом сообщение, а дождавшись его освобождения, использует его в течение некоторого времени, также выдавая информацию о том, что ресурс используется. К сожалению, у меня не получилось никакими методами перенести вывод из одного виртуального терминала на другой. (рис. 2.1 - 2.8)

```
#!/bin/bash
```

```
// Пока существует файл lockfile
```

```
while test -f lockfile
```

```
do
```

```
    echo Ждем освобождения ресурса \(\удаления файла lockfile\)
```

```
    echo 'Ждем освобождения ресурса \(\удаления файла lockfile\)' >> res.txt
```

```
    sleep 5
```

```
done
```

```
// Создаем файл lockfile
```

```
touch lockfile
```

```
echo -n 'Данный терминал занимает ресурс (файл lockfile создан): '
```

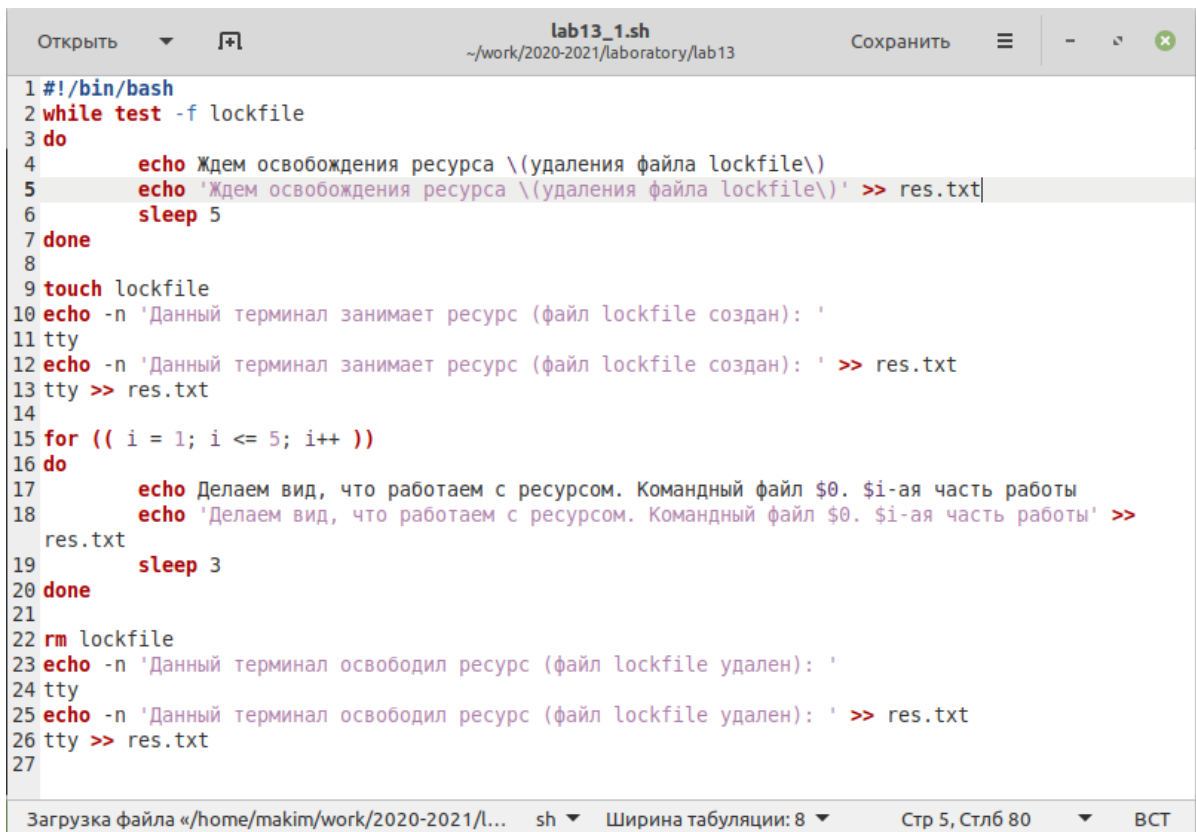
```
tty
```

```
echo -n 'Данный терминал занимает ресурс (файл lockfile создан): ' >> res.tx
```

```
tty >> res.txt
```

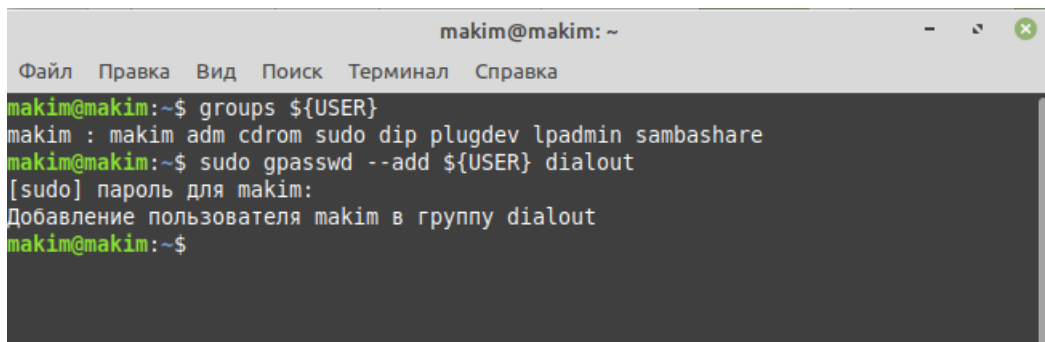
```
// Сообщения просто для того, чтобы показать возможность работы в это время
for (( i = 1; i <= 5; i++ ))
do
    echo Делаем вид, что работаем с ресурсом. Командный файл $0. $i-ая часть
    echo 'Делаем вид, что работаем с ресурсом. Командный файл $0. $i-ая часть'
    sleep 3
done
```

```
// Удаляем файл lockfile
rm lockfile
echo -n 'Данный терминал освободил ресурс (файл lockfile удален): '
tty
echo -n 'Данный терминал освободил ресурс (файл lockfile удален): ' >> res.txt
tty >> res.txt
```



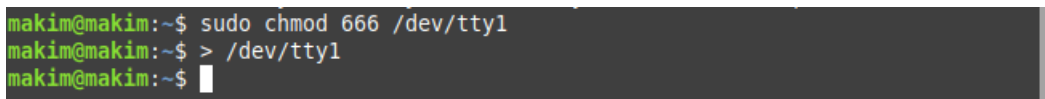
```
1#!/bin/bash
2while test -f lockfile
3do
4    echo Ждем освобождения ресурса \(\удаления файла lockfile\)
5    echo 'Ждем освобождения ресурса \(\удаления файла lockfile\) ' >> res.txt
6    sleep 5
7done
8
9touch lockfile
10echo -n 'Данный терминал занимает ресурс (файл lockfile создан): '
11tty
12echo -n 'Данный терминал занимает ресурс (файл lockfile создан): ' >> res.txt
13tty >> res.txt
14
15for (( i = 1; i <= 5; i++ ))
16do
17    echo Делаем вид, что работаем с ресурсом. Командный файл $0. $i-ая часть работы
18    echo 'Делаем вид, что работаем с ресурсом. Командный файл $0. $i-ая часть работы' >>
19    res.txt
20    sleep 3
21done
22rm lockfile
23echo -n 'Данный терминал освободил ресурс (файл lockfile удален): '
24tty
25echo -n 'Данный терминал освободил ресурс (файл lockfile удален): ' >> res.txt
26tty >> res.txt
27
```

Рис. 2.1: Исходный код



```
makim@makim:~$ groups ${USER}
makim : makim adm cdrom sudo dip plugdev lpadmin sambashare
makim@makim:~$ sudo gpasswd --add ${USER} dialout
[sudo] пароль для makim:
Добавление пользователя makim в группу dialout
makim@makim:~$
```

Рис. 2.2: Проблема с виртуальными терминалами



```
makim@makim:~$ sudo chmod 666 /dev/tty1
makim@makim:~$ > /dev/tty1
makim@makim:~$
```

Рис. 2.3: Проблема с виртуальными терминалами

```
Linux Mint 20.2 Uma makim tty2

makim login: tty
Password:

Login incorrect
makim login: makim
Password:
Last login: Sat Oct 23 02:06:41 MSK 2021 on tty1
makim@makim:~$ tty
/dev/tty2
makim@makim:~$ > /dev/tty1
-bash: /dev/tty1: Отказано в доступе
makim@makim:~$
```

Рис. 2.4: Проблема с виртуальными терминалами

```
makim@makim:~/work/2020-2021/laboratory/lab13$ tty
/dev/tty1
makim@makim:~/work/2020-2021/laboratory/lab13$ > /dev/tty2
makim@makim:~/work/2020-2021/laboratory/lab13$ bash lab13_1.sh &
[1] 1991
makim@makim:~/work/2020-2021/laboratory/lab13$ Данный терминал занимает ресурс (файл lockfile создан): /dev/tty1
Делаем вид, что работаем с ресурсом. Командный файл lab13_1.sh. 1-ая часть работы
(arg: 3) Делаем вид, что работаем с ресурсом. Командный файл lab13_1.sh. 2-ая часть работы
Делаем вид, что работаем с ресурсом. Командный файл lab13_1.sh. 3-ая часть работы
Делаем вид, что работаем с ресурсом. Командный файл lab13_1.sh. 4-ая часть работы
Делаем вид, что работаем с ресурсом. Командный файл lab13_1.sh. 5-ая часть работы
Данный терминал освободил ресурс (файл lockfile удален): /dev/tty1
makim@makim:~/work/2020-2021/laboratory/lab13$ bash lab13_1.sh
Ждем освобождения ресурса (удаления файла lockfile)
Ждем освобождения ресурса (удаления файла lockfile)
Данный терминал занимает ресурс (файл lockfile создан): /dev/tty1
Делаем вид, что работаем с ресурсом. Командный файл lab13_1.sh. 1-ая часть работы
Делаем вид, что работаем с ресурсом. Командный файл lab13_1.sh. 2-ая часть работы
Делаем вид, что работаем с ресурсом. Командный файл lab13_1.sh. 3-ая часть работы
Делаем вид, что работаем с ресурсом. Командный файл lab13_1.sh. 4-ая часть работы
Делаем вид, что работаем с ресурсом. Командный файл lab13_1.sh. 5-ая часть работы
rm: невозможно удалить 'lockfile': Нет такого файла или каталога
Данный терминал освободил ресурс (файл lockfile удален): /dev/tty1
[1]+  Завершён          bash lab13_1.sh
makim@makim:~/work/2020-2021/laboratory/lab13$
```

Рис. 2.5: Результат

```

makim@makim:~/work/2020-2021/laboratory/lab13$ tty
/dev/tty2
makim@makim:~/work/2020-2021/laboratory/lab13$ sudo bash lab13_1.sh
Ждем освобождения ресурса (удаления файла lockfile)
Ждем освобождения ресурса (удаления файла lockfile)
Ждем освобождения ресурса (удаления файла lockfile)
Ждем освобождения ресурса (удаления файла lockfile)
Ждем освобождения ресурса (удаления файла lockfile)
Данный терминал занимает ресурс (файл lockfile создан): /dev/tty2
Делаем вид, что работаем с ресурсом. Командный файл lab13_1.sh. 1-ая часть работы
Делаем вид, что работаем с ресурсом. Командный файл lab13_1.sh. 2-ая часть работы
Делаем вид, что работаем с ресурсом. Командный файл lab13_1.sh. 3-ая часть работы
Делаем вид, что работаем с ресурсом. Командный файл lab13_1.sh. 4-ая часть работы
Делаем вид, что работаем с ресурсом. Командный файл lab13_1.sh. 5-ая часть работы
Данный терминал освободил ресурс (файл lockfile удален): /dev/tty2
makim@makim:~/work/2020-2021/laboratory/lab13$ _

```

Рис. 2.6: Результат

```

makim@makim:~/work/2020-2021/laboratory/lab13$ tty
/dev/tty3
makim@makim:~/work/2020-2021/laboratory/lab13$ bash lab13_1.sh
Данный терминал занимает ресурс (файл lockfile создан): /dev/tty3
Делаем вид, что работаем с ресурсом. Командный файл lab13_1.sh. 1-ая часть работы
Делаем вид, что работаем с ресурсом. Командный файл lab13_1.sh. 2-ая часть работы
Делаем вид, что работаем с ресурсом. Командный файл lab13_1.sh. 3-ая часть работы
Делаем вид, что работаем с ресурсом. Командный файл lab13_1.sh. 4-ая часть работы
Делаем вид, что работаем с ресурсом. Командный файл lab13_1.sh. 5-ая часть работы
Данный терминал освободил ресурс (файл lockfile удален): /dev/tty3
makim@makim:~/work/2020-2021/laboratory/lab13$

```

Рис. 2.7: Результат


```

Данный терминал занимает ресурс (файл lockfile создан): /dev/tty1
Делаем вид, что работаем с ресурсом. Командный файл $0. $i-ая часть работы
Делаем вид, что работаем с ресурсом. Командный файл $0. $i-ая часть работы
Делаем вид, что работаем с ресурсом. Командный файл $0. $i-ая часть работы
Делаем вид, что работаем с ресурсом. Командный файл $0. $i-ая часть работы
Данный терминал освободил ресурс (файл lockfile удален): /dev/tty1
Данный терминал занимает ресурс (файл lockfile создан): /dev/tty2
Делаем вид, что работаем с ресурсом. Командный файл $0. $i-ая часть работы
Делаем вид, что работаем с ресурсом. Командный файл $0. $i-ая часть работы
Делаем вид, что работаем с ресурсом. Командный файл $0. $i-ая часть работы
Делаем вид, что работаем с ресурсом. Командный файл $0. $i-ая часть работы
Делаем вид, что работаем с ресурсом. Командный файл $0. $i-ая часть работы
Данный терминал освободил ресурс (файл lockfile удален): /dev/tty2
Данный терминал занимает ресурс (файл lockfile создан): /dev/tty1
Делаем вид, что работаем с ресурсом. Командный файл $0. $i-ая часть работы
Делаем вид, что работаем с ресурсом. Командный файл $0. $i-ая часть работы
Делаем вид, что работаем с ресурсом. Командный файл $0. $i-ая часть работы
Ждем освобождения ресурса \удаления файла lockfile\
Делаем вид, что работаем с ресурсом. Командный файл $0. $i-ая часть работы
Делаем вид, что работаем с ресурсом. Командный файл $0. $i-ая часть работы
Ждем освобождения ресурса \удаления файла lockfile\
Данный терминал освободил ресурс (файл lockfile удален): /dev/tty1
Данный терминал занимает ресурс (файл lockfile создан): /dev/tty3
Делаем вид, что работаем с ресурсом. Командный файл $0. $i-ая часть работы
Ждем освобождения ресурса \удаления файла lockfile\
Делаем вид, что работаем с ресурсом. Командный файл $0. $i-ая часть работы
Ждем освобождения ресурса \удаления файла lockfile\
Делаем вид, что работаем с ресурсом. Командный файл $0. $i-ая часть работы
Ждем освобождения ресурса \удаления файла lockfile\
Делаем вид, что работаем с ресурсом. Командный файл $0. $i-ая часть работы
Данный терминал освободил ресурс (файл lockfile удален): /dev/tty3
Данный терминал занимает ресурс (файл lockfile создан): Данный терминал занимает ресурс (файл lockfile создан): /dev/tty1
/dev/tty2
Делаем вид, что работаем с ресурсом. Командный файл $0. $i-ая часть работы
Делаем вид, что работаем с ресурсом. Командный файл $0. $i-ая часть работы
Делаем вид, что работаем с ресурсом. Командный файл $0. $i-ая часть работы
Делаем вид, что работаем с ресурсом. Командный файл $0. $i-ая часть работы
Делаем вид, что работаем с ресурсом. Командный файл $0. $i-ая часть работы
Делаем вид, что работаем с ресурсом. Командный файл $0. $i-ая часть работы
Делаем вид, что работаем с ресурсом. Командный файл $0. $i-ая часть работы
Делаем вид, что работаем с ресурсом. Командный файл $0. $i-ая часть работы
Делаем вид, что работаем с ресурсом. Командный файл $0. $i-ая часть работы
Делаем вид, что работаем с ресурсом. Командный файл $0. $i-ая часть работы
Данный терминал освободил ресурс (файл lockfile удален): Данный терминал освободил ресурс (файл lockfile удален): /dev/tty2
/dev/tty1

```

Рис. 2.8: Результат

2. Реализуем команду `man` с помощью командного файла. Изучаем содержимое каталога `/usr/share/man/man1`. В нем находятся архивы текстовых файлов, содержащих справку по большинству установленных в системе программ и команд. Каждый архив можно открыть командой `less` сразу же просмотрев содержимое справки. Командный файл должен получает в виде аргумента командной строки название команды и в виде результата выдаёт справку об этой команде или сообщение об отсутствии справки, если соответствующего файла нет в каталоге `man1`. (рис. 2.9 - 2.12)

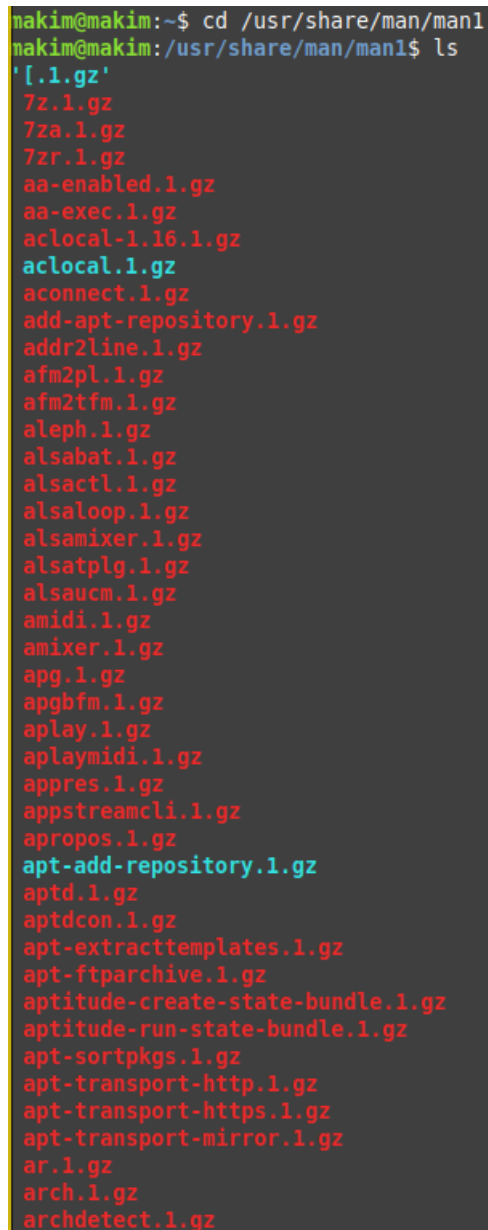
```
#!/bin/bash
```

// перемещаемся в каталог, где содержатся справки о командах

```
cd /usr/share/man/man1
```

// выводим информацию по команде

```
less $1*
```

A terminal window with a dark background and light green text. The prompt is 'nakim@makim:~\$'. The user enters 'cd /usr/share/man/man1'. The prompt changes to 'nakim@makim:/usr/share/man/man1\$'. The user enters 'ls'. The output is a list of files in the directory, including '.1.gz', '7z.1.gz', '7za.1.gz', '7zr.1.gz', 'aa-enabled.1.gz', 'aa-exec.1.gz', 'aclocal-1.16.1.gz', 'aclocal.1.gz', 'aconnect.1.gz', 'add-apt-repository.1.gz', 'addr2line.1.gz', 'afm2pl.1.gz', 'afm2tfm.1.gz', 'aleph.1.gz', 'alsabat.1.gz', 'alsactl.1.gz', 'alsaloop.1.gz', 'alsamixer.1.gz', 'alsatplg.1.gz', 'alsaucm.1.gz', 'amidi.1.gz', 'amixer.1.gz', 'apg.1.gz', 'apgbfm.1.gz', 'aplay.1.gz', 'aplaymidi.1.gz', 'appres.1.gz', 'appstreamcli.1.gz', 'apropos.1.gz', 'apt-add-repository.1.gz', 'aptd.1.gz', 'aptfdcon.1.gz', 'apt-extracttemplates.1.gz', 'apt-ftparchive.1.gz', 'aptitude-create-state-bundle.1.gz', 'aptitude-run-state-bundle.1.gz', 'apt-sortpkgs.1.gz', 'apt-transport-http.1.gz', 'apt-transport-https.1.gz', 'apt-transport-mirror.1.gz', 'ar.1.gz', 'arch.1.gz', and 'archdetect.1.gz'.

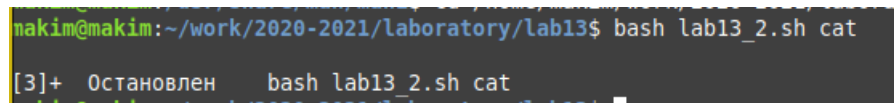
```
nakim@makim:~$ cd /usr/share/man/man1
nakim@makim:/usr/share/man/man1$ ls
'.1.gz'
7z.1.gz
7za.1.gz
7zr.1.gz
aa-enabled.1.gz
aa-exec.1.gz
aclocal-1.16.1.gz
aclocal.1.gz
aconnect.1.gz
add-apt-repository.1.gz
addr2line.1.gz
afm2pl.1.gz
afm2tfm.1.gz
aleph.1.gz
alsabat.1.gz
alsactl.1.gz
alsaloop.1.gz
alsamixer.1.gz
alsatplg.1.gz
alsaucm.1.gz
amidi.1.gz
amixer.1.gz
apg.1.gz
apgbfm.1.gz
aplay.1.gz
aplaymidi.1.gz
appres.1.gz
appstreamcli.1.gz
apropos.1.gz
apt-add-repository.1.gz
aptd.1.gz
aptfdcon.1.gz
apt-extracttemplates.1.gz
apt-ftparchive.1.gz
aptitude-create-state-bundle.1.gz
aptitude-run-state-bundle.1.gz
apt-sortpkgs.1.gz
apt-transport-http.1.gz
apt-transport-https.1.gz
apt-transport-mirror.1.gz
ar.1.gz
arch.1.gz
archdetect.1.gz
```

Рис. 2.9: Содержимое /man1



```
1 #!/bin/bash
2 cd /usr/share/man/man1
3 less $1*
```

Рис. 2.10: Исходный код



```
makim@makim:~/work/2020-2021/laboratory/lab13$ bash lab13_2.sh cat
[3]+ Остановлен bash lab13_2.sh cat
```

Рис. 2.11: Результат

```

.\" DO NOT MODIFY THIS FILE! It was generated by help2man 1.47.3.
.TH CAT "1" "September 2019" "GNU coreutils 8.30" "User Commands"
.SH NAME
cat \- concatenate files and print on the standard output
.SH SYNOPSIS
.B cat
[\fI\,OPTION\...\fR]... [\fI\,FILE\...\fR]...
.SH DESCRIPTION
.\" Add any additional description here
.PP
Concatenate FILE(s) to standard output.
.PP
With no FILE, or when FILE is \-, read standard input.
.TP
\fB\-A\fR, \fB\--show\-all\fR
equivalent to \fB\--vET\fR
.TP
\fB\-b\fR, \fB\--number\--nonblank\fR
number nonempty output lines, overrides \fB\-n\fR
.TP
\fB\-e\fR
equivalent to \fB\--vE\fR
.TP
\fB\-E\fR, \fB\--show\--ends\fR
display $ at end of each line
.TP
\fB\-n\fR, \fB\--number\fR
number all output lines
.TP
\fB\-s\fR, \fB\--squeeze\--blank\fR
suppress repeated empty output lines
.TP
\fB\-t\fR
equivalent to \fB\--vT\fR
.TP
\fB\-T\fR, \fB\--show\--tabs\fR
display TAB characters as ^I
.TP
\fB\-u\fR
(ignore)
.TP
\fB\-v\fR, \fB\--show\--nonprinting\fR
use ^ and M\ notation, except for LFD and TAB
.TP
\fB\--help\fR
display this help and exit
.TP
\fB\--version\fR
output version information and exit
.SH EXAMPLES
.TP
cat f \- g
Output f's contents, then standard input, then g's contents.
.TP
cat
Copy standard input to standard output.
cat.1.gz (file 1 of 4)

```

Рис. 2.12: Результат

- Используя встроенную переменную \$RANDOM, пишем командный файл, генерирующий случайную последовательность букв латинского алфавита. Учитываем, что \$RANDOM выдаёт псевдослучайные числа в диапазоне от 0 до 32767. (рис. 2.12 - 2.14)

```
#!/bin/bash
```

```
// создаём массив символов
```

```

declare -a array_en

// заполняем массив алфавитом
array_en=(a b c d e f g h i j k l m n o p q r s t u v w x y x)

// запускаем цикл до 10 итераций, что означает вывод 10 последовательностей
for (( i = 0; i < 10; i++))
do
    // запускаем цикл на случайное число итераций (до 30)
    // это значит, что наши последовательности будут случайного размера (от
    for (( j = 1; j < 15; j++ ))
    do
        // выводим случайный элемент массива chars
        // добавляем опцию -n, чтобы элементы выводились друг за другом
        echo -n ${array_en[$(($RANDOM % 26))]}
    done

    // переход на следующую строку
    echo
done

```



```

lab13_3.sh
~/work/2020-2021/laboratory/lab13
Сохранить

1 #!/bin/bash
2 declare -a array_en
3 array_en=(a b c d e f g h i j k l m n o p q r s t u v w x y x)
4 for (( i = 0; i < 10; i++ ))
5 do
6     for (( j = 0; j < 15; j++ ))
7     do
8         echo -n ${array_en[$(($RANDOM % 26))]}
9     done
10    echo
11 done

```

Рис. 2.13: Исходный код

```
makim@makim:~/work/2020-2021/laboratory/lab13$ bash lab13_3.sh
nqcjiaemmtmwaix
temptehxtvxickr
rtqifvmouxpucbt
cxduxdsxbvlwfo
dqqiclbhyjmxllr
jjnxgixhlylkjpmf
dgdstkelhgfokux
ndmquysixdmuwbj
yjxxwixmjxypwab
eejwebphaxjtox
```

Рис. 2.14: Результат

3. Выводы

Мы изучили основы программирования в оболочке ОС UNIX. Закрепили знания, полученные в прошлых работах. Научились писать более сложные командные файлы с использованием логических управляющих конструкций и циклов.

4. Термины

- Командный процессор (командная оболочка, интерпретатор команд shell) — это программа, позволяющая пользователю взаимодействовать с операционной системой компьютера.
- POSIX (Portable Operating System Interface for Computer Environments) — набор стандартов описания интерфейсов взаимодействия операционной системы и прикладных программ.
- Последовательность команд может быть помещена в текстовый файл. Такой файл называется командным.
- Флаги — это опции командной строки, обычно помеченные знаком минус; Например, для команды ls флагом может являться -F.
- Каталог, он же директория, (от английского Directory) – это объект в ФС (файловой системе), необходимый для того, чтобы упростить работу с файлами.
- Домашний каталог - каталог, предназначенный для хранения собственных данных пользователя Linux. Как правило, является текущим непосредственно после регистрации пользователя в системе.
- Команда - записанный по специальным правилам текст (возможно с аргументами), представляющий собой указание на выполнение какой-либо функций (или действий) в операционной системе.