

# Лабораторная работа № 14

Операционные системы

---

Казазаев Д. М.

Российский университет дружбы народов, Москва, Россия

## Информация

---

- Казазаев Даниил Михайлович
- Студент бакалавриата
- Российский университет дружбы народов
- [1132231427@rudn.ru]
- [https://github.com/KazazaevDaniil/study\\_2023-2024\\_os-intro](https://github.com/KazazaevDaniil/study_2023-2024_os-intro)

## Вводная часть

---

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

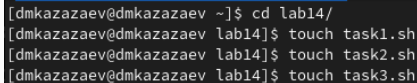
1. Выполнить задания лабораторной работы

## Основная часть

---

### Задание 1

Создаю файлы для заданий. (рис. 1)

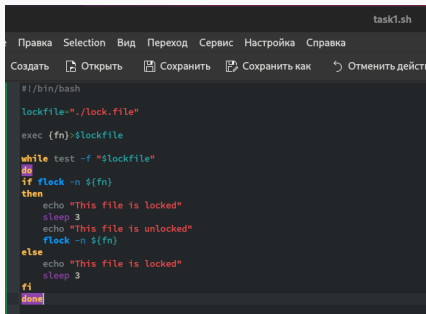
A terminal window with a dark background and light-colored text. It shows a sequence of four commands and their outputs. The first command changes the directory to 'lab14/'. The next three commands use the 'touch' command to create files named 'task1.sh', 'task2.sh', and 'task3.sh' in the current directory. The prompt changes from '~' to 'lab14/' after the first command.

```
[dmkazazaev@dmkazazaev ~]$ cd lab14/  
[dmkazazaev@dmkazazaev lab14]$ touch task1.sh  
[dmkazazaev@dmkazazaev lab14]$ touch task2.sh  
[dmkazazaev@dmkazazaev lab14]$ touch task3.sh
```

Рис. 1: Создание файлов для заданий



Пишу программу, которая реализует упрощенный механизм семафоров. (рис. 2)



```
task1.sh
Правка Selection Вид Переход Сервис Настройка Справка
Создать Открыть Сохранить Сохранить как Отменить действие

#!/bin/bash

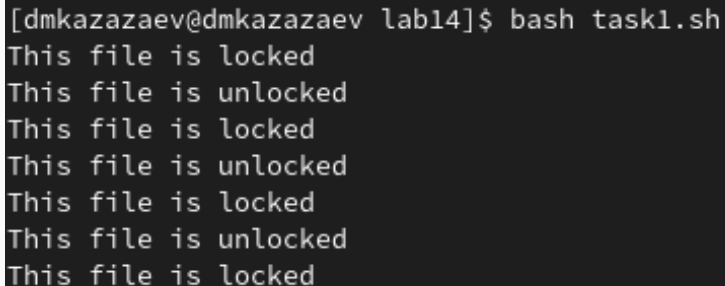
lockfile="./lock.file"

exec {fn}>$lockfile

while test -f "$lockfile"
do
if flock -n ${fn}
then
echo "This file is locked"
sleep 3
echo "This file is unlocked"
flock -n ${fn}
else
echo "This file is locked"
sleep 3
fi
done
```

Рис. 2: Код программы

После написания кода сохраняю файл и запускаю его. (рис. 3)



```
[dmkazazaev@dmkazazaev lab14]$ bash task1.sh  
This file is locked  
This file is unlocked  
This file is locked  
This file is unlocked  
This file is locked  
This file is unlocked  
This file is locked
```

Рис. 3: Запуск и результат

Листинг первой программы:

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

```
lockfile="./lock.file"
```

```
exec {fn}>$lockfile
```

```
while test -f "$lockfile"
```

```
do
```

```
if flock -n ${fn}
```

```
then
```

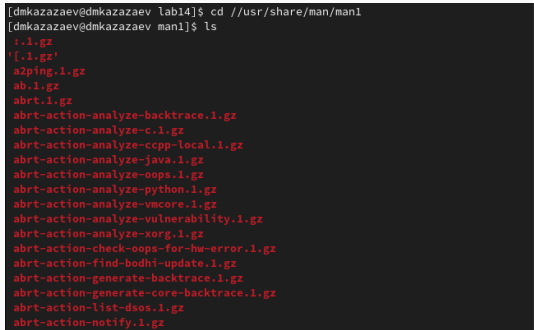
```
    echo "This file is locked"
```

```
    sleep 3
```

```
    echo "This file is unlocked"
    flock -n ${fn}
else
    echo "This file is locked"
    sleep 3
fi
done
```

## Задание 2

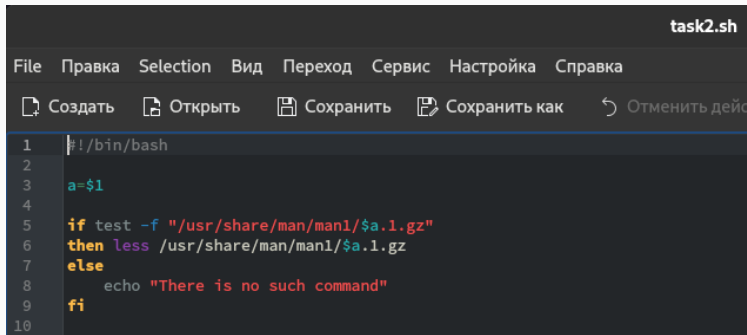
Перехожу в каталог `/usr/share/man/man1`, чтобы изучить архивы для успешного выполнения задания. (рис. 4)



```
[dmkazazaev@dmkazazaev lab14]$ cd /usr/share/man/man1
[dmkazazaev@dmkazazaev man1]$ ls
.:.gz
'(.gz'
a2ping.1.gz
ab.1.gz
abrt.1.gz
abrt-action-analyze-backtrace.1.gz
abrt-action-analyze-c.1.gz
abrt-action-analyze-ccpp-local.1.gz
abrt-action-analyze-java.1.gz
abrt-action-analyze-oops.1.gz
abrt-action-analyze-python.1.gz
abrt-action-analyze-vmcore.1.gz
abrt-action-analyze-vulnerability.1.gz
abrt-action-analyze-xorg.1.gz
abrt-action-check-oops-for-hw-error.1.gz
abrt-action-find-bodhi-update.1.gz
abrt-action-generate-backtrace.1.gz
abrt-action-generate-core-backtrace.1.gz
abrt-action-list-dsos.1.gz
abrt-action-notify.1.gz
```

Рис. 4: Каталог `/usr/share/man/man1`

Открываю второй файл и начинаю писать код, который реализует команду man. (рис. 5)



```
task2.sh
File  Правка  Selection  Вид  Переход  Сервис  Настройка  Справка
[Icon] Создать  [Icon] Открыть  [Icon] Сохранить  [Icon] Сохранить как  [Icon] Отменить дейс
1  #!/bin/bash
2
3  a=$1
4
5  if test -f "/usr/share/man/man1/$a.1.gz"
6  then less /usr/share/man/man1/$a.1.gz
7  else
8      echo "There is no such command"
9  fi
10
```

Рис. 5: Код программы

Сохраняю файл и запускаю его и смотрю справку к команде mkdir. (рис. 6)

```
MKDIR(1)                                User Commands                                MKDIR(1)

ESC[1mNAMEESC[0m
mkdir - make directories

ESC[1mSYNOPSISESC[0m
ESC[1mmkdir ESC[22m[ESC[4mOPTIONESC[24m]... ESC[4mDIRECTORYESC[24m...

ESC[1mDESCRIPTIONESC[0m
Create the DIRECTORY(ies), if they do not already exist.

Mandatory arguments to long options are mandatory for short options
too.

ESC[1m-mESC[22m, ESC[1m--modeESC[22m=ESC[4mMODEESC[0m
set file mode (as in chmod), not a=rwx - umask

ESC[1m-pESC[22m, ESC[1m--parentsESC[0m
no error if existing, make parent directories as needed, with
their file modes unaffected by any ESC[1m-m ESC[22moption.

ESC[1m-vESC[22m, ESC[1m--verboseESC[0m
print a message for each created directory

/usr/share/man/man1/mkdir.1.gz
```

Рис. 6: Результат программы

Листинг второй программы:

```
#!/bin/bash

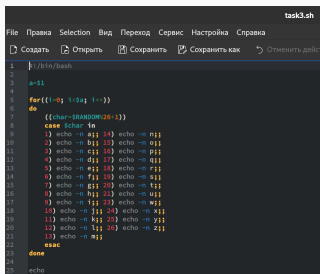
a=$1

if test -f "/usr/share/man/man1/$a.1.gz"
then less /usr/share/man/man1/$a.1.gz
else
    echo "There is no such command"
fi
```



## Задание 3

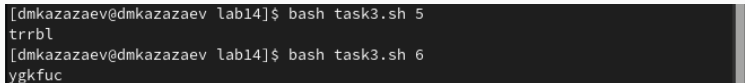
Открываю последний файл и пишу код, коотрый будет выводить в консоль случайную последовательность латинский букв. (рис. 7)



```
task3.sh
File Правка Selection Вид Переход Сервис Настройка Справка
Создать Открыть Сохранить Сохранить как Отменить дейст
1 #!/bin/bash
2
3 a=51
4
5 for((i=0; i<26; i++))
6 do
7     ((char=$((RANDOM%26-1)))
8     case $char in
9         0) echo -n a;; 14) echo -n n;;
10        1) echo -n b;; 15) echo -n o;;
11        2) echo -n c;; 16) echo -n p;;
12        3) echo -n d;; 17) echo -n q;;
13        4) echo -n e;; 18) echo -n r;;
14        5) echo -n f;; 19) echo -n s;;
15        6) echo -n g;; 20) echo -n t;;
16        7) echo -n h;; 21) echo -n u;;
17        8) echo -n i;; 22) echo -n v;;
18        9) echo -n j;; 23) echo -n w;;
19        10) echo -n k;; 24) echo -n x;;
20        11) echo -n l;; 25) echo -n y;;
21        12) echo -n m;; 26) echo -n z;;
22        *) echo -n ;;;
23    esac
24 done
25 echo
```

Рис. 7: Код программы

Проверяю работу программы. (рис. 8)



```
[dmkazazaev@dmkazazaev lab14]$ bash task3.sh 5  
trrbl  
[dmkazazaev@dmkazazaev lab14]$ bash task3.sh 6  
ygkfuc
```

A terminal window with a dark background. The prompt is [dmkazazaev@dmkazazaev lab14]\$. The first command is bash task3.sh 5, followed by the output trrbl. The second command is bash task3.sh 6, followed by the output ygkfuc.

Рис. 8: Результат выполнения программы

Листинг третьей программы:

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

```
a=$1
```

```
for((i=0; i<$a; i++))
```

```
do
```

```
    ((char=$RANDOM%26+1))
```

```
    case $char in
```

```
        1) echo -n a;; 14) echo -n n;;
```

```
        2) echo -n b;; 15) echo -n o;;
```

```
        3) echo -n c;; 16) echo -n p;;
```

```
        4) echo -n d;; 17) echo -n q;;
```

```
        5) echo -n e;; 18) echo -n r;;
```

```
8) echo -n h;; 21) echo -n u;;  
9) echo -n i;; 23) echo -n w;;  
10) echo -n j;; 24) echo -n x;;  
11) echo -n k;; 25) echo -n y;;  
12) echo -n l;; 26) echo -n z;;  
13) echo -n m;;
```

```
esac
```

```
done
```

```
echo
```

## Заключительная часть

---

- Получены практические программирования более сложных командных файлов с использованием логических управляющих конструкций и циклов.

В результате выполнения лабораторной работы я изучил основы программирования в оболочке ОС UNIX, научился писать более сложные командные файлы с использованием логических управляющих конструкций и циклов.