

# Программирование в командном процессоре ОС UNIX.

---

Одеджими Олуваколаде<sup>1</sup>

25 апреля, 2023, Москва, Россия

<sup>1</sup>Российский Университет Дружбы Народов

# Цели и задачи работы

---

## Цель лабораторной работы

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

# Задачи лабораторной работы

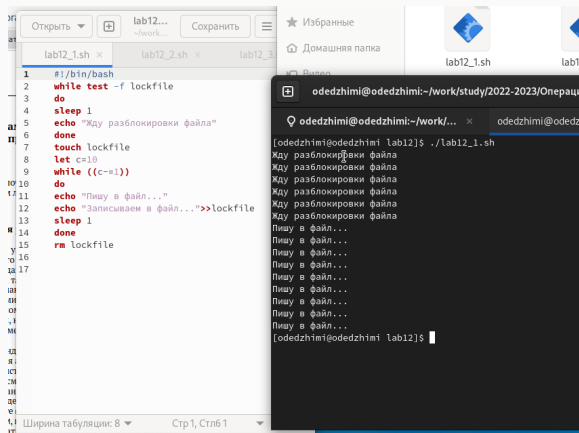
1 Выполнить 3 задания

# **Процесс выполнения лабораторной работы**

---

1. Написали командный файл, реализующий упрощённый механизм семафоров. Командный файл в течение некоторого времени  $t_1$  дожидается освобождения ресурса, выдавая об этом сообщение, а дождавшись его освобождения, использует его в течение некоторого времени  $t_2 < t_1$ , также выдавая информацию о том, что ресурс используется соответствующим командным файлом (процессом).

# Выполнение работы



The image shows a code editor window with a file named `lab12_1.sh` open. The script is a shell script that implements a file locking mechanism. It uses a `while test -f lockfile` loop to wait for a file named `lockfile` to be removed. Inside the loop, it prints a message in Russian, touches the file, increments a counter, and sleeps for 1 second. Once the file is removed, it prints another message and removes the file.

```
1 #!/bin/bash
2 while test -f lockfile
3 do
4     sleep 1
5     echo "Жду разблокировки файла"
6 done
7 touch lockfile
8 let c=c+1
9 while ((c<=1))
10 do
11     echo "Пишу в файл..."
12     echo "Записываем в файл...">>lockfile
13     sleep 1
14 done
15 rm lockfile
```

Below the code editor, a terminal window is open, showing the execution of the script. The prompt is `odedzhimi@odedzhimi:~/work/study/2022-2023/Операци`. The user has run `./lab12_1.sh`, and the terminal output shows the script's execution, including the waiting phase and the writing phase.

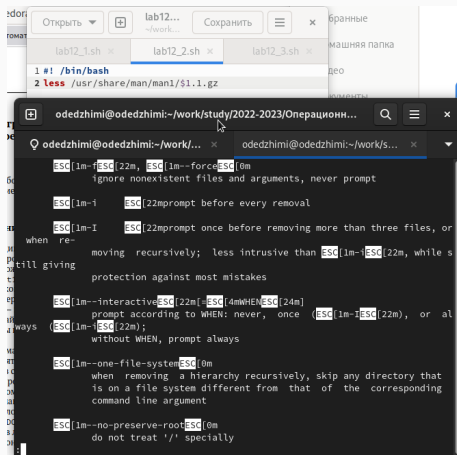
```
[odedzhimi@odedzhimi lab12]$ ./lab12_1.sh
Жду разблокировки файла
Жду разблокировки файла
Жду разблокировки файла
Жду разблокировки файла
Жду разблокировки файла
Жду разблокировки файла
Пишу в файл...
Пишу в файл...
Пишу в файл...
Пишу в файл...
Пишу в файл...
Пишу в файл...
Пишу в файл...
Пишу в файл...
Пишу в файл...
[odedzhimi@odedzhimi lab12]$
```

Рис. 1: Задание 1

2. Реализовали команду `man` с помощью командного файла. Изучили содержимое каталога `/usr/share/man/man1`. В нем находятся архивы текстовых файлов, содержащих справку по большинству установленных в системе программ и команд.



# Выполнение работы



The screenshot shows a terminal window with a file manager interface at the top. The file manager has tabs for 'lab12\_1.sh', 'lab12\_2.sh' (selected), and 'lab12\_3.sh'. Below the tabs, the terminal shows the following commands and output:

```
1 #! /bin/bash
2 less /usr/share/man/man1/$1.1.gz
```

The terminal window itself has a title bar that reads 'odedzhimi@odedzhimi:~/work/study/2022-2023/Операционн...'. The terminal content displays the help text for the 'less' command, with various options and their descriptions. The text is as follows:

```
ESC[1m--forceESC[0m
    ignore nonexistent files and arguments, never prompt

ESC[1m-iESC[22mprompt before every removal

ESC[1m-IESC[22mprompt once before removing more than three files, or
when re-
moving recursively; less intrusive than ESC[1m-IESC[22m, while s
till giving
protection against most mistakes

ESC[1m--interactiveESC[22m[+ESC[4mWHENESC[24m]
    prompt according to WHEN: never, once (ESC[1m-IESC[22m), or al
ways (ESC[1m-IESC[22m);
    without WHEN, prompt always

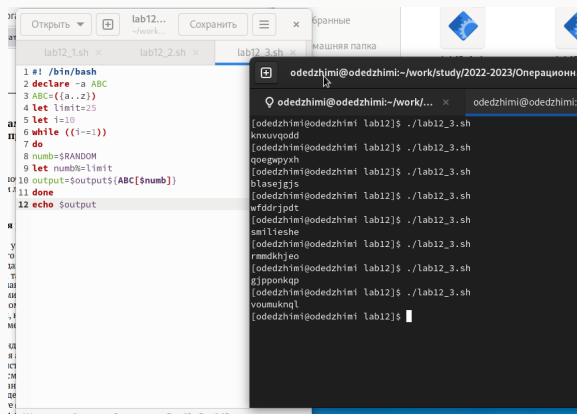
ESC[1m--one-file-systemESC[0m
    when removing a hierarchy recursively, skip any directory that
is on a file system different from that of the corresponding
command line argument

ESC[1m--no-preserve-rootESC[0m
    do not treat '/' specially
```

Рис. 2: Задание 2

3. Используя встроенную переменную `$RANDOM`, написали командный файл, генерирующий случайную последовательность букв латинского алфавита

# Выполнение работы



The screenshot shows a terminal window with a file manager interface at the top. The file manager has tabs for 'lab12\_1.sh', 'lab12\_2.sh', and 'lab12\_3.sh'. The 'lab12\_3.sh' tab is active, showing a shell script. The terminal window below shows the execution of this script, which generates a series of random strings.

```
1 #!/bin/bash
2 declare -a ABC
3 ABC={a..z}
4 let limit=25
5 let i=10
6 while ((i-->0))
7 do
8   numb=$((RANDOM%limit))
9   let numb=numb-1
10  output=${ABC[numb]}
11 done
12 echo $output
```

The terminal output shows the execution of the script, resulting in a series of random strings:

```
odedzhimi@odedzhimi:~/work/study/2022-2023/Операционн...
odedzhimi@odedzhimi:~/work/... x odedzhimi@odedzhimi:~
[odedzhimi@odedzhimi lab12]$ ./lab12_3.sh
knxuvqodd
[odedzhimi@odedzhimi lab12]$ ./lab12_3.sh
qoegwpyxh
[odedzhimi@odedzhimi lab12]$ ./lab12_3.sh
blasejgjs
[odedzhimi@odedzhimi lab12]$ ./lab12_3.sh
wfddrjpdtd
[odedzhimi@odedzhimi lab12]$ ./lab12_3.sh
smillieshe
[odedzhimi@odedzhimi lab12]$ ./lab12_3.sh
ramdkhjeo
[odedzhimi@odedzhimi lab12]$ ./lab12_3.sh
gjpponkqp
[odedzhimi@odedzhimi lab12]$ ./lab12_3.sh
voumuknql
[odedzhimi@odedzhimi lab12]$
```

Рис. 3: Задание 3

## **Выводы по проделанной работе**

---

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