

# Программирование в командном процессоре ОС UNIX.

---

Никита Власкин<sup>1</sup>

26 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

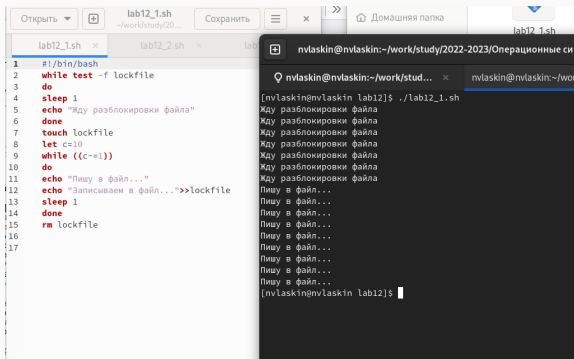
1 Выполнить 3 задания

# **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` and a terminal window running the script.

**Code Editor Content:**

```
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=10
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
16
17
```

**Terminal Content:**

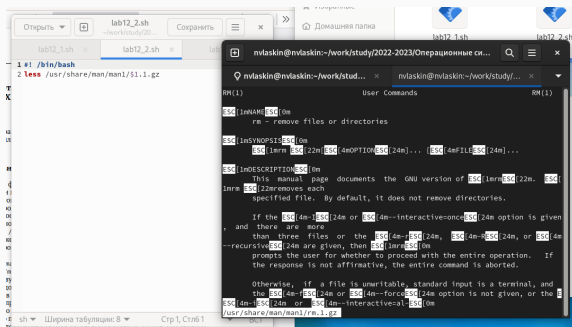
```
[nvlaskin@nvlaskin lab12]$ ./lab12_1.sh
Жду разблокировки файла
Жду разблокировки файла
Жду разблокировки файла
Жду разблокировки файла
Жду разблокировки файла
Жду разблокировки файла
Жду разблокировки файла
Пишу в файл...
Пишу в файл...
Пишу в файл...
Пишу в файл...
Пишу в файл...
Пишу в файл...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



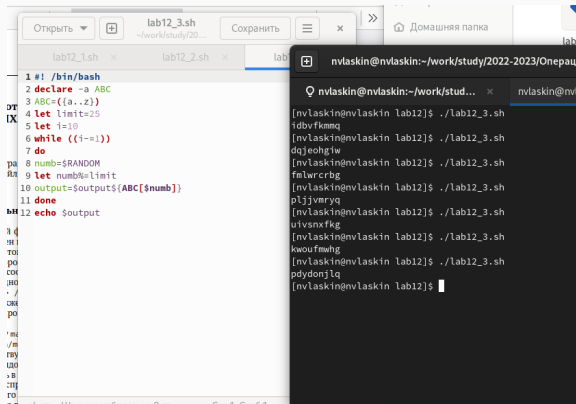
The screenshot shows a terminal window with two panes. The left pane shows a shell prompt where the user has entered `1 #! /bin/bash` and `2 less /usr/share/man/man1/rm.1.gz`. The right pane shows the manual page for the `rm` command, titled `rm(1) User Commands`. The manual page includes the command `rm - remove files or directories`, a synopsis `rm [OPTION]... FILE...`, and a description of the command's behavior, including options like `--recursive` and `--interactive`.

```
nvlskin@nvlskin:~/work/study/2022-2023/Операционные си...
RM(1) User Commands RM(1)
ESC[1NAMEESC[0m
rm - remove files or directories
ESC[1SYNOPSISESC[0m
ESC[1rmESC[22mESC[4mOPTIONESC[24m...ESC[4mFILEESC[24m...
ESC[1DESCRIPTIONESC[0m
This manual page documents the GNU version of ESC[1rmESC[22m.
ESC[1rmESC[22m removes each specified file. By default, it does not remove directories.
If the ESC[4m-ESC[24m or ESC[4m--interactive=onceESC[24m option is given
, and there are more
than three files or the ESC[4m-ESC[24m, ESC[4m--recursiveESC[24m, or ESC[4m
--recursiveESC[24m are given, then ESC[1rmESC[0m
prompts the user for whether to proceed with the entire operation. If
the response is not affirmative, the entire command is aborted.
Otherwise, if a file is unwritable, standard input is a terminal, and
the ESC[4m-fESC[24m or ESC[4m--forceESC[24m option is not given, or the
ESC[4m-iESC[24m or ESC[4m--interactive=alESC[0m
/usr/share/man/man1/rm.1.gz
```

Рис. 2: Задание 2

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

# Выполнение работы



The image shows a code editor window with a file named `lab12_3.sh` and a terminal window running the script. The code editor has tabs for `lab12_1.sh`, `lab12_2.sh`, and `lab12_3.sh`. The terminal window shows the execution of the script, which prints a sequence of 10 random characters.

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

```
[nvlaskin@nvaskin lab12]$ ./lab12_3.sh
idbvfkmq
[dq]eohgiw
fmlwrcrbg
pljjvmryq
uivsnxfkg
kwoufwmhg
pdydonjlq
```

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.