

# Операционные системы

Программирование в командном процессоре ОС UNIX.

---

Курбанов Рахман

16 июня 2025

Российский университет дружбы народов, Москва, Россия

## Цели и задачи работы

---

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

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

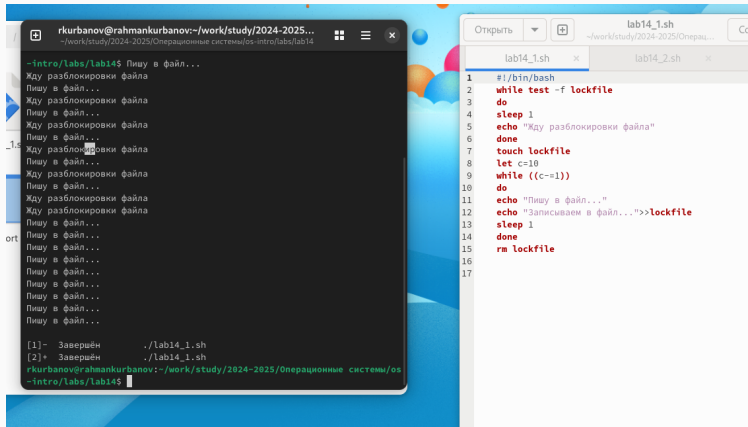
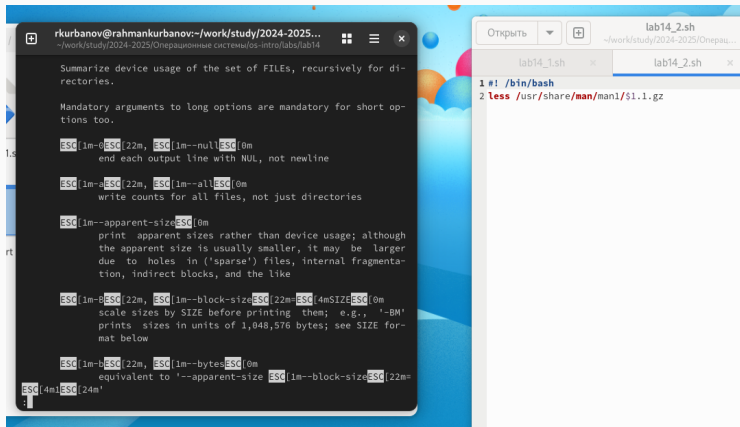


Рис. 1: Задание 1

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



```
rkurbanov@rahmkurbanov:~/work/study/2024-2025...  
~/work/study/2024-2025/Операционные системы/os-intro/labs/lab14
```

Summarize device usage of the set of FILES, recursively for directories.

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

`ESC[1m-0ESC[22m, ESC[1m--nullESC[0m`  
end each output line with NUL, not newline

`ESC[1m-2ESC[22m, ESC[1m--allESC[0m`  
write counts for all files, not just directories

`ESC[1m--apparent-sizeESC[0m`  
print apparent sizes rather than device usage; although the apparent size is usually smaller, it may be larger due to holes in ('sparse') files, internal fragmentation, indirect blocks, and the like

`ESC[1m-BESC[22m, ESC[1m--block-sizeESC[22m=ESC[4mSIZEESC[0m`  
scale sizes by SIZE before printing them; e.g., '-BM' prints sizes in units of 1,048,576 bytes; see SIZE format below

`ESC[1m-bESC[22m, ESC[1m--bytesESC[0m`  
equivalent to '--apparent-size ESC[1m--block-sizeESC[22m=ESC[4m1ESC[24m'

`ESC[4m1ESC[24m'`

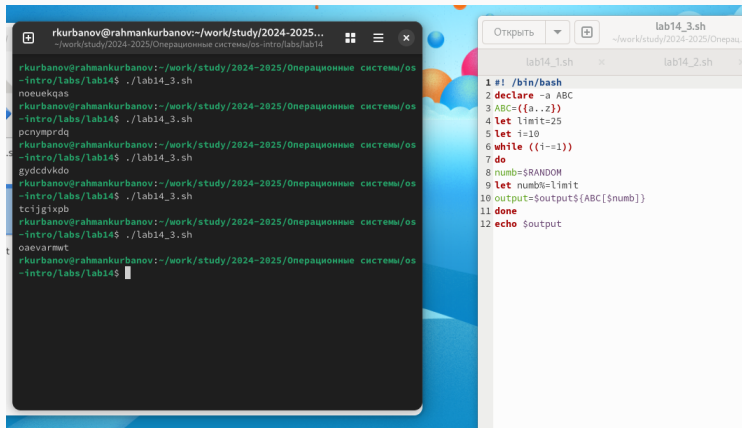
```
lab14_2.sh  
~/work/study/2024-2025/Операционные системы/os-intro/labs/lab14
```

```
1 #!/bin/bash  
2 less /usr/share/man/man1/$1.1.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 terminal window on the left and a script editor on the right. The terminal window displays the execution of a script named `lab14_3.sh` in a directory `~/work/study/2024-2025/Операционные системы/ос-intro/labs/lab14`. The script's output consists of a series of random characters: `noeuekqas`, `pcnymprdq`, `gydcdvkdo`, `tcijgixpb`, and `oaevarmwt`. The script editor on the right shows the content of `lab14_3.sh`, which is a Bash script that declares an array `ABC` of lowercase letters, sets a `limit` of 25, and then uses a `while` loop to generate 10 random characters from the array and store them in a variable `output`.

```
rkurbanov@rahmankurbanov:~/work/study/2024-2025/Операционные системы/ос-intro/labs/lab14$ ./lab14_3.sh
noeuekqas
rkurbanov@rahmankurbanov:~/work/study/2024-2025/Операционные системы/ос-intro/labs/lab14$ ./lab14_3.sh
pcnymprdq
rkurbanov@rahmankurbanov:~/work/study/2024-2025/Операционные системы/ос-intro/labs/lab14$ ./lab14_3.sh
gydcdvkdo
rkurbanov@rahmankurbanov:~/work/study/2024-2025/Операционные системы/ос-intro/labs/lab14$ ./lab14_3.sh
tcijgixpb
rkurbanov@rahmankurbanov:~/work/study/2024-2025/Операционные системы/ос-intro/labs/lab14$ ./lab14_3.sh
oaevarmwt
rkurbanov@rahmankurbanov:~/work/study/2024-2025/Операционные системы/ос-intro/labs/lab14$
```

```
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
```

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.