

# **Операционные системы**

Программирование в командном процессоре ОС UNIX.

---

Фатима Халилова

2025-11-14

# Содержание i

---

1. Цели и задачи работы
2. Процесс выполнения лабораторной работы
3. Выводы по проделанной работе

## **1. Цели и задачи работы**

---

## 1.1 Цель лабораторной работы

---

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

## 1.2 Задачи лабораторной работы

---

1 Выполнить 3 задания

## **2. Процесс выполнения лабораторной работы**

---

## 2.1 Выполнение 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).

## **2.2 Выполнение работы**

## Рисунок 1: Задание 1

## 2.3 Выполнение работы

---

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.

## 2.4 Выполнение работы

The screenshot shows a terminal window with the following content:

```
frhalolova@frhalolova:~/work/study/2025-2026/Операционные системы/os-intro/labs/lab14
ESC[1mDUESC[24m(1)
ESC[4mDUESC[24m(1)

ESC[1mNAMEESC[0m
du - estimate file space usage

ESC[1mSYNOPSISESC[0m
ESC[1mdu ESC[22m[ESC[4mOPTIONESC[24m]... [ESC[4mFILEESC[24m]...
ESC[1mdu ESC[22m[ESC[4mOPTIONESC[24m]... [ESC[4m--files0-from=ESC[0m

ESC[1mDESCRIPTIONESC[0m
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-aESC[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., '-8M' 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[1m-cESC[22m, ESC[1m--totalESC[0m
produce a grand total

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

In the background, there is another window titled "lab14\_2.sh" containing the following script:

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

Рисунок 2: Задание 2

## 2.5 Выполнение работы

---

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

## 2.6 Выполнение работы

The screenshot shows a terminal window and a file browser window. The terminal window displays a user's session on a Linux system. The user runs several shell scripts: `lab14_3.sh`, `aiocnnjt`, and `psfxamkrv`. The output of these scripts is mostly illegible noise. The user then runs `./lab14_3.sh` again, which outputs a sequence of letters: `htxnsryfj`. Finally, the user runs `./lab14_3.sh` one last time, which outputs a single character: `█`.

The file browser window shows a folder named `lab14_3` containing two files: `lab14_1.sh` and `lab14_2`. The `lab14_1.sh` file contains the following script:

```
#!/bin/bash
declare -a ABC
ABC=(a..z)
let limit=25
let i=18
while ((i>1))
do
numb=$RANDOM
let numb%=$limit
output=$output${ABC[$numb]}
done
echo $output
```

Рисунок 3: Задание 3

### **3. Выводы по проделанной работе**

---

### 3.1 Вывод

---

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