

## Лабораторная работа №12

---

Сиссе Мохамед Ламин;НММбд-01-22

<sup>1</sup>RUDN University, Moscow, Russian Federation

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

1. Написать командный файл, реализующий упрощённый механизм семафоров. Командный файл должен в течение некоторого времени  $t_1$  дожидаться освобождения ресурса, выдавая об этом сообщение, а дождавшись его освобождения, использовать его в течение некоторого времени  $t_2 < t_1$ , также выдавая информацию о том, что ресурс используется соответствующим командным файлом (процессом). Запустить командный файл в одном виртуальном терминале в фоновом режиме, перенаправив его вывод в другой (`> /dev/tty#`, где `#` — номер терминала куда перенаправляется вывод), в котором также запущен этот файл, но не фоновом, а в привилегированном режиме. Доработать программу так, чтобы имелась возможность взаимодействия трёх и более процессов.

2. Реализовать команду `man` с помощью командного файла. Изучите содержимое каталога `/usr/share/man/man1`. В нем находятся архивы текстовых файлов, содержащих справку по большинству установленных в системе программ и команд. Каждый архив можно открыть командой `less` сразу же просмотрев содержимое справки. Командный файл должен получать в виде аргумента командной строки название команды и в виде результата выдавать справку об этой команде или сообщение об отсутствии справки, если соответствующего файла нет в каталоге `man1`.
3. Используя встроенную переменную `$RANDOM`, напишите командный файл, генерирующий случайную последовательность букв латинского алфавита. Учтите, что `$RANDOM` выдаёт псевдослучайные числа в диапазоне от 0 до 32767.

Командный процессор (командная оболочка, интерпретатор команд shell) — это программа, позволяющая пользователю взаимодействовать с операционной системой компьютера. В операционных системах типа UNIX/Linux наиболее часто используются следующие реализации командных оболочек:

- оболочка Борна (Bourne shell или sh) — стандартная командная оболочка UNIX/Linux, содержащая базовый, но при этом полный набор функций;
- C-оболочка (или csh) — надстройка на оболочкой Борна, использующая C-подобный синтаксис команд с возможностью сохранения истории выполнения команд;
- оболочка Корна (или ksh) — напоминает оболочку C, но операторы управления программой совместимы с операторами оболочки Борна;

POSIX (Portable Operating System Interface for Computer Environments) — набор стандартов описания интерфейсов взаимодействия операционной системы и прикладных программ. Стандарты POSIX разработаны комитетом IEEE (Institute of Electrical and Electronics Engineers) для обеспечения совместимости различных UNIX/Linux-подобных операционных систем и переносимости прикладных программ на уровне исходного кода. POSIX-совместимые оболочки разработаны на базе оболочки Корна (**Prog:bash?**).

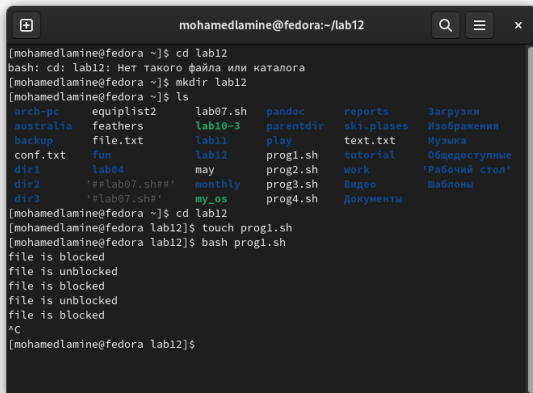
1. Написать командный файл, реализующий упрощённый механизм семафоров. Командный файл должен в течение некоторого времени  $t_1$  дожидаться освобождения ресурса, выдавая об этом сообщение, а дождавшись его освобождения, использовать его в течение некоторого времени  $t_2 < t_1$ , также выдавая информацию о том, что ресурс используется соответствующим командным файлом (процессом). Запустить командный файл в одном виртуальном терминале в фоновом режиме, перенаправив его вывод в другой (`> /dev/tty#`, где `#` — номер терминала куда перенаправляется вывод), в котором также запущен этот файл, но не фоновом, а в привилегированном режиме. Доработать программу так, чтобы имелась возможность взаимодействия трёх и более процессов. (рис. (fig:001?; fig:002?))

## Выполнение лабораторной работы

Рис. 1: Текст первой программы



# Выполнение лабораторной работы

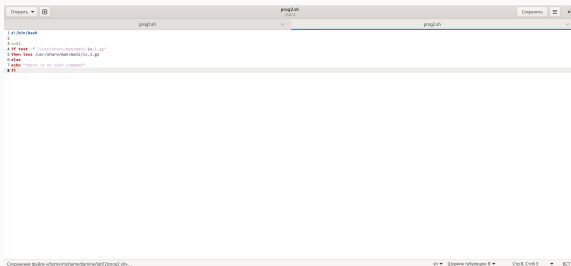


```
mohamedlamine@fedora:~/lab12
[mohamedlamine@fedora ~]$ cd lab12
bash: cd: lab12: Нет такого файла или каталога
[mohamedlamine@fedora ~]$ mkdir lab12
[mohamedlamine@fedora ~]$ ls
arch-pc      equiplist2    lab07.sh      pandoc      reports      Загрузки
australia    feathers      lab10-3       parentdir   ski.plases   Изображения
backup       file.txt      lab11         play        text.txt     Музыка
conf.txt     fun          lab12         prog1.sh    tutorial     Общедоступные
dir1         lab04        may           prog2.sh    work         'Рабочий стол'
dir2         '##lab07.sh##' monthly       prog3.sh    Видео       Шаблоны
dir3         '#lab07.sh#' my_os         prog4.sh    Документы

[mohamedlamine@fedora ~]$ cd lab12
[mohamedlamine@fedora lab12]$ touch prog1.sh
[mohamedlamine@fedora lab12]$ bash prog1.sh
file is blocked
file is unblocked
file is blocked
file is unblocked
file is blocked
^C
[mohamedlamine@fedora lab12]$
```

Рис. 2: Результат

2. Реализовать команду `man` с помощью командного файла. Изучите содержимое каталога `/usr/share/man/man1`. В нем находятся архивы текстовых файлов, содержащих справку по большинству установленных в системе программ и команд. Каждый архив можно открыть командой `less` сразу же просмотрев содержимое справки. Командный файл должен получать в виде аргумента командной строки название команды и в виде результата выдавать справку об этой команде или сообщение об отсутствии справки, если соответствующего файла нет в каталоге `man1`. (рис. (fig:003?; fig:004?; fig:005?))



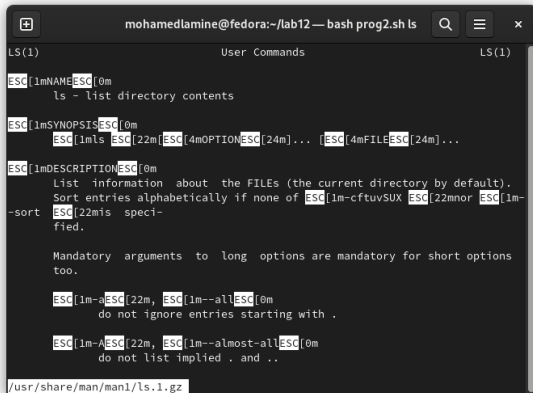
The image shows a screenshot of a code editor window. The window has a title bar with "Открыть" (Open) and "Сохранить" (Save) buttons. The editor displays the text of a program in a light blue font on a white background. The text is as follows:

```
1 #! /usr/bin/bash
2
3 exit
4 if test -f /usr/bin/bash; then
5 then bash /usr/bin/bash 1 2 3
6 else
7 echo "there is no such command"
8 fi
```

The status bar at the bottom of the window shows the file path "Сопровождающие материалы к лабораторной работе 1", the editor version "v1", the language "C++", and the file encoding "UTF-8".

Рис. 3: Текст второй программы

# Выполнение лабораторной работы



```
mohamedlamine@fedora:~/lab12 — bash prog2.sh ls
LS(1)                                User Commands                                LS(1)

ESC[1mNAMEESC[0m
    ls - list directory contents

ESC[1mSYNOPSISESC[0m
    ESC[1mls ESC[22mESC[4mOPTIONESC[24m]... ESC[4mFILEESC[24m]...

ESC[1mDESCRIPTIONESC[0m
    List information about the FILES (the current directory by default).
    Sort entries alphabetically if none of ESC[1m-cftuvSUX ESC[22mnor ESC[1m-
-sort ESC[22mis speci-
    fied.

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

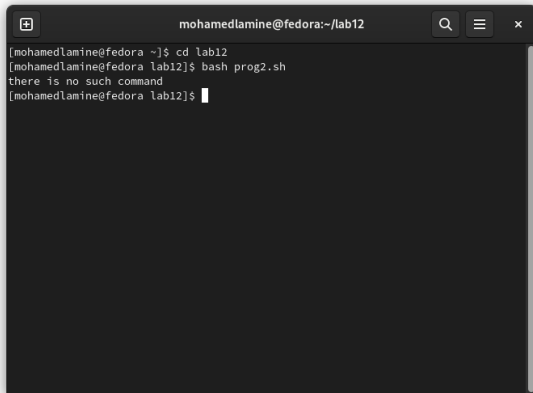
    ESC[1m-aESC[22m, ESC[1m--allESC[0m
        do not ignore entries starting with .

    ESC[1m-AESC[22m, ESC[1m--almost-allESC[0m
        do not list implied . and ..

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

Рис. 4: Результат

# Выполнение лабораторной работы

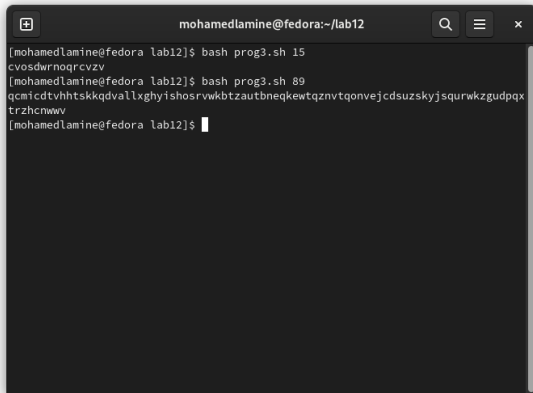
A terminal window with a dark background. The title bar shows the user 'mohamedlamine@fedora' and the current directory '~/lab12'. The terminal content shows the user navigating to the 'lab12' directory and attempting to run a script named 'prog2.sh'. The system responds with an error message: 'there is no such command'.

```
mohamedlamine@fedora: ~/lab12
[mohamedlamine@fedora ~]$ cd lab12
[mohamedlamine@fedora lab12]$ bash prog2.sh
there is no such command
[mohamedlamine@fedora lab12]$
```

Рис. 5: Результат



# Выполнение лабораторной работы

A terminal window with a dark background. The title bar shows 'mohamedlamine@fedora:~/lab12' and standard window controls. The terminal content shows a command being executed and its output.

```
mohamedlamine@fedora lab12]$ bash prog3.sh 15
cvosdwrnoqrcvzv
mohamedlamine@fedora lab12]$ bash prog3.sh 89
qcmicdtvhhtskkqdvallxghyishosrvwkbztautbneqkewtqznvtqonvejcdsuzskyjsqurwkzgudppx
trzhcnwvv
mohamedlamine@fedora lab12]$
```

Рис. 7: Результат

В процессе выполнения этой лабораторной работы я продолжил осваивать программирование на `bash`.



Спасибо за внимание!