

# Лабораторная работа No 12. Программирование в командном процессоре ОС UNIX. Расширенное программирование

---

Горяйнова Алёна

Российский университет дружбы народов, Москва, Россия

## Информация

---

- Гоояйнова Алёна
- студентка
- Российский университет дружбы народов

## Цель работы

---

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

## Задание

---

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

## Выполнение лабораторной работы

---



## Выполнение лабораторной работы

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.

```
[aagoryayjnova@fedora ~]$ gedit lab_12.1.sh
[aagoryayjnova@fedora ~]$ chmod +x lab_12.1.sh
[aagoryayjnova@fedora ~]$ bash lab_12.1.sh
File is blocked
File is unlocked
File is blocked
```

```
1 #!/bin/bash
2
3 lockfile="./lock.file"
4 exec {fn}>$lockfile
5
6 while test -f "$lockfile"
7 do
8   if flock -n ${fn}
9   then
10     echo "File is blocked"
11     sleep 5
12     echo "File is unlocked"
13     flock -u ${fn}
14   else
15     echo "File is blocked"
16     sleep 5
17     fi
18 done
```

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

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`.



The image shows a terminal window with a title bar containing a dropdown menu with the text "Открыть", a plus icon, and the filename "\*lab\_12.2.sh" with a tilde symbol below it. The terminal content is a shell script with the following lines:

```
1 #!/bin/bash
2
3 a=$1
4 if test -f "/usr/share/man/man1/$a.1.gz"
5 then less /usr/share/man/man1/$a.1.gz
6 else
7 echo "There is no such command"
8 fi
```

Рис. 2: Код

```
aagoryayjnova@fedora:~ — bash lab_12.2.sh ls
```

aagoryayjnova@fedora:~ — bash l... x aagoryayjnova@fedora:~ x

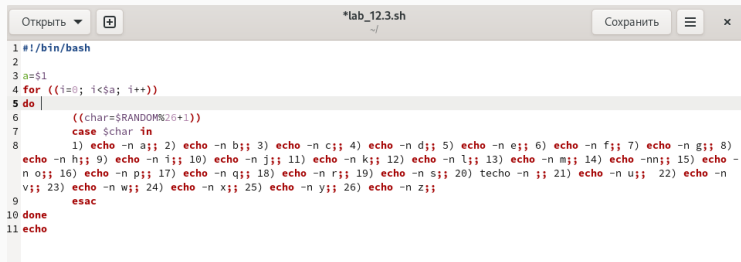
LS(1)	User Commands	LS(1)
<p>ESC[1mNAMEESC[0m ls - list directory contents</p> <p>ESC[1mSYNOPSISESC[0m ESC[1m[ls ESC[22m[ESC[4mOPTIONESC[24m)... [ESC[4mFILEESC[24m)...</p> <p>ESC[1mDESCRIPTIONESC[0m List information about the FILES (the current directory by default). Sort entries alphabetically if none of ESC[1m-cftuvSUX ESC[22mnor ESC[1m- -sort ESC[22mis speci- fied.</p> <p>Mandatory arguments to long options are mandatory for short options too.</p> <p>ESC[1m-aESC[22m, ESC[1m--allESC[0m do not ignore entries starting with .</p> <p>ESC[1m-AESC[22m, ESC[1m--almost-allESC[0m do not list implied . and ..</p> <p>/usr/share/man/man1/ls.1.gz</p>		

```
[aagoryayjnova@fedora ~]$ gedit lab_12.2.sh
[aagoryayjnova@fedora ~]$ chmod +x lab_12.2.sh
[aagoryayjnova@fedora ~]$ bash lab_12.2.sh ls
[aagoryayjnova@fedora ~]$ bash lab_12.2.sh kjfws
There is no such command
[aagoryayjnova@fedora ~]$
```

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.

---

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.

A screenshot of a terminal window titled '\*lab\_12.3.sh' with a tilde icon. The window contains a shell script that generates a random sequence of 26 lowercase letters. The script uses a for loop to iterate 26 times, each time generating a random number with \$RANDOM%26 and mapping it to a letter from 'a' to 'z' using a case statement. The letters are printed with a delay of 1 second using 'echo -n' and 'sleep 1'.

```
1 #!/bin/bash
2
3 a=$1
4 for ((i=0; i<$a; i++))
5 do
6     ((char=$RANDOM%26+1))
7     case $char in
8         1) echo -n a;; 2) echo -n b;; 3) echo -n c;; 4) echo -n d;; 5) echo -n e;; 6) echo -n f;; 7) echo -n g;; 8)
echo -n h;; 9) echo -n i;; 10) echo -n j;; 11) echo -n k;; 12) echo -n l;; 13) echo -n m;; 14) echo -n n;; 15) echo -
n o;; 16) echo -n p;; 17) echo -n q;; 18) echo -n r;; 19) echo -n s;; 20) echo -n t;; 21) echo -n u;; 22) echo -n
v;; 23) echo -n w;; 24) echo -n x;; 25) echo -n y;; 26) echo -n z;;
9         esac
10    done
11    echo
```

Рис. 3: Код



```
[aagoryayjnova@fedora ~]$ gedit lab_12.3.sh  
e[aagoryayjnova@fedora ~]$ chmod +x lab_12.3.sh  
[aagoryayjnova@fedora ~]$ bash lab_12.3.sh 14  
jsmesxxamyuoxz
```

## Выводы

---

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

...