

# Презентация по лабораторной работе №12

Дисциплина: операционные системы

---

Лобанова П. И.

29.04.2023

Российский университет дружбы народов, Москва, Россия

## Информация

---

- Лобанова Полина Иннокентьевна
- учащаяся факультета физико-математических и естественных наук
- учащаяся направления “Компьютерные и информационные науки”
- Российский университет дружбы народов
- [polla-2004@mail.ru](mailto:polla-2004@mail.ru)

## Цель работы

---

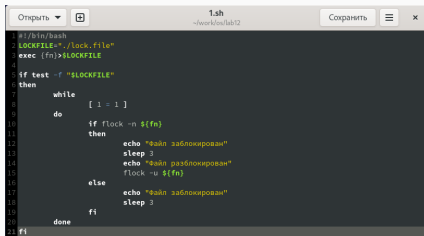
Изучить основы программирования в оболочке ОС UNIX. Научиться писать более сложные командные файлы с использованием логических управляющих конструкций и циклов.

## Ход работы

---

# 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).

A screenshot of a terminal window titled '1.sh' with a path '~/.work/loz/lab1/2'. The window contains a shell script that implements a semaphore mechanism. The script starts with a shebang, sets a lock file, and defines a function 'fn' that checks for the lock file. If the lock file exists, it enters a loop where it prints 'Файл заблокирован' and sleeps for 3 seconds. If the lock file does not exist, it prints 'Файл разблокирован', acquires the lock, prints 'Файл заблокирован', and sleeps for 3 seconds. The script ends with a 'fi' statement.

```
#!/bin/bash
2 LOCKFILE="./lock.file"
3 exec {fn}>$LOCKFILE
4
5 if test -f "$LOCKFILE"
6 then
7     while
8     [ 1 = 1 ]
9     do
10         if flock -n ${fn}
11         then
12             echo "Файл заблокирован"
13             sleep 3
14             echo "Файл разблокирован"
15             flock -u ${fn}
16         else
17             echo "Файл заблокирован"
18             sleep 3
19         fi
20     done
21 fi
```

Рис. 1: Текст скрипта.

```
[pilobanova@fedora lab12]$ bash 1.sh  
Файл заблокирован  
Файл разблокирован  
Файл заблокирован
```

Рис. 2: *Результат.*



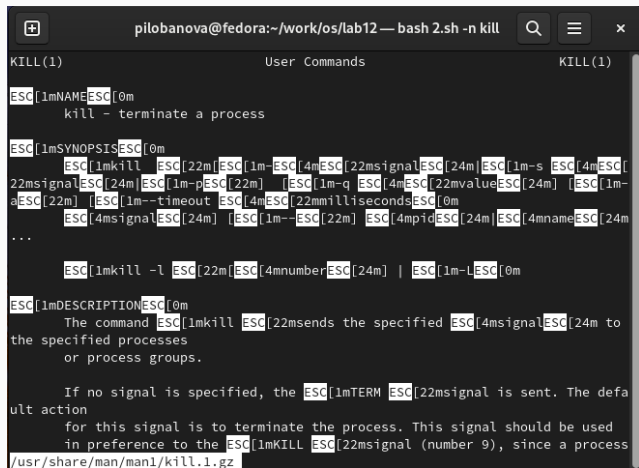
## Задание 2

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`.



```
Открыть [иконка] 2.sh ~/work/os/lab12 Сохранить [меню] [крестик]
1 #!/bin/bash
2 command=""
3
4 while getopts :n: opt
5 do
6     case $opt in
7         n) command="$OPTARG";;
8         esac
9     done
10
11 if test -f "/usr/share/man/man1/$command.1.gz"
12 then less /usr/share/man/man1/$command.1.gz
13
14 else
15     echo "no such command"
16
17 fi
```

Рис. 3: Текст скрипта.



```
pilobanova@fedora:~/work/os/lab12 — bash 2.sh -n kill
KILL(1)                                User Commands                                KILL(1)

ESC[1mNAMEESC[0m
    kill - terminate a process

ESC[1mSYNOPSISESC[0m
    ESC[1mkill ESC[22m[ESC[1m-ESC[4mESC[22msignalESC[24m]ESC[1m-s ESC[4mESC[22msignalESC[24m]ESC[1m-pESC[22m] ESC[1m-q ESC[4mESC[22mvalueESC[24m] ESC[1m-aESC[22m] ESC[1m--timeout ESC[4mESC[22mmillisecondsESC[0m
    ESC[4msignalESC[24m] ESC[1m--ESC[22m] ESC[4mpidESC[24m]ESC[4mnameESC[24m
    ...

    ESC[1mkill -l ESC[22m[ESC[4mnumberESC[24m] | ESC[1m-LESC[0m

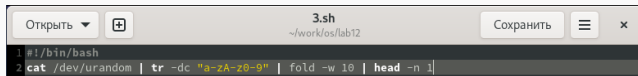
ESC[1mDESCRIPTIONESC[0m
    The command ESC[1mkill ESC[22msends the specified ESC[4msignalESC[24m to
    the specified processes
    or process groups.

    If no signal is specified, the ESC[1mTERM ESC[22msignal is sent. The default action
    for this signal is to terminate the process. This signal should be used
    in preference to the ESC[1mKILL ESC[22msignal (number 9), since a process
    /usr/share/man/man1/kill.1.gz
```

Рис. 4: Результат.

## Задание 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.



The image shows a terminal window titled "3.sh" with the path "~/work/os/lab12". The window has buttons for "Открыть" (Open), "Сохранить" (Save), and a close button. The terminal content shows two lines of a script:

```
1 #!/bin/bash
2 cat /dev/urandom | tr -dc "a-zA-z0-9" | fold -w 10 | head -n 1
```

Рис. 5: Текст скрипта.

```
[pilobanova@fedora lab12]$ bash 3.sh  
gfy\0mZeMd
```

Рис. 6: *Результат.*

## Контрольные вопросы

---

1. Найдите синтаксическую ошибку в следующей строке: `1 while [$1 != "exit"]`
2. Как объединить (конкатенация) несколько строк в одну?
3. Найдите информацию об утилите `seq`. Какими иными способами можно реализовать её функционал при программировании на `bash`?
4. Какой результат даст вычисление выражения `$((10/3))`?
5. Укажите кратко основные отличия командной оболочки `zsh` от `bash`.
6. Проверьте, верен ли синтаксис данной конструкции `1 for ((a=1; a <= LIMIT; a++))`
7. Сравните язык `bash` с какими-либо языками программирования. Какие преимущества у `bash` по сравнению с ними? Какие недостатки?

## Вывод

---

Я научилась писать более сложные командные файлы с использованием логических управляющих конструкций и циклов.