

## Лабораторная работа №12

Программирование в командном процессоре ОС UNIX. Расширенное программирование

---

Зарифбеков А. П.

26 апреля 2023

Российский университет дружбы народов, Москва, Россия

## Информация

---

- Зарифбеков Амир Пайшанбиевич
- Студент НБИбд-01-22
- Российский университет дружбы народов

Актуально для тех кто хочет освоить программирование в командном процессе ОС UNIX.

- Презентация как текст
- Программное обеспечение для создания презентаций
- Входные и выходные форматы презентаций

- Изучить основы программирования в оболочке ОС UNIX/Linux. Научиться писать небольшие командные файлы.

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.

```
apzarifbekov@dk8n72 ~/work/study/2022-2023/Операционные системы/os-intro/labs/lab12 $ touch lab12.sh
apzarifbekov@dk8n72 ~/work/study/2022-2023/Операционные системы/os-intro/labs/lab12 $ chmod +x lab12.sh
apzarifbekov@dk8n72 ~/work/study/2022-2023/Операционные системы/os-intro/labs/lab12 $ ./lab12.sh
lock
work
work
work
work
work
work
apzarifbekov@dk8n72 ~/work/study/2022-2023/Операционные системы/os-intro/labs/lab12 $
```

```
~/work/study/2022-2023/Операционн  
1 #!/bin/bash  
2 lockfile="./lockfile"  
3 exec {fn}>$lockfile  
4 echo "lock"  
5 until flock -n ${fn}  
6 do  
7     echo "not lock"  
8     sleep 1  
9     flock -n ${fn}  
10 done  
11 for ((i=0;i<=5; i++))  
12 do  
13     echo "work"  
14     sleep 1  
15 done
```

Рис. 2: как он работает



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`.

```
work
apzarifbekov@dk8n72 ~/work/study/2022-2023/Операционные системы/os-intro/labs/lab12 $ touch lab12-1.sh
apzarifbekov@dk8n72 ~/work/study/2022-2023/Операционные системы/os-intro/labs/lab12 $ chmod +x lab12-1.sh
apzarifbekov@dk8n72 ~/work/study/2022-2023/Операционные системы/os-intro/labs/lab12 $ ./lab12-1.sh
```

Рис. 3: написали скрипт

A screenshot of a terminal window with a light gray background. On the left side, there is a vertical sidebar with a dark gray background and some white text, including a minus sign and the letters 'Им'. The terminal area shows three lines of text: the first line is '1 #!/bin/bash' in blue, the second line is '2 cd /usr/share/man/man1' in red, and the third line is '3 less \$1\*' in red. The third line is highlighted with a light gray background.

```
1 #!/bin/bash
2 cd /usr/share/man/man1
3 less $1*
```

Рис. 4: то как он работает

```
411toppm User Manual(0) 411toppm User Manual(0)
NAME
    411toppm - convert Sony Mavica .411 image to PPM

SYNOPSIS
    411toppm [-width width] [-height height] [411file]

DESCRIPTION
    This program is part of Netpbm(1).

    411toppm reads a .411 file, such as from a Sony Mavic camera, and converts it to a PPM image as output.

    Output is to Standard Output.

    The originator of this program and decipherer of the .411 format, Steve Allen <sla@alumni.caltech.edu>, has this to say about the utility of this program: "There's so little image in a 64x48 thumbnail (especially when you have the full size JPG file) that the only point in doing this was to answer the implicit challenge posed by the manual stating that only the camera can use these files."

OPTIONS
    In addition to the options common to all programs based on libnetpbm (most notably -quiet, see Common Options (index.html#commonoptions) ), 411toppm recognizes the following command line options:

    All options may be abbreviated to the shortest unique prefix.

    -width The width (number of columns) of the input image. Default is 64.
```

Рис. 5: то как он работает

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

```
[1]# Остановлен ./lab12-1.sh
apzarifbekov@dk8n72 ~/work/study/2022-2023/Операционные системы/os-intro/labs/lab12 $ touch lab12-2.sh
apzarifbekov@dk8n72 ~/work/study/2022-2023/Операционные системы/os-intro/labs/lab12 $ chmod +x lab12-2.sh
h
apzarifbekov@dk8n72 ~/work/study/2022-2023/Операционные системы/os-intro/labs/lab12 $ ./lab12-2.sh

10 random words:
bcchcggbbb
1
fbbbcihhcc
2
cbcdcbchicb
3
cbcjbbdccb
4
jbjbccbbcb
5
ecbdbiccdb
6
dbjfdcccg
7
dcdcebbcf
8
jbdcbcedcb
9
bccbddhhbb
10
apzarifbekov@dk8n72 ~/work/study/2022-2023/Операционные системы/os-intro/labs/lab12 $
```

Рис. 6: смотрим как он работает



The image shows a terminal window with a light gray header bar. On the left, there is a button labeled 'Открыть' (Open) with a dropdown arrow and a plus icon. In the center, the text 'lab12-z.sh' is displayed above the file path '~/.work/study/2022-2023/Операционные системы/os-intro/labs/lab12'. On the right, there is a button labeled 'Сохранить' (Save) and a hamburger menu icon. The main area of the terminal contains a shell script with line numbers 1 through 12. The script starts with a shebang, sets variables M, c, and d, echoes a message, and then enters a while loop that generates random words using a combination of RANDOM and tr commands. The script ends with a 'done' statement.

```
1 #!/bin/bash
2 M=10
3 c=1
4 d=1
5 echo
6 echo "10 random words:"
7 while (($c!=($M+1)))
8 do
9     echo $(for((i=1;i<=10;i++)); do printf '%s' "${RANDOM:0:1}"; done) | tr '[:0-9:]' '[:a-z:]'
10    echo $d
11    ((c+=1))
12    ((d+=1))
13 done
```

Рис. 7: смотрим как он работает



- Я изучил основы программирования в оболочке ОС UNIX/Linux. Научился писать небольшие командные файлы