

# **Лабораторная работа №12**

**Операционные системы**

Савурская Полина

# Содержание

1	Цель работы	4
2	Задание	5
3	Выполнение лабораторной работы	7
4	Выводы	12

## Список иллюстраций

3.1	создание файла . . . . .	7
3.2	пишем код . . . . .	8
3.3	запускаем файл . . . . .	8
3.4	создание файла . . . . .	9
3.5	пишем код . . . . .	9
3.6	запуск файла file2.sh . . . . .	9
3.7	результат file2.sh . . . . .	9
3.8	создание файла file11.sh . . . . .	10
3.9	пишем код . . . . .	10
3.10	запуск файла file11.sh . . . . .	11

# 1 Цель работы

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

## 2 Задание

1. Написать командный файл, реализующий упрощенный 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 трех 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 латинско-

го алфавита. Учтите, что \$RANDOM выдаёт псевдослучайные числа в диапазоне от 0 до 32767.

### 3 Выполнение лабораторной работы

1. Создаю файл file1.sh. Задаем ему необходимые разрешения.

```
[pasavurskaya@username ~]$ touch file1.sh  
[pasavurskaya@username ~]$ chmod +x file1.sh  
[pasavurskaya@username ~]$
```

Рис. 3.1: создание файла

2. В файле file1.sh пишем код.

```
#!/bin/bash

lockfile = "./lockfile"
exec {fn}>$lockfile
echo "lock"
until flock -n ${fn}
do
    echo "not lock"
    sleep 1
    flock -n ${fn}
done
for ((i=0; i<=5; i++))
do
    echo "work"
    sleep 1
done
```

Рис. 3.2: пишем код

3. Запускаем.

```
flock: requires file descriptor, file or directory
not lock
flock: requires file descriptor, file or directory
flock: requires file descriptor, file or directory
not lock
flock: requires file descriptor, file or directory
flock: requires file descriptor, file or directory
not lock
flock: requires file descriptor, file or directory
flock: requires file descriptor, file or directory
not lock
```

Рис. 3.3: запускаем файл

4. Создаю файл file2.sh. Задаю ему необходимые разрешения.



```
[pasavurskaya@username ~]$ touch file2.sh
[pasavurskaya@username ~]$ chmod +x file2.sh
[pasavurskaya@username ~]$
```

Рис. 3.4: создание файла

5. Открываем эти файлы и пишем там нужные коды.

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

Рис. 3.5: пишем код

6. Запускаем файл file2.sh. Смотрим результат.

```
[pasavurskaya@username ~]$ ./file2.sh less
```

Рис. 3.6: запуск файла file2.sh

```
LESS(1)                                General Commands Manual                                LESS(1)
ESC[1mNAMEESC[0m
less - opposite of more
ESC[1mSYNOPSISESC[0m
ESC[1mless -?ESC[0m
ESC[1mless --helpESC[0m
ESC[1mless -VESC[0m
```

Рис. 3.7: результат file2.sh

7. Создаем файл file11.sh. Задаем ему необходимые разрешения.

```
[pasavurskaya@username ~]$ touch file11.sh
[pasavurskaya@username ~]$ chmod +x file11.sh
[pasavurskaya@username ~]$
```

Рис. 3.8: создание файла file11.sh

8. Открываем этот файл и пишем там нужный код.

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

Рис. 3.9: пишем код

9. Запускаем файл file11.sh.

```
[pasavurskaya@username ~]$ ./file11.sh  
  
10 random words:  
dbdceihcgc  
1  
bdjbbcbcb  
2  
cgbcibdicc  
3  
bbbccjdccc  
4  
dcbfccbjcc  
5  
cccbccbceg  
6  
dcbcbccccg  
7  
cbdbbjbdb  
8  
bdbjcbcdhc  
9  
dcbcbcccc
```

Рис. 3.10: запуск файла file11.sh

## 4 Выводы

Я изучить основы программирования в оболочке ОС UNIX. Научилась писать более сложные командные файлы с использованием логических управляющих конструкций и циклов. :::