

Лабораторная работа №12

Операционные системы

Савурская П. А.

23 апреля 2023

Российский университет дружбы народов, Москва, Россия

- Савурская Полина
- № студ.билета 1132222827
- НБИбд-04-22
- Российский университет дружбы народов

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

Создаю файл file1.sh. Задаем ему необходимые разрешения.

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

Рис. 1: создание файла

Выполнение лабораторной работы. Шаг 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
```

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: запускаем файл

Создаю файл file2.sh. Задаю ему необходимые разрешения.

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

Рис. 4: создание файла

Открываем файл и пишем там нужный код.

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

Рис. 5: пишем код

Выполнение лабораторной работы. Шаг 6.

Запускаем файл file2.sh. Смотрим результат.

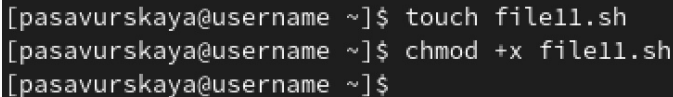
```
[pasavurskaya@username ~]$ ./file2.sh less
```

Рис. 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
    ESC[1mless -wESC[0m
```

Рис. 7: результат file2.sh

Создаем файл file11.sh. Задаем ему необходимые разрешения.

A terminal window with a dark background and light-colored text. It shows three lines of commands and their prompts. The first line is '[pasavurskaya@username ~]\$ touch file11.sh'. The second line is '[pasavurskaya@username ~]\$ chmod +x file11.sh'. The third line is '[pasavurskaya@username ~]\$' with no output.

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

Рис. 8: создание файла file11.sh

Выполнение лабораторной работы. Шаг 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. 9: пишем код

Выполнение лабораторной работы. Шаг 9.

Запускаем файл file11.sh.

```
[pasavurskaya@username ~]$ ./file11.sh
```

```
10 random words:
```

```
dbdceihcgc
```

```
1
```

```
bdjbbcbcb
```

```
2
```

```
cgbci b d i c c
```

```
3
```

```
bbbccjdccc
```

```
4
```

```
dcbfccbjcc
```

```
5
```

```
cccbccbceg
```

```
6
```

```
dcbcbccccg
```

```
7
```

```
cbedbbjbdb
```

```
8
```

```
bdbjcbdbc
```

```
9
```

```
dcbcbccccg
```

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