

Отчет по лабораторной работе №12

Шмаков Максим¹

2022, 28 мая , Москва

¹RUDN University, Moscow, Russian Federation

Цель работы

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

Выполнение лабораторной работы

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.

Создаю файл number1.sh и пишу в нем скрипт. (рис. [-@fig:001]) (рис. [-@fig:002])

A terminal window with a dark background. The prompt is [mpshmakov@fedora ~]\$. The command touch number1.sh is entered.

```
[mpshmakov@fedora ~]$ touch number1.sh
```

Рис. 1: рис. 1

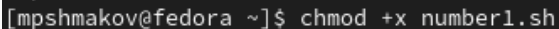
Создаю файл number1.sh и пишу в нем скрипт. (рис. [-@fig:001]) (рис. [-@fig:002])



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

Рис. 2: рис. 2

Даю право на исполнение и проверяю работу скрипта. Все работает правильно. (рис. [-@fig:003]) (рис. [-@fig:004])



```
[mpshmakov@fedora ~]$ chmod +x number1.sh
```

Рис. 3: рис. 3

Даю право на исполнение и проверяю работу скрипта. Все работает правильно. (рис. [-@fig:003]) (рис. [-@fig:004])

```
[mpshmakov@fedora ~]$ ./number1.sh  
lock  
work  
work  
work  
work  
work  
work
```

Рис. 4: рис. 4

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`.

Просмотрел содержимое каталога /usr/share/man/man1. (рис. [-@fig:005])

```
[mpshmakov@fedora ~]$ cd /usr/share/man/man1
[mpshmakov@fedora man1]$ ls
:.1.gz
' [.1.gz'
a2ping.1.gz
ab.1.gz
abrt.1.gz
abrt-action-analyze-backtrace.1.gz
abrt-action-analyze-c.1.gz
abrt-action-analyze-ccpp-local.1.gz
```

Рис. 5: рис. 5

Создаю файл number2.sh и пишу в нем скрипт. (рис. [-@fig:006]) (рис. [-@fig:007])

A terminal window with a black background and white text. The prompt is [mpshmakov@fedora ~]\$. The command touch number2.sh has been entered.

```
[mpshmakov@fedora ~]$ touch number2.sh
```

Рис. 6: рис. 6

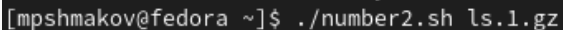
Создаю файл number2.sh и пишу в нем скрипт. (рис. [-@fig:006]) (рис. [-@fig:007])



```
1 cd /usr/share/man/man1
2 less $1
```

Рис. 7: рис. 7

Даю право на исполнение, запускаю файл и проверяю результат. Все верно. (рис. [-@fig:008]) (рис. [-@fig:009]) (рис. [-@fig:010])

A terminal window with a black background and white text. The prompt is [mpshmakov@fedora ~]\$ and the command being executed is ./number2.sh ls.1.gz. The output of the command is partially visible below the prompt.

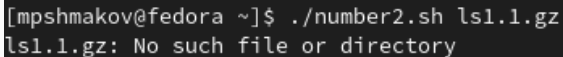
```
[mpshmakov@fedora ~]$ ./number2.sh ls.1.gz
```

Рис. 8: рис. 8

Даю право на исполнение, запускаю файл и проверяю результат. Все верно. (рис. [-@fig:008]) (рис. [-@fig:009]) (рис. [-@fig:010])

Рис. 9: рис. 9

Даю право на исполнение, запускаю файл и проверяю результат. Все верно. (рис. [-@fig:008]) (рис. [-@fig:009]) (рис. [-@fig:010])



```
[mpshmakov@fedora ~]$ ./number2.sh ls1.1.gz  
ls1.1.gz: No such file or directory
```

Рис. 10: рис. 10

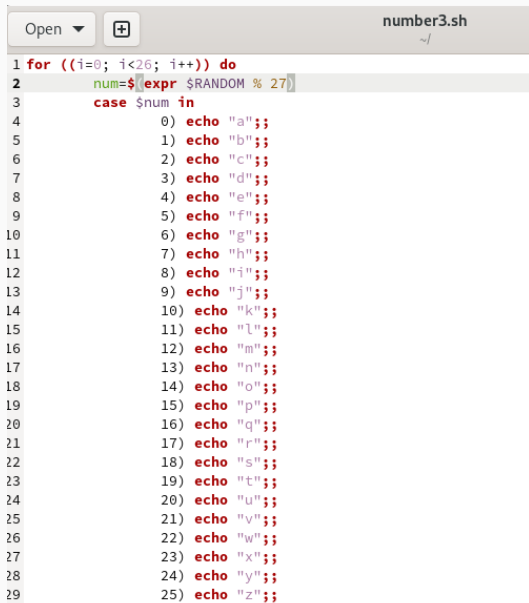
Используя встроенную переменную `$RANDOM`, напишите командный файл, генерирующий случайную последовательность букв латинского алфавита. Учтите, что `$RANDOM` выдаёт псевдослучайные числа в диапазоне от 0 до 32767.

Создаю файл number3.sh, пишу в нем скрипт. (рис. [-@fig:011]) (рис. [-@fig:012])

```
[mpshmakov@fedora ~]$ touch number3.sh
```

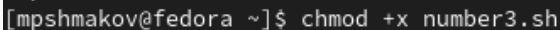
Рис. 11: рис. 11

Создаю файл number3.sh, пишу в нем скрипт. (рис. [-@fig:011]) (рис. [-@fig:012])



```
1 for ((i=0; i<26; i++)) do
2     num=$((expr $RANDOM % 27))
3     case $num in
4         0) echo "a";;
5         1) echo "b";;
6         2) echo "c";;
7         3) echo "d";;
8         4) echo "e";;
9         5) echo "f";;
10        6) echo "g";;
11        7) echo "h";;
12        8) echo "i";;
13        9) echo "j";;
14        10) echo "k";;
15        11) echo "l";;
16        12) echo "m";;
17        13) echo "n";;
18        14) echo "o";;
19        15) echo "p";;
20        16) echo "q";;
21        17) echo "r";;
22        18) echo "s";;
23        19) echo "t";;
24        20) echo "u";;
25        21) echo "v";;
26        22) echo "w";;
27        23) echo "x";;
28        24) echo "y";;
29        25) echo "z";;
```

Даю право на исполнение и проверяю результат. На всякий случай, запускаю 2 раза, чтобы убедиться что выводятся разные буквы. Все верно. (рис. [-@fig:0013]) (рис. [-@fig:0014])

A terminal window with a dark background. The prompt is [mpshmakov@fedora ~]\$. The command entered is chmod +x number3.sh.

```
[mpshmakov@fedora ~]$ chmod +x number3.sh
```

Рис. 13: рис. 13

Даю право на исполнение и проверяю результат. На всякий случай, запускаю 2 раза, чтобы убедиться что выводятся разные буквы. Все верно. (рис. [-@fig:0013]) (рис. [-@fig:0014])

```
[mpshmakov@fedora ~]$ ./number3.sh
o
g
p
e
k
n
h
w
s
g
v
a
q
j
b
w
t
q
j
d
j
c
o
g
l
x
[mpshmakov@fedora ~]$ ./number3.sh
e
x
k
q
u
a
y
w
```

Выводы

В ходе работы я научился писать более сложные командные файлы с использованием логических управляющих конструкций и циклов.