

Отчет по лабораторной работе №12

**Программирование в командном процессоре ОС UNIX. Расширенное
программирование**

Максим Сергеевич Белов

Содержание

1	Цель работы	5
2	Задание	6
3	Теоретическое введение	8
4	Выполнение лабораторной работы	9
5	Выводы	12
6	Контрольные вопросы	13

Список иллюстраций

4.1	Скрипт 1	9
4.2	Проверка скрипта 1	9
4.3	Скрипт 2	10
4.4	Проверка скрипта 2	10
4.5	Скрипт 3	10
4.6	Проверка скрипта 3	11

List of Tables

3.1 Описание команд для работы с командными файлами 8

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

32767.

3 Теоретическое введение

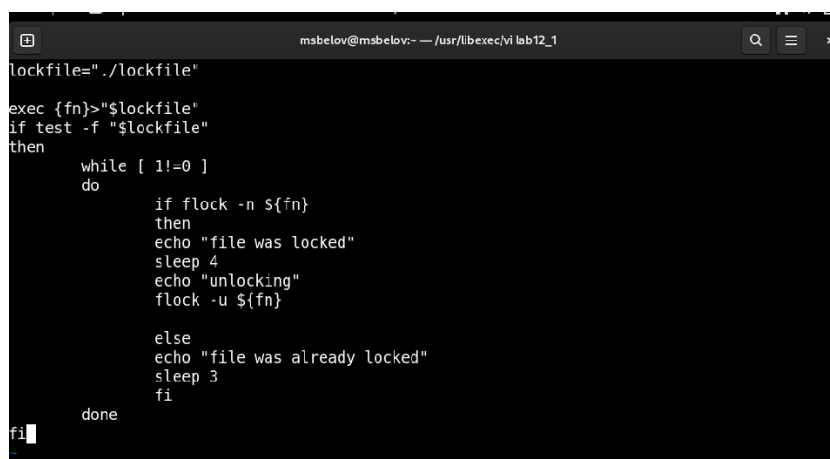
В табл. 3.1 приведено краткое описание команд для работы с командными файлами.

Таблица 3.1: Описание команд для работы с командными файлами

Команды	
Команда	Описание
<code>getopts</code>	Осуществляет синтаксический анализ командной строки, выделяя флаги, и используется для объявления переменных.

4 Выполнение лабораторной работы

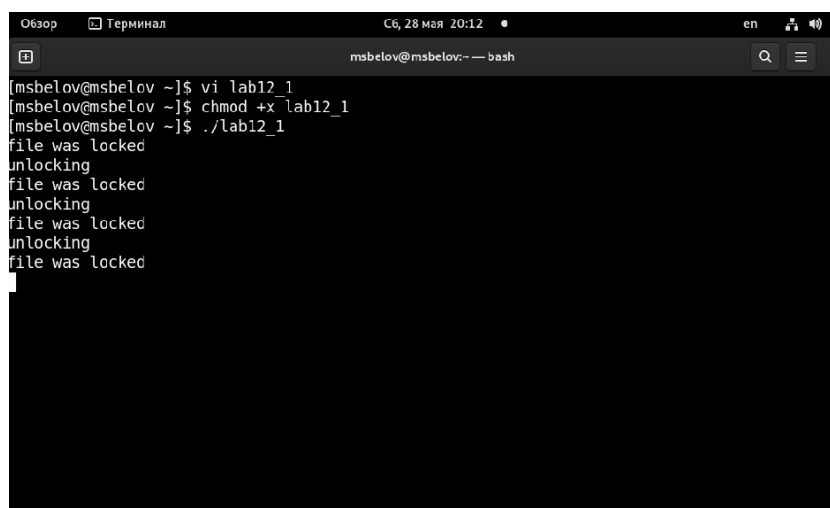
1. Скрипт 1 (Рис. 4.1)



```
lockfile="./lockfile"
exec {fn}>"$lockfile"
if test -f "$lockfile"
then
    while [ 1!=0 ]
    do
        if flock -n $fn
        then
            echo "file was locked"
            sleep 4
            echo "unlocking"
            flock -u $fn
        else
            echo "file was already locked"
            sleep 3
        fi
    done
fi
```

Рис. 4.1: Скрипт 1

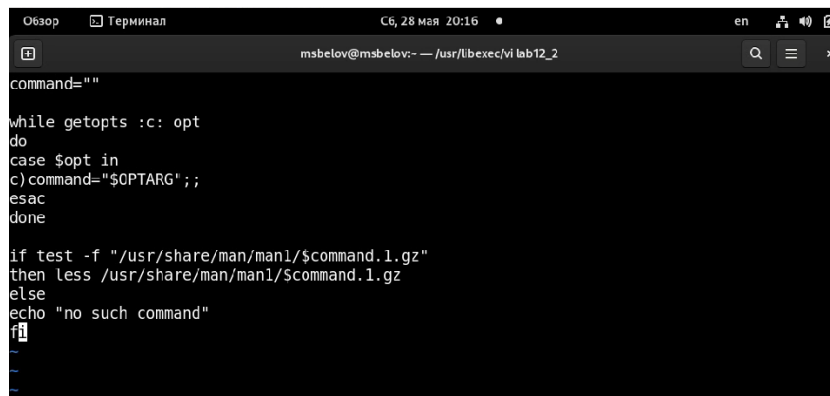
Сделаем файл lab12_1 исполняемым и проверим его работу (Рис. 4.2)



```
msbelov@msbelov:~$ vi lab12_1
msbelov@msbelov:~$ chmod +x lab12_1
msbelov@msbelov:~$ ./lab12_1
file was locked
unlocking
file was locked
unlocking
file was locked
unlocking
file was locked
```

Рис. 4.2: Проверка скрипта 1

2. Скрипт 2 (Рис. 4.3)

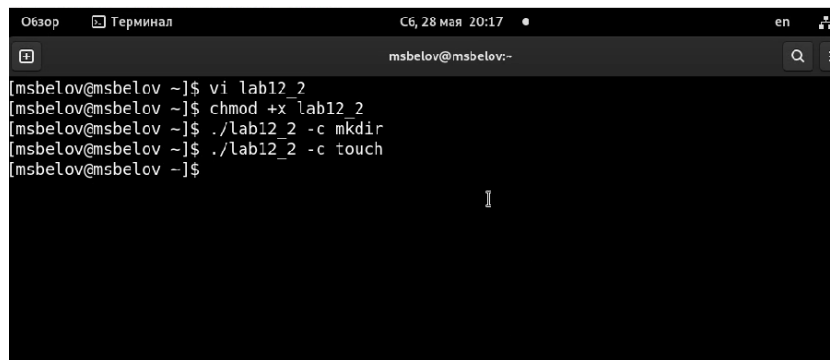


```
command=""
while getopts :c: opt
do
case $opt in
c) command="$OPTARG";;
esac
done

if test -f "/usr/share/man/man1/$command.1.gz"
then less /usr/share/man/man1/$command.1.gz
else
echo "no such command"
fi
```

Рис. 4.3: Скрипт 2

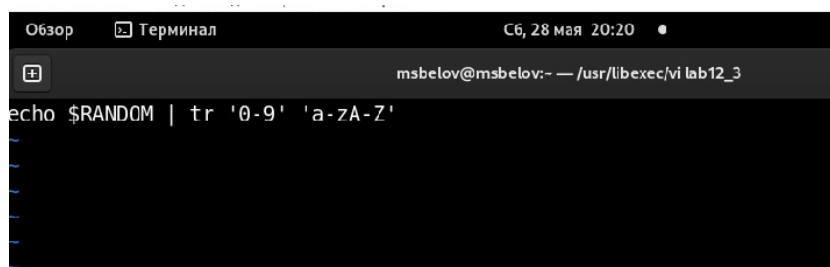
Сделаем файл lab12_2 исполняемым и проверим его работу (Рис. 4.4)



```
[msbelov@msbelov ~]$ vi lab12_2
[msbelov@msbelov ~]$ chmod +x lab12_2
[msbelov@msbelov ~]$ ./lab12_2 -c mkdir
[msbelov@msbelov ~]$ ./lab12_2 -c touch
[msbelov@msbelov ~]$
```

Рис. 4.4: Проверка скрипта 2

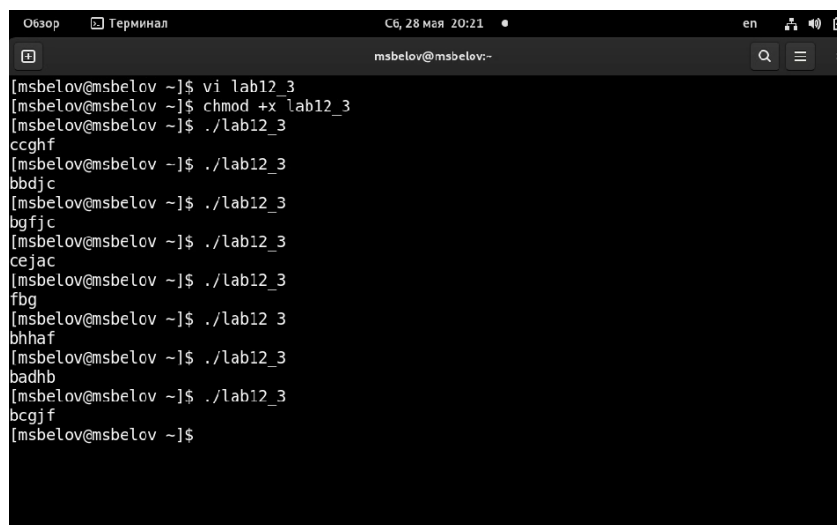
3. Скрипт 3 (Рис. 4.5)



```
echo $RANDOM | tr -dc 'a-zA-Z'
```

Рис. 4.5: Скрипт 3

Сделаем файл lab12_3 исполняемым и проверим его работу
(Рис. 4.6)



```
msbelov@msbelov:~$ vi lab12_3
msbelov@msbelov:~$ chmod +x lab12_3
msbelov@msbelov:~$ ./lab12_3
ccghf
msbelov@msbelov:~$ ./lab12_3
bbdj c
msbelov@msbelov:~$ ./lab12_3
bgfjc
msbelov@msbelov:~$ ./lab12_3
cejac
msbelov@msbelov:~$ ./lab12_3
fbg
msbelov@msbelov:~$ ./lab12_3
bhhaf
msbelov@msbelov:~$ ./lab12_3
badhb
msbelov@msbelov:~$ ./lab12_3
bcgj f
msbelov@msbelov:~$
```

Рис. 4.6: Проверка скрипта 3

5 Выводы

В ходе работы я изучил основы программирования в оболочке ОС UNIX. Научился писать более сложные командные файлы с использованием логических управляющих конструкций и циклов.

6 Контрольные вопросы

1. Между скобками должны быть пробелы. В противном случае скобки и рядом стоящие символы будут восприниматься как одно целое.
2. С помощью +=
3. seq - выдает последовательность чисел.
4. 5
5. Zsh очень сильно упрощает работу. Но существуют различия. Например, в zsh после for обязательно вставлять пробел, нумерация массивов в zsh начинается с 1 (что не особо удобно на самом деле).
6. Верен.
7. Bash позволяет очень легко работать с файловой системой без лишних конструкций (в отличие от обычного языка программирования). Но относительно обычных языков программирования bash очень сжат. Тот же Си имеет гораздо более широкие возможности для разработчика.