

# Программирование в командном процессоре ОС UNIX. Расширенное программирование

---

Лебедева Ольга Андреевна

<sup>1</sup>RUDN University, Moscow, Russian Federation

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

`$RANDOM` – внутренняя функция Bash (не константа), которая возвращает псевдослучайные целые числа в диапазоне 0 - 32767. Функция `$RANDOM` не должна использоваться для генерации ключей шифрования.

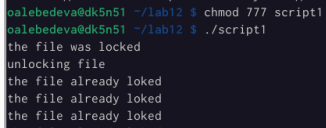
1. Написали командный файл, реализующий упрощённый механизм семафоров. Командный файл должен в течение некоторого времени  $t_1$  дожидаться освобождения ресурса, выдавая об этом сообщение, а дождавшись его освобождения, использовать его в течение некоторого времени  $t_2 < t_1$ , также выдавая информацию о том, что ресурс используется соответствующим командным файлом (процессом). Запустили 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. Доработали программу так, чтобы имелась возможность взаимодействия трёх и более процессов. (рис. 1) (рис. 2) (рис. 3)

```
oalebedeva@dk5n51 ~ $ mkdir lab12
oalebedeva@dk5n51 ~ $ cd lab12
oalebedeva@dk5n51 ~/lab12 $ touch script1
oalebedeva@dk5n51 ~/lab12 $ ls
script1
oalebedeva@dk5n51 ~/lab12 $ vi script1
```

Figure 1: Создание папки и файла, запуск редактора vi

```
lockfile="./locking.file"
exec {fn}>"$lockfile"
if test -f "$lockfile"
then
    while [ 1!=0 ]
    do
        if flock -n ${fn}
        then
            echo "the file was locked"
            sleep 5
            echo "unlocking file"
            flock -u ${fn}
        else
            echo "the file already locked"
            sleep 3
        fi
    done
fi
```

Figure 2: Текст программы1



```
oalebedeva@dk5n51 ~/lab12 $ chmod 777 script1
oalebedeva@dk5n51 ~/lab12 $ ./script1
the file was locked
unlocking file
the file already locked
the file already locked
the file already locked
```

Figure 3: Работа программы1

2 . Реализовали команду `man` с помощью командного файла. Изучили 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`. (рис. 4) (рис. 6) (рис. 5)

```

command=""
while getopts :n: opt
do
case $opt in
n)command="$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 command"
fi

```

Figure 4: Текст программы2

```

oalebedeva@dk5n51 ~/lab12 $ ./script2 -n cd

```

Figure 5: Вызов программы2

```

CD(1P)                                POSIX Programmer's Manual                                CD(1P)
PROLOG
This manual page is part of the POSIX Programmer's Manual. The Linux
implementation of this interface may differ (consult the corresponding
Linux manual page for details of Linux behavior), or the interface may
not be implemented on Linux.
NAME
cd - change the working directory
SYNOPSIS
cd [-L|-P] [directory]

```

Figure 6: Работа программы2



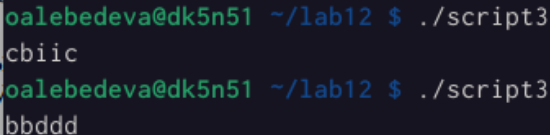
## Задание 3

3. Используя встроенную переменную \$RANDOM, написали командный файл, генерирующий случайную последовательность букв латинского алфавита. Учли, что \$RANDOM выдаёт псевдослучайные числа в диапазоне от 0 до 32767. (рис. 7) (рис. 8)



```
echo $RANDOM | tr '0-9' 'a-z'
```

Figure 7: Текст программы3



```
oalebedeva@dk5n51 ~/lab12 $ ./script3
cbiic
oalebedeva@dk5n51 ~/lab12 $ ./script3
bbddd
```

Figure 8: Работа программы3

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