

# Лабораторная работа №12. Программирование в командном процессоре ОС UNIX. Расширенное программирование

Дисциплина: Операционные системы

---

Стариков Д. А., студент НПИбд-02-22

29 апреля 2023

Российский университет дружбы народов, Москва, Россия

## 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

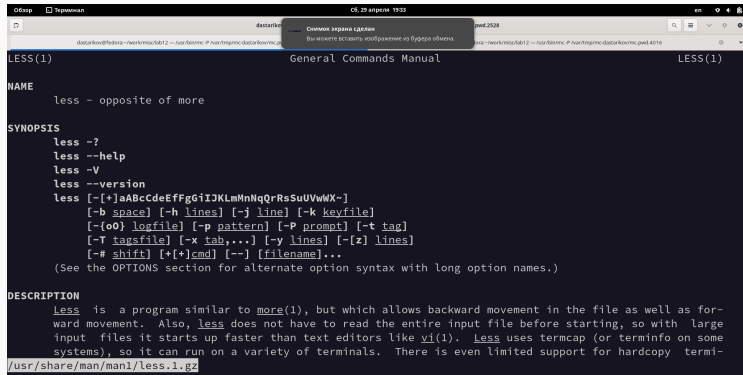
---

## Задание 1

```
1  #!/bin/bash
2  COMMAND=$1
3  FILE=`find /usr/share/man/man1 -type f -name $COMMAND.1.gz -print -quit`
4  if [ -n "$FILE" ]
5  then
6      less -R /usr/share/man/man1/$COMMAND.1.gz 2>/dev/null
7  else
8      echo command not found
9  fi
```

```
[dastarikov@fedora lab12]$ ./manlike.sh les
command not found
```

# Задание 1



```
LESS(1)                                     General Commands Manual                                     LESS(1)

NAME
    less - opposite of more

SYNOPSIS
    less -?
    less --help
    less -V
    less --version
    less [-[+]aAbCcDeEfFgGiIjKlMmNnQqRrRsSuUVwWx~]
        [-b space] [-h lines] [-j line] [-k keyfile]
        [-{oO} logfile] [-p pattern] [-P prompt] [-t tag]
        [-T tagfile] [-x tab,...] [-y lines] [-[z] lines]
        [-# shift] [+[#]cmd] [--] [filename]...
    (See the OPTIONS section for alternate option syntax with long option names.)

DESCRIPTION
    less is a program similar to more(1), but which allows backward movement in the file as well as forward movement. Also, less does not have to read the entire input file before starting, so with large input files it starts up faster than text editors like vi(1). less uses termcap (or terminfo on some systems), so it can run on a variety of terminals. There is even limited support for hardcopy termi-
    /usr/share/man/man1/less.1.gz
```

## Задание 2

Задание: Используя встроенную переменную `$RANDOM`, напишите командный файл, генерирующий случайную последовательность букв латинского алфавита. Учтите, что `1$RANDOM` выдаёт псевдослучайные числа в диапазоне от 0 до 32767.

```
1  #!/bin/bash
2  let NUMBER=$RANDOM
3  while [ ! $NUMBER -eq 0 ]
4  do
5      let VALUE=33+$NUMBER%94
6      echo -ne "\x$(printf %02x $VALUE)"
7      let NUMBER=$NUMBER/2
8  done
9  echo
```

```
[dastarikov@fedora lab12]$ ./randchar.sh
HcB`@0(S:-' $"
[dastarikov@fedora lab12]$ ./randchar.sh
CapHcqIdB`@0($"
[dastarikov@fedora lab12]$ ./randchar.sh
B1)T:\>^?_@0($"
[dastarikov@fedora lab12]$ ./randchar.sh
HcB1)%#"P8,&#"
[dastarikov@fedora lab12]$ ./randchar.sh
vK6+U;]?0W<.' $"
```



## Выводы

---

В рамках лабораторной работы изучены основы программирования в оболочке ОС UNIX. Получены навыки написания более сложных командных файлов с использованием логических управляющих конструкций и циклов.