

Отчет по лабораторной работе №12.

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

Данила Андреевич Стариков

Содержание

1	Цель работы	3
2	Выполнение лабораторной работы	4
2.1	Задание 1	4
2.2	Задание 2	6
3	Выводы	7

1 Цель работы

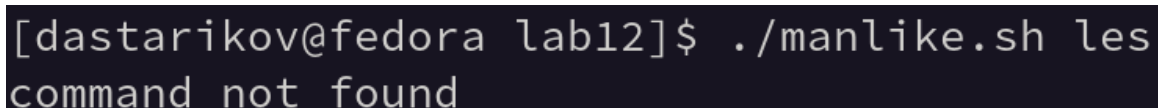
Изучить основы программирования в оболочке ОС UNIX. Научиться писать более сложные командные файлы с использованием логических управляющих конструкций и циклов.

2 Выполнение лабораторной работы

2.1 Задание 1

Задание: 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`.

Создан командный файл `manlike.sh` (Листинг 2.1). Примеры выполнения скрипта представлены на Рисунках 2.1 и 2.2 .



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

Рис. 2.1: Пример №1 работы программы `manlike.sh`.

Листинг 2.1 Текст командного файла Задания №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
10 OPTIONS+="-e $pval $ival"
11 if [ ! $Cflag ]
12 then OPTIONS+=" -i"
13 fi
14 if [ $nflag ]
15 then OPTIONS+=" -n"
16 fi
17 grep ${OPTIONS} > $oval
```

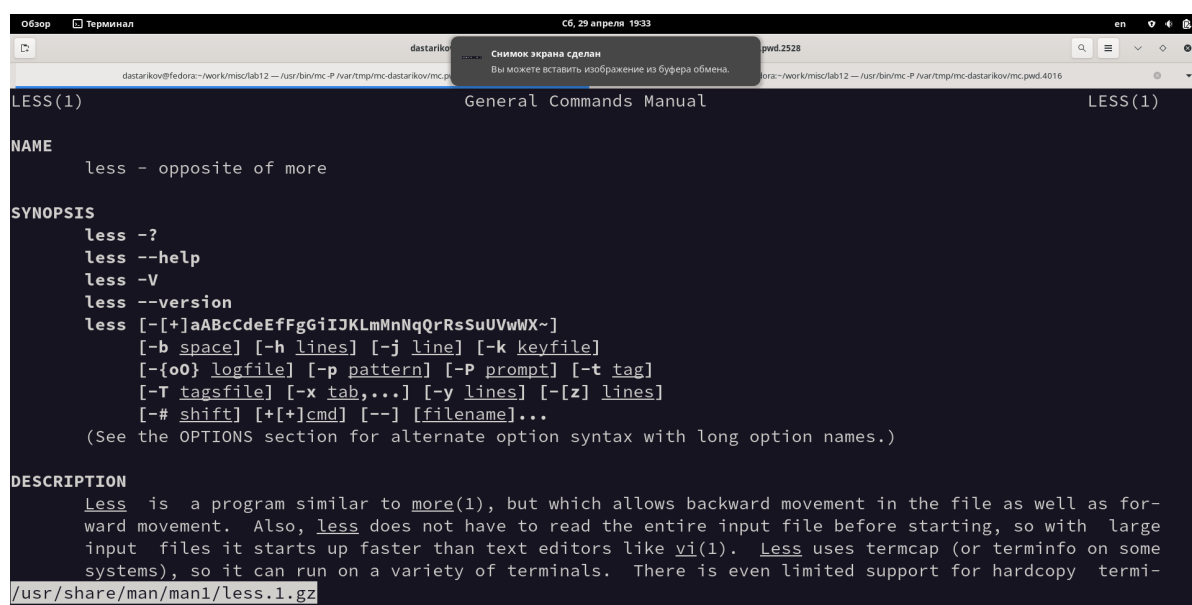


Рис. 2.2: Пример №2 работы программы manlike.sh.

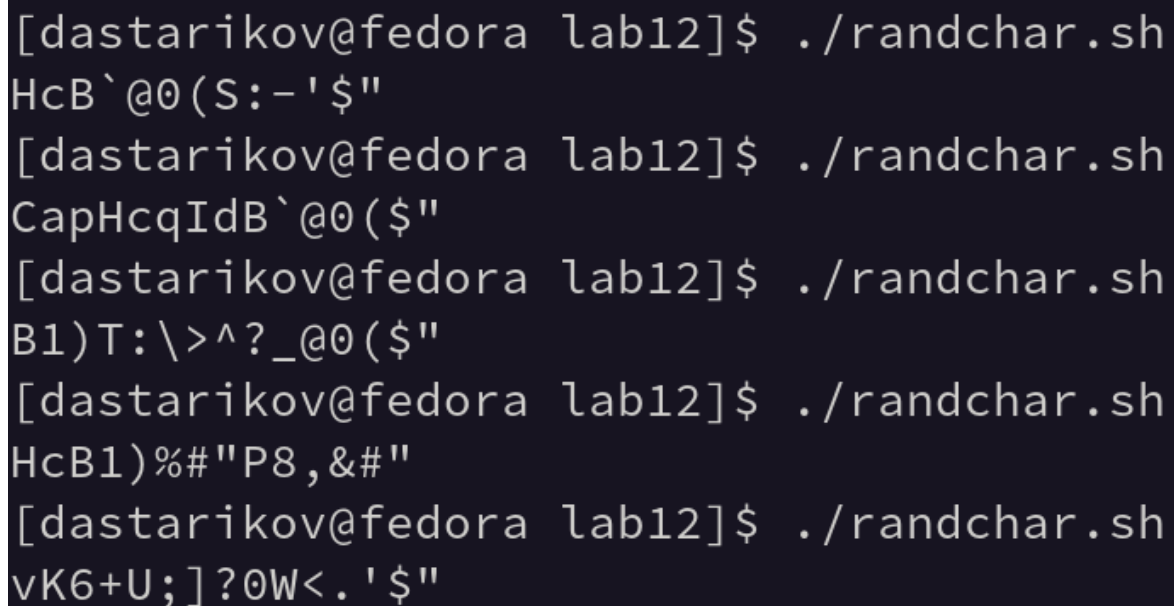
2.2 Задание 2

Задание: 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.

Создан командный файл randchar.sh (Листинг 2.2). Примеры выполнения скрипта представлены на Рисунке 2.3.

Листинг 2.2 Текст командного файла Задания №2.

```
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<.' $"
```

Рис. 2.3: Пример работы программы randchar.sh.

3 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.