

Лабораторная работа №12

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

Рытов Алексей Константинович НФИбд-02-21

Список иллюстраций

2.1	Создание файлов для дальнейшей работы	4
2.2	Скрипт 2.sh	4
2.3	Запуск скрипта	4
2.4	Результат работы скрипта	5
2.5	Скрипт 3.sh	5
2.6	Результаты работы скрипта	6

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

1. Реализовали команду man с помощью командного файла.

```
[akrihtov@fedora Lab12]$ touch 1.sh 2.sh 3.sh  
[akrihtov@fedora Lab12]$ chmod +x 1.sh 2.sh 3.sh  
[akrihtov@fedora Lab12]$
```

Рис. 2.1: Создание файлов для дальнейшей работы

A screenshot of a text editor window titled '2.sh' with a path '~/.bin/Lab12'. The editor shows a shell script with 11 lines of code. The code is as follows:

```
1 while getopts a: opt  
2 do  
3 case $opt in  
4 a) command_name=$OPTARG;;  
5 esac  
6 done  
7  
8 if test -f "/usr/share/man/man1/$command_name.1.gz"  
9 then less /usr/share/man/man1/$command_name.1.gz  
10 else echo "no such command"  
11 fi
```

Рис. 2.2: Скрипт 2.sh

```
[akrihtov@fedora Lab12]$ ./2.sh -a ls
```

Рис. 2.3: Запуск скрипта



Рис. 2.4: Результат работы скрипта

2. Используя встроенную переменную \$RANDOM, написали командный файл, генерирующий случайную последовательность букв латинского алфавита.

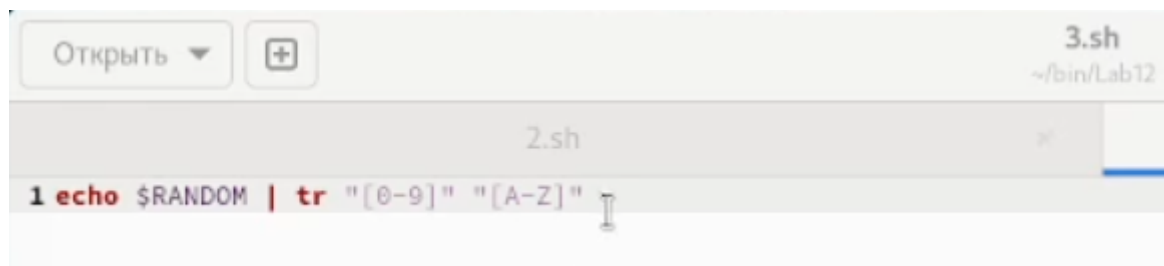


Рис. 2.5: Скрипт 3.sh

```
[akrihtov@fedora Lab12]$ ./3.sh
BCHBE
[akrihtov@fedora Lab12]$ ./3.sh
IADI
[akrihtov@fedora Lab12]$ ./3.sh
HDEE
[akrihtov@fedora Lab12]$ ./3.sh
DBDGI
[akrihtov@fedora Lab12]$
```

Рис. 2.6: Результаты работы скрипта

3 Выводы

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

4 Ответы на контрольные вопросы:

1. Найдите синтаксическую ошибку в следующей строке:

```
while [$1 != "exit"]
```

отсутствуют пробелы рядом с квадратными скобками

2. Как объединить (конкатенация) несколько строк в одну?

```
VAR1="Hello,"
```

```
VAR2=" World"
```

```
VAR3="VAR1VAR2"
```

```
echo "$VAR3"
```

3. Найдите информацию об утилите seq. Какими иными способами можно реализовать её функционал при программировании на bash?

4. Какой результат даст вычисление выражения $\$((10/3))$?

3,(3)

5. Укажите кратко основные отличия командной оболочки zsh от bash.

Zsh более интерактивный и настраиваемый, чем Bash.

У Zsh есть поддержка с плавающей точкой, которой нет у Bash.

В Zsh поддерживаются структуры хеш-данных, которых нет в Bash.

Функции вызова в Bash лучше по сравнению с Zsh.

6. Проверьте, верен ли синтаксис данной конструкции `for ((a=1; a <= LIMIT; a++))`

верен

7. Сравните язык `bash` с какими-либо языками программирования. Какие преимущества у `bash` по сравнению с ними? Какие недостатки?

Shell-сценарий — исходный файл, специально созданный для интерпретатора командной строки типа `Bash`. Программисты обычно пишут Shell-сценарии для повышения производительности за счет автоматизации повторяющихся задач, таких как:

обработка файлов; установка среды; запуск тестовых стеков и развертывания. Кроме того, Shell-сценарии используются внутри виртуальных машин или CI/CD сервисов, обеспечивая чистые и настраиваемые тестовые прогоны или развертывания.