

Отчёт по лабораторной работе №10

**Программирование в командной процессоре ОС UNIX. Командные
файлы**

Кучеренко София

Содержание

0.1	Цель работы	4
1	Выполнение лабораторной работы	5
1.1	Ответы на контрольные вопросы	8
1.2	Выводы	10

Список иллюстраций

1.1	Программа №1	5
1.2	Программа №2	6
1.3	Программа №3	7
1.4	Программа №4	8

Список таблиц

0.1 Цель работы

Цель работы — изучить основы программирования в командной оболочке OS Unix.

1 Выполнение лабораторной работы

1. Написать скрипт, который при запуске будет делать резервную копию самого себя (то есть файла, в котором содержится его исходный код) в другую директорию backup в вашем домашнем каталоге. При этом файл должен архивироваться одним из архиваторов на выбор zip, bzip2 или tar. Способ использования команд архивации необходимо узнать, изучив справку.

```
#!/bin/bash
zayac=$0
outDir=~/.backup/
outFile=${outDir}${zayac:2}.tar
mkdir -p $outDir
tar -cf $outFile $zayac
echo Created backup file $outFile
```

Рис. 1.1: Программа №1

2. Написать пример командного файла, обрабатывающего любое произвольное число аргументов командной строки, в том числе превышающее десять. Например, скрипт может последовательно распечатывать значения всех переданных аргументов.

```
#!/bin/bash
let N=0

function PrintArgument () { # two arguments have to be passed
    echo $1. $2
}

echo This program prints out all the arguments you have just passed:

PrintArgument $N $0
let N++

for parameter in $*
do
    PrintArgument $N $parameter
    let N++
done
```

Рис. 1.2: Программа №2

3. Написать командный файл — аналог команды `ls` (без использования самой этой команды и команды `dir`). Требуется, чтобы он выдавал информацию о нужном каталоге и выводил информацию о возможностях доступа к файлам этого каталога.

```
#!/bin/bash
dir=$1
if [ ! $dir ]
then
    # If dir specified is empty
    dir=./
fi
cd $dir

for file in $(echo *)
do
    if [[ -r $file ]]
    then echo -n r
    else echo -n -
    fi

    if [[ -w $file ]]
    then echo -n w
    else echo -n -
    fi

    echo ' '$file
done
```

Рис. 1.3: Программа №3

4. Написать командный файл, который получает в качестве аргумента командной строки формат файла (.txt, .doc, .jpg, .pdf и т.д.) и вычисляет количество таких файлов в указанной директории. Путь к директории также передаётся в виде аргумента командной строки.

```
#!/bin/bash

# First parameter to be passed is an extension and
# second is a path like this
# ./whatever.sh .tex ~/Documents

N=$#
if (($N != 2))
then
    echo There must be 2 parameters
    exit
fi

lines=0
ext=$1
dir=$2
echo -n There are\
find $dir -maxdepth 1 -type f -name "$*$ext" | wc -l | tr -d "\n"
echo \ $ext files in $dir
```

Рис. 1.4: Программа №4

1.1 Ответы на контрольные вопросы

1. Объясните понятие командной оболочки. Приведите примеры командных оболочек. Чем они отличаются?

Ответ: командная оболочка позволяет исполнять команды.

2. Что такое POSIX?

Ответ: POSIX — набор стандартов описания интерфейсов взаимодействия операционной системы и прикладных программ.

3. Как определяются переменные и массивы в языке программирования bash?

Ответ: через равно.

4. Каково назначение операторов let и read?

Ответ: let позволяет выполнять арифметические операции при задании переменных, read считывает стандартный поток вывода.

5. Какие арифметические операции можно применять в языке программирования bash?

Ответ: стандартные.

6. Что означает операция (())?

Ответ: (()) вычисляют логические условные выражения.

7. Какие стандартные имена переменных Вам известны?

Ответ: PATH, ENV, TERM.

8. Что такое метасимволы?

Ответ: специальные символы.

9. Как экранировать метасимволы?

Ответ: как угодно, но можно через .

10. Как создавать и запускать командные файлы?

Ответ: для создания файла применить команду `touch <file> && chmod +x <file>`. Для запуска ввести `./<file>`

11. Как определяются функции в языке программирования bash?

Ответ: при помощи ключевого слова `function`.

12. Каким образом можно выяснить, является файл каталогом или обычным файлом?

Ответ: `ls -l` выведет дополнительную информацию.

13. Каково назначение команд `set`, `typeset` и `unset`?

Ответ: таково: изменение значения внутренних переменных сценария, наложение ограничений на переменные, удаление переменной.

14. Как передаются параметры в командные файлы?

Ответ: через пробел при запуске программы.

15. Назовите специальные переменные языка bash и их назначение.

Ответ: см. вопрос 7.

1.2 Выводы

В ходе выполнения лабораторной работы были изучены основы программирования в командной оболочке OS Unix. Цель работы была достигнута.