

Министерство образования Республики Беларусь

**Учреждение образования
«Белорусский государственный университет
информатики и радиоэлектроники»**

ФАКУЛЬТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ И УПРАВЛЕНИЯ

Кафедра интеллектуальных информационных технологий

Отчет

По дисциплине: Операционные системы
Лабораторная работа №1

Выполнили:
Липский Р. В.,
Жолнерчик И. А.
гр. 121701

Проверил: Цирук В. А.

Минск 2022

Цель: ознакомиться с основными командами ОС Linux, научиться писать скрипты и запускать с командной строки, запускать компиляцию кода на C++.

Индивидуальное задание: Написать скрипт, подсчитывающий суммарный размер файлов в заданном каталоге и всех его подкаталогах (имя каталога задаётся пользователем в качестве первого аргумента командной строки). Скрипт выводит результаты подсчета в файл (второй аргумент командной строки) в виде "каталог (полный путь), суммарный размер файлов, число просмотренных файлов".

Ход выполнения:

1. Изучить теоретическую часть лабораторной работы.

1.1. *Что такое Bash?*

Bash — это командная оболочка для UNIX-подобных операционных систем (UNIX, GNU/Linux, MacOS). Она дает пользователю систему команд для работы с файлами и папками, поиском, настройкой окружения и позволяет управлять ОС прямо из командной строки. Слово bash читается как «баш» и расшифровывается как Bourne-Again Shell.

1.2. *Какой командой можно посмотреть весь файл, не открывая его?*

cat <имя_файла> - просмотр содержимого файла

1.3. Если файл большой, как в командной строке сделать так, чтобы он выводился по частям?

nl <имя_файла> - как cat, но выводит содержимое с номерами строк

less <имя_файла> - как cat, но отображает содержимое только в рамках текущего окна в режиме просмотра

more <имя_файла> - как less, но вместо того, чтобы ждать выхода из режим просмотра нажатием q, сразу же возвращает управление в терминал

head <имя_файла> - как cat, но выводит на экран только первые 10 строк

tail <имя_файла> - как cat, но выводит на экран только последние 10 строк

less <имя_файла> - так как отображает содержимое только в рамках текущего окна в режиме просмотра

1.4. *Напишите конструкцию условного оператора if в bash.*

if <условие> then

 <действие>

else

 <действие в случае невыполнения условия>

fi

test - используется для логического сравнения. после выражения необходима закрывающая скобка "]"

[- синоним команды test

[[- расширенная версия "[" (начиная с версии 2.02), внутри которой могут быть использованы || (или), & (и). Должна иметь закрывающую скобку "]"

(()) - математическое сравнение

1.5. Напишите конструкцию цикла for.

```
for <переменная> in <строка с разделителем> do
    echo The $var item
done
```

2. В любом текстовом редакторе написать программу 1.с, выводящую на экран фразу "HELLO Ubuntu".

Описание: сложностей с написанием данной программы не возникло, всё, что требовалось для выполнения этого этапа, мы изучили на прошлых курсах.

```
#include <stdio.h>
```

```
int main(void) {
    printf("HELLO Ubuntu\n");
}
```

3. Написать скрипт, выводящий на консоль и в файл все аргументы командной строки.

Описание: команда `tee` одновременно выводит текст и в stdout и в указанный файл.

```
#!/bin/bash
echo "$@" | tee $1
```

4. Написать скрипт, выводящий в файл (имя файла задаётся пользователем в качестве первого аргумента командной строки) имена всех файлов с заданным расширением (третий аргумент командной строки) из заданного каталога (имя каталога задаётся пользователем в качестве второго аргумента командной строки).

Описание: сложностей с выполнением не возникло, подобные задания мы выполняли на прошлом курсе.

```
#!/bin/bash
ls $2/*.$3 | tee $1
```

5. Написать скрипт, компилирующий и запускающий программу (имя исходного файла и exe- файла результата задаётся пользователем в качестве аргументов командной строки). В случае ошибок при компиляции вывести на консоль сообщение об ошибках и не запускать программу на выполнение.

Описание: здесь мы использовали условную конструкцию, если exit code компилятора = 0 (успех), программа выполняется, иначе выводится сообщение об ошибке.

```
#!/bin/bash
if gcc $1 -o $2; then
    echo "Compilation succesful"
    ./$2
else
    echo "Compilation error"
fi
```

6. Индивидуальное задание: Написать скрипт, подсчитывающий суммарный размер файлов в заданном каталоге и всех его подкаталогах (имя каталога задаётся пользователем в качестве первого аргумента командной строки). Скрипт выводит результаты подсчета в файл (второй аргумент командной строки) в виде "каталог (полный путь), суммарный размер файлов, число просмотренных файлов".

Описание: для выполнения этого задания пришлось использовать много командных утилит, realpath необходим для получения абсолютного пути из относительного, ls -lR рекурсивно обходит все директории и выводит подробную информацию о файлах, awk — мощная утилита для обработки текста, в данном примере она используется для сложения веса всех файлов, bc используется для перевода веса из байт в мегабайты, wc -l — подсчитывает количество строк в тексте, соответственно количество обработанных файлов.

```
#!/bin/bash
abs=$(realpath $1)
files=$(ls -lR $1 | grep '^-')
size=$(echo "$files" | awk '{size += $5} END { print size }')
sizeMB=$(echo "scale=2 ; $size / 1000000" | bc)
count=$(echo "$files" | wc -l)
echo "abs path: ${abs}, total size: ${sizeMB}MB, total count: ${count}"
```