

Министерство образования Республики Беларусь  
Учреждение образования  
БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
ИНФОРМАТИКИ И РАДИОТЕХНИКИ

Факультет компьютерного проектирования  
Кафедра проектирования информационно-компьютерных систем

**ОТЧЁТ**  
к лабораторной работе №1 на тему  
**«УПРАВЛЕНИЕ ОС LINUX, ИНТЕРПРЕТАТОР BASH»**  
по дисциплине «ОСМУ»

Выполнил студент:

К.Г. Хоменок

Проверил:

А.Д. Станкевич

Минск 2024

**Цель работы:** изучить основные объекты, команды, типы данных и операторы управления интерпретатора BASH; создать скрипт-файл.

## 1 Краткие теоретические сведения

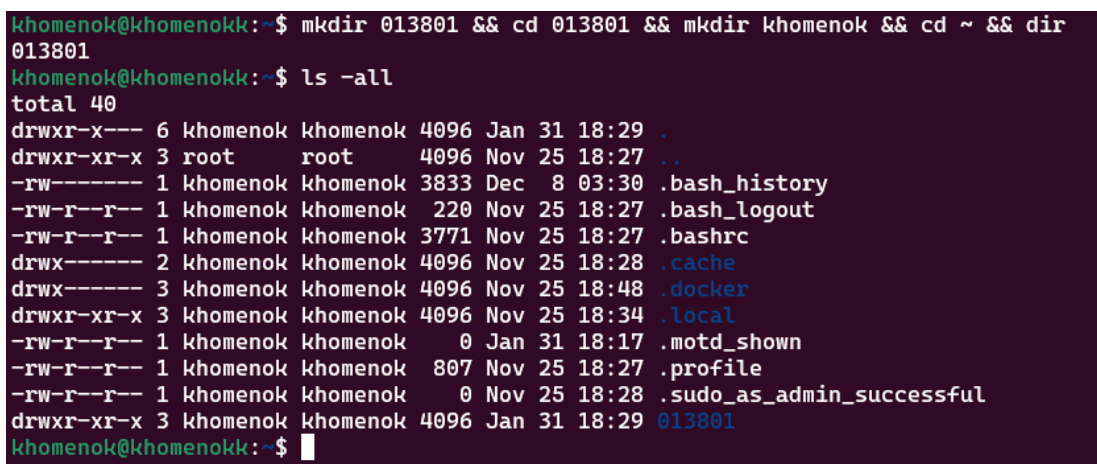
**Bash** — это sh-совместимый интерпретатор командного языка, выполняющий команды, прочитанные со стандартного входного потока или из файла. Скрипт-файл — это обычный текстовый файл, содержащий последовательность команд `bash`, для которого установлены права на выполнение.

## 2 Порядок выполнения работы

1 Изучить теоретическую часть лабораторной работы.

2 В консольном режиме создать, используя команды из табл.1, в домашней папке подкаталог: `/номер_группы/ФИО_студента`, где в дальнейшем будут храниться все файлы студента. Перейти в корневой каталог и вывести его содержимое используя команды `dir` и `ls -all`, проанализировать различия.

Выполним эти требования, рисунок 1.



```
khomenok@khomenokk:~$ mkdir 013801 && cd 013801 && mkdir khomenok && cd ~ && dir
013801
khomenok@khomenokk:~$ ls -all
total 40
drwxr-x--- 6 khomenok khomenok 4096 Jan 31 18:29 .
drwxr-xr-x 3 root      root      4096 Nov 25 18:27 ..
-rw----- 1 khomenok khomenok 3833 Dec  8 03:30 .bash_history
-rw-r--r-- 1 khomenok khomenok  220 Nov 25 18:27 .bash_logout
-rw-r--r-- 1 khomenok khomenok 3771 Nov 25 18:27 .bashrc
drwx----- 2 khomenok khomenok 4096 Nov 25 18:28 .cache
drwx----- 3 khomenok khomenok 4096 Nov 25 18:48 .docker
drwxr-xr-x 3 khomenok khomenok 4096 Nov 25 18:34 .local
-rw-r--r-- 1 khomenok khomenok    0 Jan 31 18:17 .motd_shown
-rw-r--r-- 1 khomenok khomenok  807 Nov 25 18:27 .profile
-rw-r--r-- 1 khomenok khomenok    0 Nov 25 18:28 .sudo_as_admin_successful
drwxr-xr-x 3 khomenok khomenok 4096 Jan 31 18:29 013801
khomenok@khomenokk:~$
```

Рисунок 1 – Выполненный пункт №2

После перехода в корневой каталог и использования команды `dir` и `ls -all`, происходит следующее: `dir` — это команда, используемая для вывода содержимого каталога, тогда как `ls -all` — более расширенная команда, выводящая подробную информацию о файлах и каталогах, включая скрытые. Различия между ними заключаются в том, что `dir` не является стандартной командой в Linux, в то время как `ls -all` является более универсальным

инструментом, доступным во всех основных дистрибутивах Linux, и предоставляет более полезную информацию для анализа содержимого каталогов.

3 Проверить действие команд `ps`, `ps -x`, `top`, `htop`. Найти в справочной системе используя команду `man` справку по функциям `fprintf`, `fputc` и команде `ls` (рисунок 2).

```
khomenok@khomenokk:~$ ps
  PID TTY          TIME CMD
  550 pts/0        00:00:00 bash
 1160 pts/0        00:00:00 ps
khomenok@khomenokk:~$ ps -x
  PID TTY          STAT TIME COMMAND
  550 pts/0        Ss   0:00 -bash
  609 ?            Ss   0:00 /lib/systemd/systemd --user
  610 ?            S    0:00 (sd-pam)
  616 pts/1        S+   0:00 -bash
 1161 pts/0        R+   0:00 ps -x
```

Рисунок 2 – Выполнение пункта №3

При выполнении команд `ps`, `ps -x`, `top`, и `htop` были изучены действия различных утилит мониторинга процессов в Linux.

Однако, не удалось найти `man`-страницы для функций `fprintf` и `fputc`. Это связано с тем, что данные функции принадлежат стандартной библиотеке языка программирования C, а не непосредственно операционной системе. Для нахождения `man`-страницы функций `fprintf` и `fputc` необходимо установить компилятор `gcc` уже на этом этапе.

Также были выполнены команды `top` и `htop`, которые позволяют мониторить нагрузку на систему в режиме реального времени.

4 В текстовом редакторе `joe` (вызов: `joe 1.c`) написать программу `1.c`, выводящую на экран фразу “HELLO SUSE Linux”. Компилировать полученную программу компилятором `gcc`: `gcc 1.c -o 1.exe`. Запустить полученный файл `1.exe` на выполнение: `./1.EXE`

Для этого выполним:

```
sudo apt install joe -y
joe 1.c
```

Содержимое файла `1.c`:

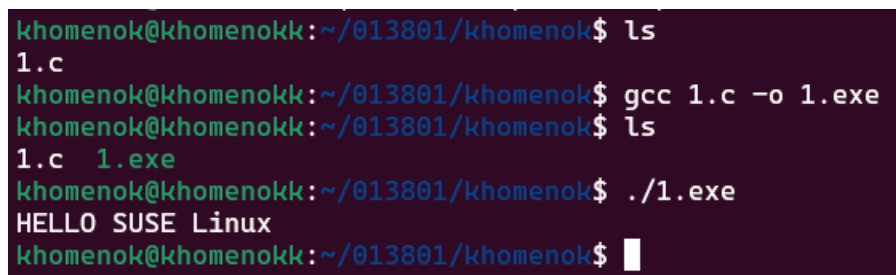
```
#include <stdio.h>
```

```
int main() {
    printf("HELLO SUSE Linux\n");
    return 0;
}
```

Чтобы сохранить файл 1.c, необходимо нажать Ctrl + K, затем X, чтобы сохранить и выйти из файла, либо D, чтобы просто сохранить. Далее необходимо установить компилятор gcc:

```
sudo apt update && sudo apt upgrade -y
sudo apt install gcc -y
```

Далее необходимо выполнить действия, показанные на рисунке 3.



```
khomenok@khomenokk:~/013801/khomenok$ ls
1.c
khomenok@khomenokk:~/013801/khomenok$ gcc 1.c -o 1.exe
khomenok@khomenokk:~/013801/khomenok$ ls
1.c 1.exe
khomenok@khomenokk:~/013801/khomenok$ ./1.exe
HELLO SUSE Linux
khomenok@khomenokk:~/013801/khomenok$
```

Рисунок 3 – Выполнение пункта №4

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

```
#!/bin/bash

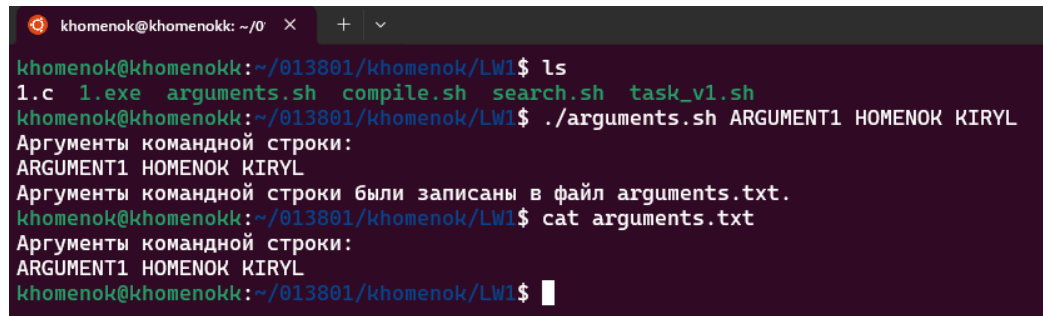
# Проверяем, переданы ли аргументы командной строки
if [ $# -eq 0 ]; then
    echo "Нет аргументов командной строки для обработки."
    exit 1
fi

# Выводим аргументы на консоль
echo "Аргументы командной строки:"
echo "$@"

# Записываем аргументы в файл arguments.txt
echo "Аргументы командной строки:" > arguments.txt
echo "$@" >> arguments.txt

echo "Аргументы командной строки были записаны в файл arguments.txt."
```

Проверим работоспособность скрипта, которую можно увидеть на рисунке 4.



```
khomenok@khomenokk: ~/0 X + v
khomenok@khomenokk:~/013801/khomenok/LW1$ ls
1.c 1.exe arguments.sh compile.sh search.sh task_v1.sh
khomenok@khomenokk:~/013801/khomenok/LW1$ ./arguments.sh ARGUMENT1 HOMENOK KIRYL
Аргументы командной строки:
ARGUMENT1 HOMENOK KIRYL
Аргументы командной строки были записаны в файл arguments.txt.
khomenok@khomenokk:~/013801/khomenok/LW1$ cat arguments.txt
Аргументы командной строки:
ARGUMENT1 HOMENOK KIRYL
khomenok@khomenokk:~/013801/khomenok/LW1$
```

Рисунок 4 – Проверка работоспособности скрипта №1

6 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).

```
#!/bin/bash

# Проверяем количество аргументов командной строки
if [ $# -ne 3 ]; then
    echo "Использование: $0 <имя_файла> <каталог> <расширение>"
    exit 1
fi

# Имя файла, каталога и расширение
output_file="$1"
directory="$2"
extension="$3"

# Проверяем, существует ли указанный каталог
if [ ! -d "$directory" ]; then
    echo "Указанный каталог не существует."
    exit 1
fi

# Переходим в указанный каталог
cd "$directory" || exit

# Проверяем, есть ли файлы с указанным расширением в каталоге
files=$(find . -maxdepth 1 -type f -name ".*$extension")

# Проверяем, найдены ли файлы
```

```

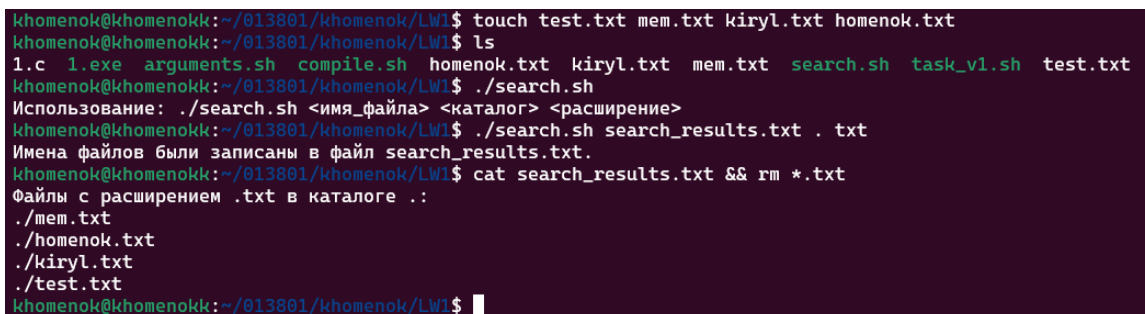
if [ -z "$files" ]; then
    echo "Файлов с расширением .$extension не найдено в каталоге
$directory."
    exit 0
fi

# Записываем имена файлов в указанный файл
echo "Файлы с расширением .$extension в каталоге $directory:" >
"$output_file"
echo "$files" >> "$output_file"

echo "Имена файлов были записаны в файл $output_file."

```

Проверим работоспособность скрипта, которую можно увидеть на рисунке 5.



```

khomenok@khomenokk:~/013801/khomenok/LW1$ touch test.txt mem.txt kiryl.txt homenok.txt
khomenok@khomenokk:~/013801/khomenok/LW1$ ls
l.c l.exe arguments.sh compile.sh homenok.txt kiryl.txt mem.txt search.sh task_v1.sh test.txt
khomenok@khomenokk:~/013801/khomenok/LW1$ ./search.sh
Использование: ./search.sh <имя_файла> <каталог> <расширение>
khomenok@khomenokk:~/013801/khomenok/LW1$ ./search.sh search_results.txt . txt
Имена файлов были записаны в файл search_results.txt.
khomenok@khomenokk:~/013801/khomenok/LW1$ cat search_results.txt && rm *.txt
Файлы с расширением .txt в каталоге .:
./mem.txt
./homenok.txt
./kiryl.txt
./test.txt
khomenok@khomenokk:~/013801/khomenok/LW1$

```

Рисунок 5 – Проверка работоспособности скрипта №2

7 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.

```

#!/bin/bash

# Проверяем, что пользователь предоставил два аргумента
if [ "$#" -ne 2 ]; then
    echo "Использование: $0 <source_c_file> <имя_output_файла>"
    exit 1
fi

# Извлекаем имена файлов из аргументов командной строки
source_file=$1
exe_file=$2

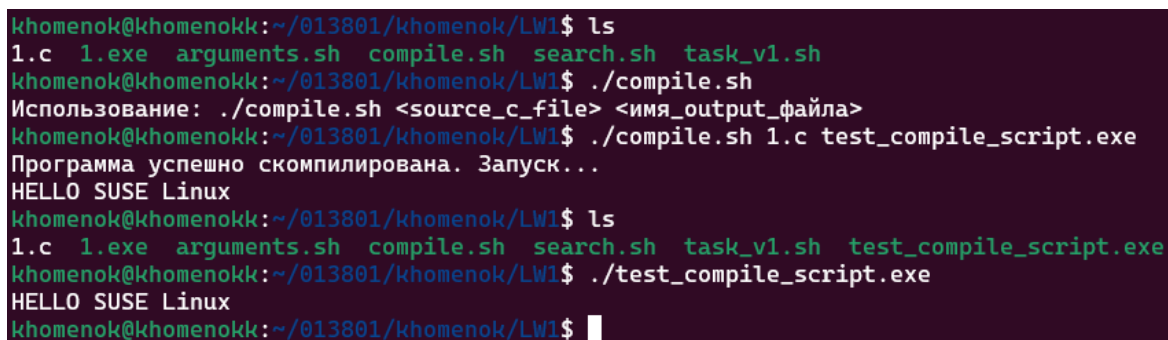
# Компилируем программу

```

```
gcc -o "$exe_file" "$source_file"

# Проверяем успешность компиляции
if [ $? -eq 0 ]; then
    echo "Программа успешно скомпилирована. Запуск..."
    ./"$exe_file"
else
    echo "Ошибка: Программа не скомпилирована."
fi
```

Проверим работоспособность скрипта, которую можно увидеть на рисунке 6.



```
khomenok@khomenokk:~/013801/khomenok/LW1$ ls
1.c 1.exe arguments.sh compile.sh search.sh task_v1.sh
khomenok@khomenokk:~/013801/khomenok/LW1$ ./compile.sh
Использование: ./compile.sh <source_c_file> <имя_output_файла>
khomenok@khomenokk:~/013801/khomenok/LW1$ ./compile.sh 1.c test_compile_script.exe
Программа успешно скомпилирована. Запуск...
HELLO SUSE Linux
khomenok@khomenokk:~/013801/khomenok/LW1$ ls
1.c 1.exe arguments.sh compile.sh search.sh task_v1.sh test_compile_script.exe
khomenok@khomenokk:~/013801/khomenok/LW1$ ./test_compile_script.exe
HELLO SUSE Linux
khomenok@khomenokk:~/013801/khomenok/LW1$ █
```

Рисунок 6 – Проверка работоспособности скрипта №3

Таким образом, общая часть лабораторной работы была выполнена. Далее необходимо приступить к выполнению индивидуального задания.

### 3 Выполнение индивидуального задания:

Написать скрипт для поиска файлов заданного размера в заданном каталоге (имя каталога задаётся пользователем в качестве третьего аргумента командной строки). Диапазон (мин.- макс.) размеров файлов задаётся пользователем в качестве первого и второго аргумента командной строки.

Реализованный скрипт представлен ниже:

```
#!/bin/bash

if [ "$#" -ne 3 ]; then
    echo "Использование: $0 <мин. размер> <макс. размер>
<каталог>"
    exit 1
fi
```

```

if ! [[ "$1" =~ ^[0-9]+$ ]]; then
    echo "Ошибка: Первый аргумент должен быть числом"
    exit 1
fi

if ! [[ "$2" =~ ^[0-9]+$ ]]; then
    echo "Ошибка: Второй аргумент должен быть числом"
    exit 1
fi

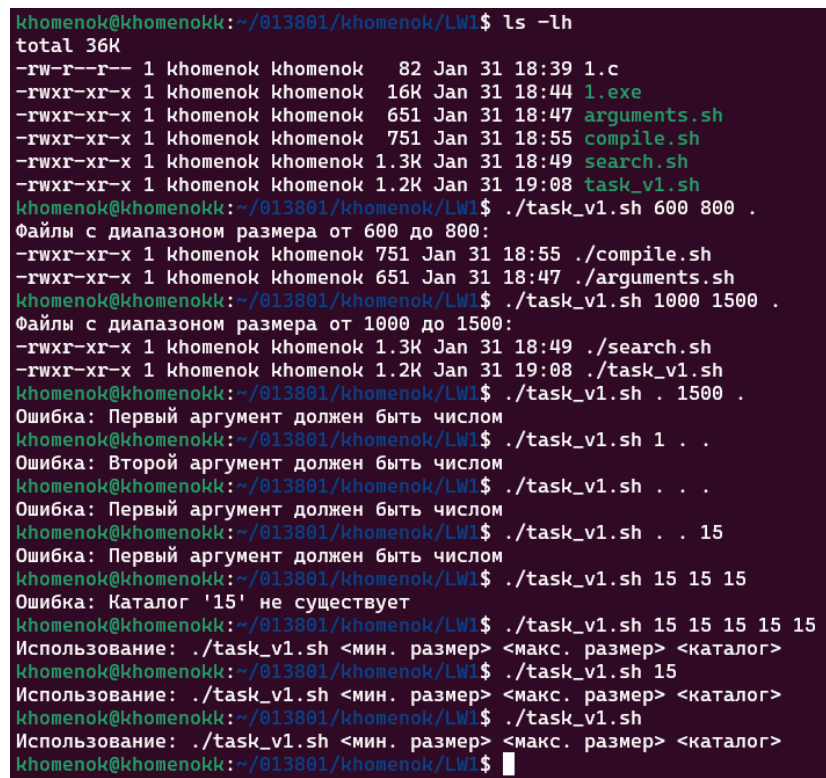
if [ ! -d "$3" ]; then
    echo "Ошибка: Каталог '$3' не существует"
    exit 1
fi

min_size=$1
max_size=$2
directory=$3

echo "Файлы с диапазоном размера от $min_size до $max_size:"
find "$directory" -type f -size +"$min_size"c -size -"$max_size"c
-exec ls -lh {} \; 2>/dev/null

```

Демонстрация работы скрипта представлена на рисунке 7.



```

khomenok@khomenokk:~/013801/khomenok/LW1$ ls -lh
total 36K
-rw-r--r-- 1 khomenok khomenok 82 Jan 31 18:39 1.c
-rwxr-xr-x 1 khomenok khomenok 16K Jan 31 18:44 1.exe
-rwxr-xr-x 1 khomenok khomenok 651 Jan 31 18:47 arguments.sh
-rwxr-xr-x 1 khomenok khomenok 751 Jan 31 18:55 compile.sh
-rwxr-xr-x 1 khomenok khomenok 1.3K Jan 31 18:49 search.sh
-rwxr-xr-x 1 khomenok khomenok 1.2K Jan 31 19:08 task_v1.sh
khomenok@khomenokk:~/013801/khomenok/LW1$ ./task_v1.sh 600 800 .
Файлы с диапазоном размера от 600 до 800:
-rwxr-xr-x 1 khomenok khomenok 751 Jan 31 18:55 ./compile.sh
-rwxr-xr-x 1 khomenok khomenok 651 Jan 31 18:47 ./arguments.sh
khomenok@khomenokk:~/013801/khomenok/LW1$ ./task_v1.sh 1000 1500 .
Файлы с диапазоном размера от 1000 до 1500:
-rwxr-xr-x 1 khomenok khomenok 1.3K Jan 31 18:49 ./search.sh
-rwxr-xr-x 1 khomenok khomenok 1.2K Jan 31 19:08 ./task_v1.sh
khomenok@khomenokk:~/013801/khomenok/LW1$ ./task_v1.sh . 1500 .
Ошибка: Первый аргумент должен быть числом
khomenok@khomenokk:~/013801/khomenok/LW1$ ./task_v1.sh 1 . .
Ошибка: Второй аргумент должен быть числом
khomenok@khomenokk:~/013801/khomenok/LW1$ ./task_v1.sh . . .
Ошибка: Первый аргумент должен быть числом
khomenok@khomenokk:~/013801/khomenok/LW1$ ./task_v1.sh . . 15
Ошибка: Первый аргумент должен быть числом
khomenok@khomenokk:~/013801/khomenok/LW1$ ./task_v1.sh 15 15 15
Ошибка: Каталог '15' не существует
khomenok@khomenokk:~/013801/khomenok/LW1$ ./task_v1.sh 15 15 15 15 15
Использование: ./task_v1.sh <мин. размер> <макс. размер> <каталог>
khomenok@khomenokk:~/013801/khomenok/LW1$ ./task_v1.sh 15
Использование: ./task_v1.sh <мин. размер> <макс. размер> <каталог>
khomenok@khomenokk:~/013801/khomenok/LW1$ ./task_v1.sh
Использование: ./task_v1.sh <мин. размер> <макс. размер> <каталог>
khomenok@khomenokk:~/013801/khomenok/LW1$

```

Рисунок 7 – Демонстрация работы написанного скрипта



Таким образом, лабораторная работа была выполнена с общим и индивидуальным заданием.

### **3 Вывод**

В ходе выполнения лабораторной работы по управлению операционной системой Linux и использованию интерпретатора Bash были освоены основные команды и инструменты Linux. Была создана структура каталогов в домашней папке, изучены команды для работы с процессами и поиска информации в справочной системе, затем была написана программа на языке Си и скрипты на Bash для работы с файлами и процессами.

Особое внимание уделено написанию скрипта для поиска файлов заданного размера в указанном каталоге с заданным диапазоном размеров. Полученные навыки и практический опыт работы с командной строкой и скриптами Bash позволят эффективно управлять и администрировать системы Linux в будущем.