



Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования

«Московский государственный технический университет
имени Н.Э. Баумана (национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ
КАФЕДРА

Информатика и системы управления
Информационная безопасность (ИУ8)

Безопасность Систем Баз Данных

Отчет по Лабораторной работе №4 “Получение потоков запускаемого приложения (C++)”

Выполнил:
Овсебян А.Н.,
студент группы ИУ8-63

Проверил:
Зенькович С. А.,
старший преподаватель
кафедры ИУ8

Москва, 2021

Оглавление

Цель работы	3
Ход работы.....	3
Практическая часть	3
Вывод.....	4
Приложение А.	5
Приложение Б.....	6

Цель работы

Реализовать приложение, запускающее произвольное приложение с получением его потоков и кода завершения на C++

Ход работы

Практическая часть

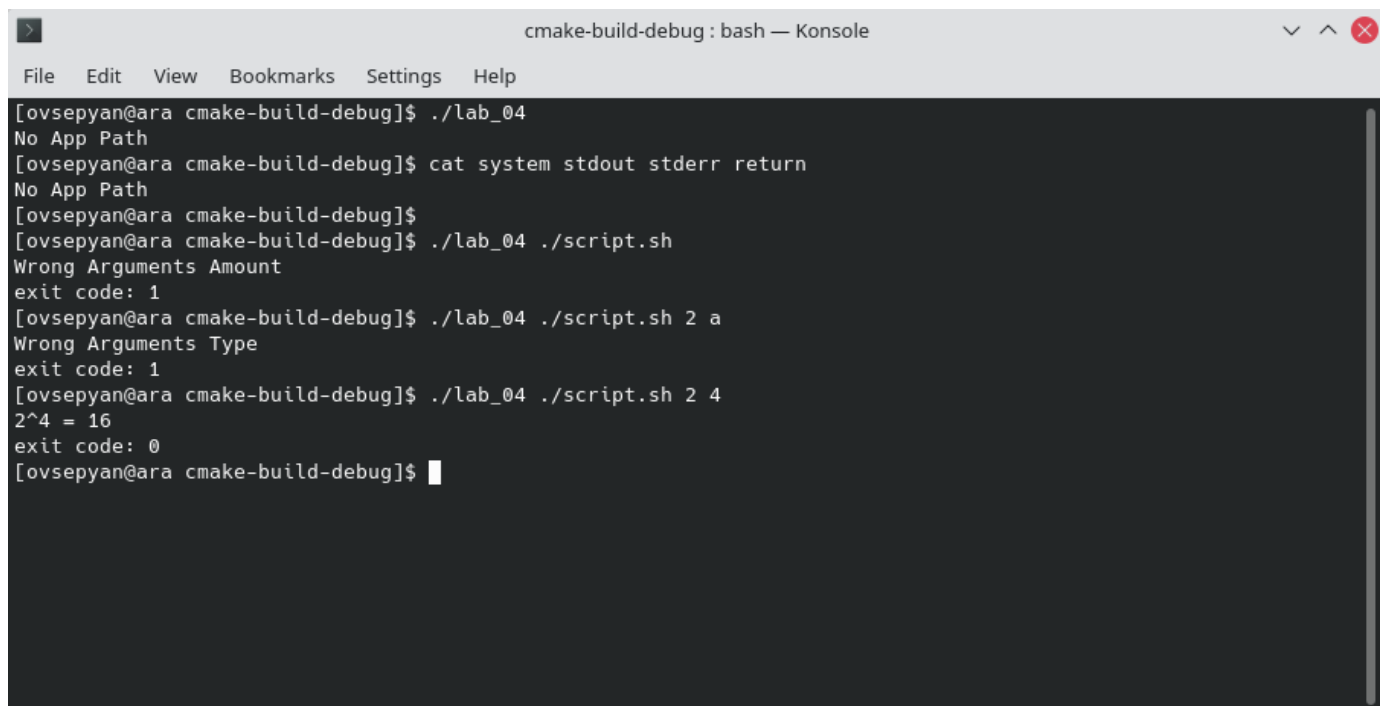
Для выполнения задания мною был реализован shell скрипт. Данный скрипт возводит число `base` в степень `power` (оба числа подаются в качестве аргументов). Скрипт проверяет введенные данные на:

- правильное количество (2)
- правильный тип (целые числа)

Исходный код скрипта `script.sh` в [приложении А](#).

C++ приложение, которое запускает вышеописанный скрипт, получает его стандартные потоки (`stdout`, `stderr`) и код завершения. Выводит полученные данные в соответствующие поток и файл. Ошибки в ходе работы самого приложения выводятся в `stderr` и записываются в отдельный файл.

Исходный код C++ приложения в [приложении Б](#).



```
cmake-build-debug : bash — Konsole
File Edit View Bookmarks Settings Help
[ovsepyan@ara cmake-build-debug]$ ./lab_04
No App Path
[ovsepyan@ara cmake-build-debug]$ cat system stdout stderr return
No App Path
[ovsepyan@ara cmake-build-debug]$
[ovsepyan@ara cmake-build-debug]$ ./lab_04 ./script.sh
Wrong Arguments Amount
exit code: 1
[ovsepyan@ara cmake-build-debug]$ ./lab_04 ./script.sh 2 a
Wrong Arguments Type
exit code: 1
[ovsepyan@ara cmake-build-debug]$ ./lab_04 ./script.sh 2 4
2^4 = 16
exit code: 0
[ovsepyan@ara cmake-build-debug]$
```

Рисунок 1 - пример использования приложения

Вывод

Были получены знания о потоках и реализовано приложение для работы с потоками запускаемого приложения.

Приложение А.

```
#!/bin/bash

if [ $# -eq 2 ]
then
    re='^[0-9]+$'
    if ! [[ $1 =~ $re ]] || ! [[ $2 =~ $re ]]
    then
        echo "Wrong Arguments Type">&2;
        exit 1;
    fi

    res=1
    base=$1
    power=$2
    for ((i=0;i<power;i++))
    {
        res=$((res * base))
    }
    echo "$1^$2 = $res">&1
    exit 0
fi

echo "Wrong Arguments Amount">&2
exit 1;
```

Приложение Б.

```
#include <iostream>
#include <unistd.h>
#include <sys/wait.h>

template<typename... ARGS>
void print(FILE *file, int code, char *format, ARGS... args)
{
    fprintf(file, format, args...);
    if (code)
    {
        fprintf(stdout, format, args...);
    } else
    {
        fprintf(stderr, format, args...);
    }
}

int main(int argc, char *argv[])
{
    // file for logging app errors
    FILE *f_log;
    f_log = fopen("system", "w");

    // files for streams and exit code
    FILE *f_stdout;
    FILE *f_stderr;
    FILE *f_return;
    f_stdout = fopen("stdout", "w");
    f_stderr = fopen("stderr", "w");
    f_return = fopen("return", "w");

    if (argc < 2)
    {
        print(f_log, 1, "No App Path\n");
        return 1;
    }

    // full path to app with args
    std::string string_path = "./";
    for (int i = 1; i < argc; i++)
    {
        string_path += std::string(argv[i]) + " ";
    }
    const char *full_path = string_path.erase(string_path.length() -
1).c_str();

    int stdout_pipe[2], stderr_pipe[2];
    if (pipe(stdout_pipe) < 0 || pipe(stderr_pipe) < 0)
    {
        print(f_log, 1, "Pipe Error\n");
        return 1;
    }

    int exitcode;
    pid_t pid = fork();
    if (pid == 0)
    {
        close(stdout_pipe[0]);
        close(stderr_pipe[0]);
```

```

        dup2(stdout_pipe[1], 1);
        dup2(stderr_pipe[1], 2);
        close(stdout_pipe[1]);
        close(stderr_pipe[1]);

        exitcode = system(full_path);
        return WEXITSTATUS(exitcode);
    } else
    {
        waitpid(pid, &exitcode, 0);
    }
    exitcode = WEXITSTATUS(exitcode);
    close(stdout_pipe[1]);
    close(stderr_pipe[1]);

    // store streams data to buffers
    char stdout_buffer[128], stderr_buffer[128];
    stdout_buffer[read(stdout_pipe[0], stdout_buffer, sizeof(stdout_buffer))]
= '\0';
    stderr_buffer[read(stderr_pipe[0], stderr_buffer, sizeof(stderr_buffer))]
= '\0';

    // print data
    print(f_stdout, 0, stdout_buffer);
    print(f_stderr, 1, stderr_buffer);
    print(f_return, 0, "exit code: %d\n", exitcode);

    return 0;
}

```