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 Информатика и системы управления КАФЕДРА Информационная безопасность (ИУ8)

**Безопасность Систем Баз Данных**

Отчет

по Лабораторной работе №3

**“Получение потоков запускаемого приложения (stderr/stdout)”**

###### Выполнил:

#### Овсепян А.Н.,

#### студент группы ИУ8-63

###### Проверил:

#### Зенькович С. А.,

#### старший преподаватель кафедры ИУ8

#### Москва, 2021

**Оглавление**

[Цель работы 3](#_Toc75704070)

[Ход работы 3](#_Toc75704071)

[Теоретическая часть 3](#_Toc75704072)

[Практическая часть 3](#_Toc75704073)

[Вывод 5](#_Toc75704074)

[Приложение А. 6](#_Toc75704075)

[Приложение Б. 7](#_Toc75704076)

# Цель работы

Необходимо реализовать POSIX-совместимый shell-скрипт для получения потоков запускаемого приложения

# Ход работы

## Теоретическая часть

**POSIX** - (англ. Portable Operating System Interface — переносимый интерфейс операционных систем) — набор стандартов, описывающих интерфейсы между операционной системой и прикладной программой (системный API), библиотеку языка C и набор приложений и их интерфейсов. Стандарт создан для обеспечения совместимости различных UNIX-подобных операционных систем и переносимости прикладных программ на уровне исходного кода, но может быть использован и для не-Unix систем.

## 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)
* правильное тип (целые числа)

Результат операции выводится в поток stdout, ошибка в stderr.

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

В соответствии с заданием необходимо получать потоки скрипта stderr, stdout, код завершения программы, выводить все в консоль и записывать в файлы.

Исходный код скрипта opener.sh в [приложении Б](#_Приложение_Б.).

Для проверки написанного shell скрипта на POSIX-совместимость была скачана утилита shellcheck.

pacman -S shellcheck

shellchek shell.sh app.sh

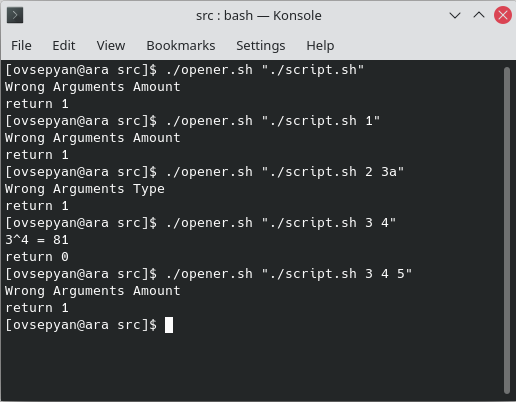


Рисунок 1- Пример использования shell-скрипта

# Вывод

В ходе выполнения данной лабораторной работы было изучено понятие стандартов POSIX, а также написан shell скрипт для получения потоков (stderr/stdout) запускаемого приложения

# Приложение А.

#!/bin/bash

# exponentiation script

**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.

#!/bin/bash

# script opener

**if** [ $# -gt 0 ]; **then**

out=`mktemp **.**/out.XXXX`

err=`mktemp **.**/err.XXXX`

ret=`mktemp **.**/ret.XXXX`

$1 1>$out 2>$err;

**echo** **return** $?>$ret;

cat $out $err $ret;

rm -f $out $err $ret;

**exit** 0;

**fi**

**echo** "Error: Wrong Arguments Amount" >&2;

**exit** 1;