



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 №3 “Получение потоков запускаемого приложения (stderr/stdout)”

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
Теоретическая часть.....	3
Практическая часть	3
Вывод.....	5
Приложение А.	6
Приложение Б.	7

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)
- правильный тип (целые числа)

Результат операции выводится в поток `stdout`, ошибка в `stderr`.

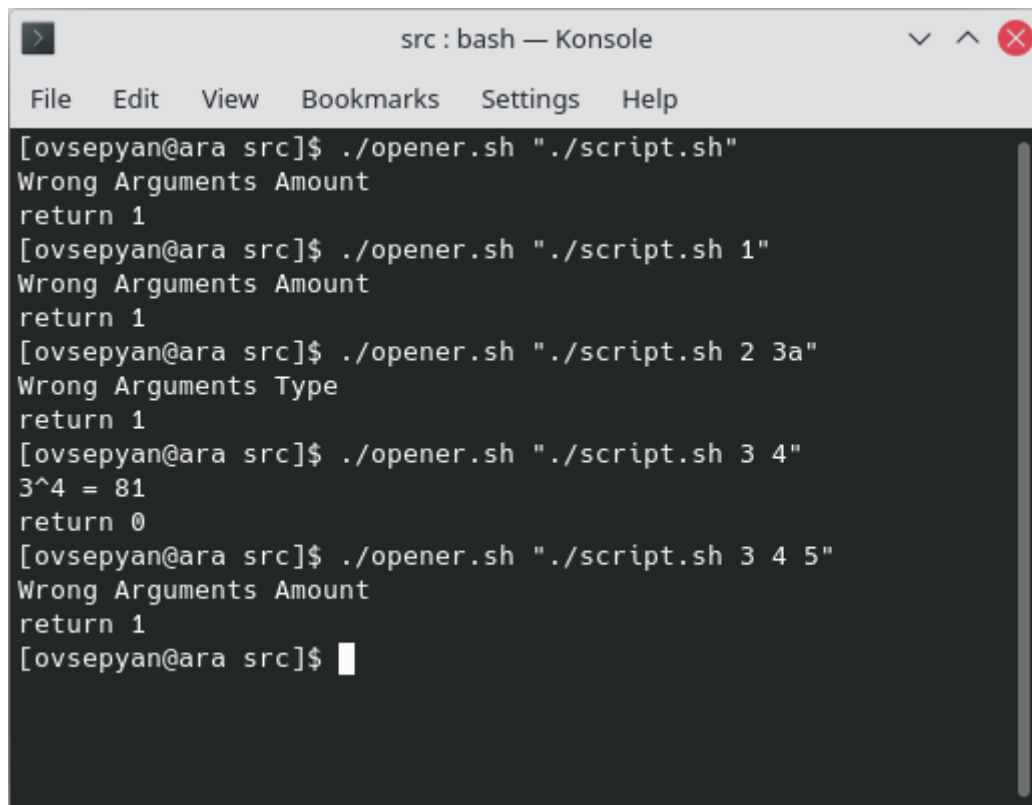
Исходный код скрипта `script.sh` в [приложении А](#).

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](#).

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

```
pacman -S shellcheck  
  
shellcheck shell.sh app.sh
```



```
src : bash — Konsole
File Edit View Bookmarks Settings Help
[ovsepyan@ara src]$ ./opener.sh "./script.sh"
Wrong Arguments Amount
return 1
[ovsepyan@ara src]$ ./opener.sh "./script.sh 1"
Wrong Arguments Amount
return 1
[ovsepyan@ara src]$ ./opener.sh "./script.sh 2 3a"
Wrong Arguments Type
return 1
[ovsepyan@ara src]$ ./opener.sh "./script.sh 3 4"
3^4 = 81
return 0
[ovsepyan@ara src]$ ./opener.sh "./script.sh 3 4 5"
Wrong Arguments Amount
return 1
[ovsepyan@ara src]$
```

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;
```