



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 (shell)”

Выполнил:
Евула А. С.,
студент группы ИУ8-63

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

Оглавление

Цель работы.....	3
Основная часть	3
1. Теоретическая часть	3
2. Практическая часть	3
Запускаемое приложение	3
Запускающее приложение.....	4
Выводы.....	4
Приложение А	5
Приложение Б.....	5

Цель работы

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

Основная часть

1. Теоретическая часть

POSIX (Portable Operating System Interface) - "Портативный Интерфейс Операционной Системы". Набор стандартов, описывающих интерфейсы между ОС и прикладной программой, библиотеку языка С и набор приложений и их интерфейсов. Необходим для обеспечения совместимости UNIX-ОС и переносимости прикладных программ.

POSIX нужен для облегчения переноса кодов на иные платформы; унификации интерфейсов; определения минимума интерфейсов программ; развития стандартов в направлениях коммуникационные сети; обработки данных, защиты информации.

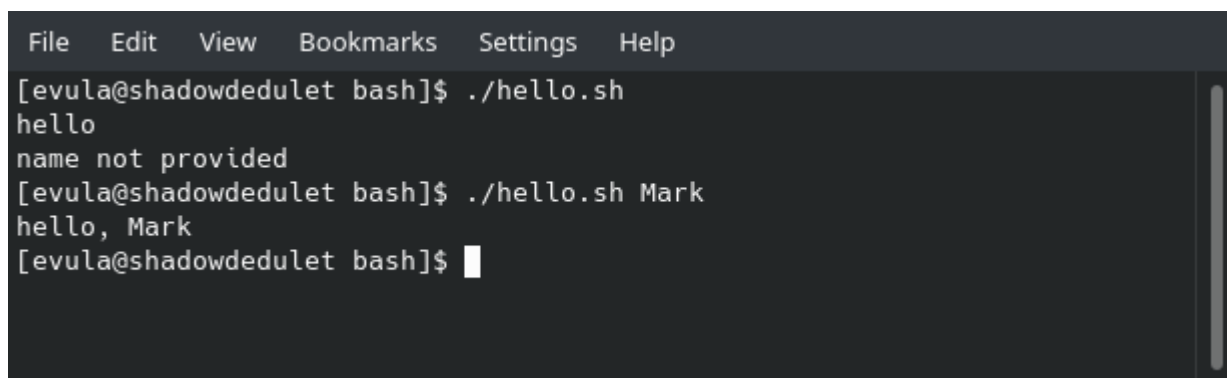
2. Практическая часть

Запускаемое приложение

Создадим скрипт hello с помощью команды `nano hello.sh`

При наличии аргументов скрипт выведет "hello, <аргумент>", иначе – "hello" и ошибку "name not provided".

Также выдадим права на него - `chmod 755 hello.sh`



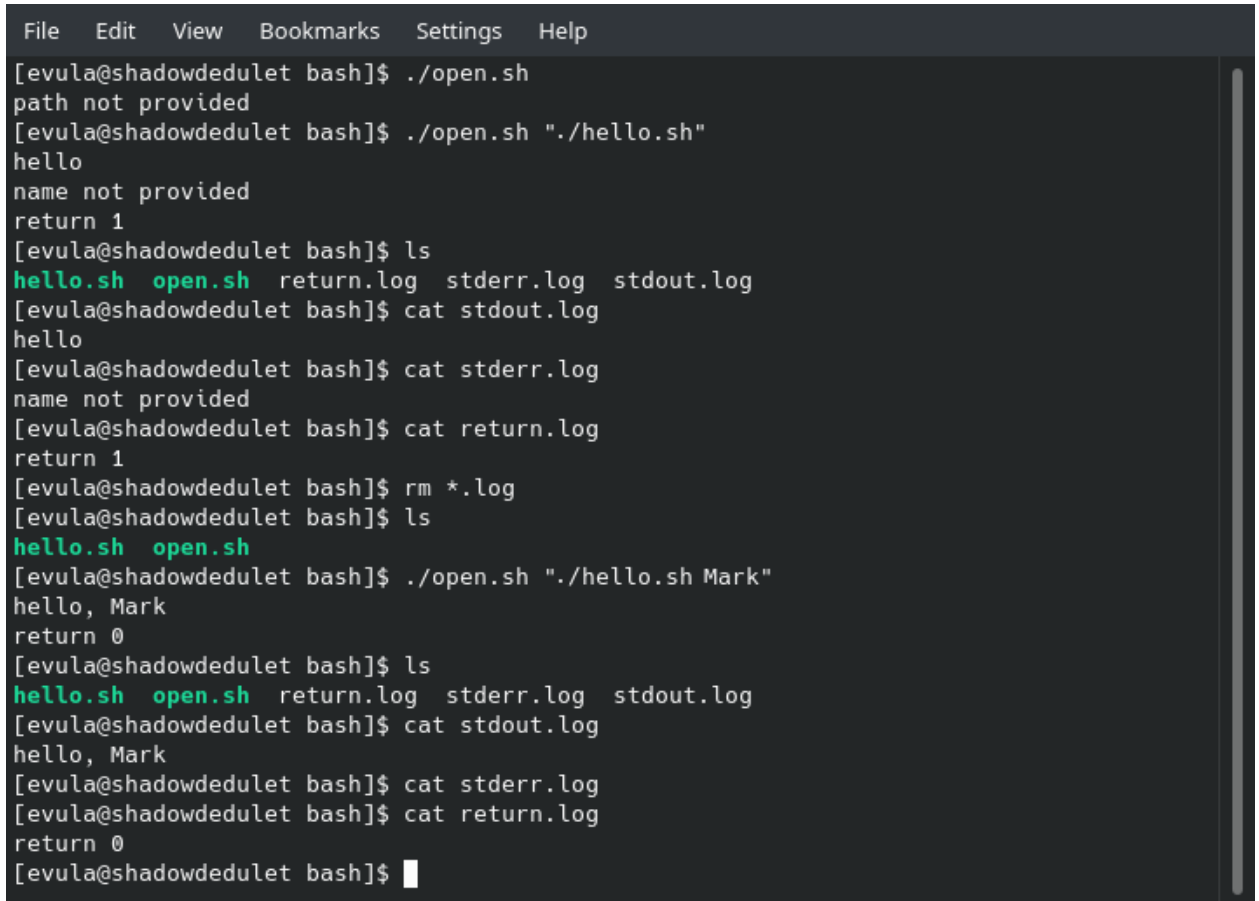
```
File Edit View Bookmarks Settings Help
[evula@shadowdedulet bash]$ ./hello.sh
hello
name not provided
[evula@shadowdedulet bash]$ ./hello.sh Mark
hello, Mark
[evula@shadowdedulet bash]$
```

Рисунок 1- использование скрипта `hello.sh` с разными аргументами

Исходный код в Приложение А.

Запускающее приложение

Создадим скрипт `open` с помощью команды `nano open.sh`, который будет запускать приложения. Поток вывода приложения переходит в `stdout.log`, ошибок переходит в `stderr.log`, код завершения – в `return.log`.



```
File Edit View Bookmarks Settings Help
[evula@shadowdedulet bash]$ ./open.sh
path not provided
[evula@shadowdedulet bash]$ ./open.sh "./hello.sh"
hello
name not provided
return 1
[evula@shadowdedulet bash]$ ls
hello.sh open.sh return.log stderr.log stdout.log
[evula@shadowdedulet bash]$ cat stdout.log
hello
[evula@shadowdedulet bash]$ cat stderr.log
name not provided
[evula@shadowdedulet bash]$ cat return.log
return 1
[evula@shadowdedulet bash]$ rm *.log
[evula@shadowdedulet bash]$ ls
hello.sh open.sh
[evula@shadowdedulet bash]$ ./open.sh "./hello.sh Mark"
hello, Mark
return 0
[evula@shadowdedulet bash]$ ls
hello.sh open.sh return.log stderr.log stdout.log
[evula@shadowdedulet bash]$ cat stdout.log
hello, Mark
[evula@shadowdedulet bash]$ cat stderr.log
[evula@shadowdedulet bash]$ cat return.log
return 0
[evula@shadowdedulet bash]$
```

Рисунок 2- использование скрипта `open.sh` с различными аргументами

Исходный код в

Приложение Б.

Выводы

Были получены знания о стандартах POSIX, получении потоков приложения, написании и использовании shell скриптов.

Приложение А

```
#!/bin/bash
# greeting script

if [ $# -gt 0 ]; then
    echo "hello, $1";
    exit 0;
fi

echo "hello";
echo "name not provided" 1>&2;
exit 1;
```

Приложение Б

```
#!/bin/bash
# open script

if [ $# -gt 0 ]; then
    app_path=$1;
    $app_path 1>stdout.log 2>stderr.log;
    echo return $?>return.log;
    cat stdout.log stderr.log return.log;
    exit 0;
fi

echo "path not provided">&2;
exit 1;
```