

ЦЕЛЬ РАБОТЫ

Следует реализовать POSIX-совместимый shell-скрипт для получения потоков запускаемого приложения (stderr/stdout).

ОСНОВНАЯ ЧАСТЬ

1.1 Теоретическая часть

POSIX — 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.

Задачи POSIX:

- содействовать облегчению переноса кода прикладных программ на иные платформы;
- способствовать определению и унификации интерфейсов заранее при проектировании, а не в процессе их реализации;
- сохранять по возможности и учитывать все главные, созданные ранее и используемые прикладные программы;
- определять необходимый минимум интерфейсов прикладных программ, для ускорения создания, одобрения и утверждения документов;
- развивать стандарты в направлении обеспечения коммуникационных сетей, распределенной обработки данных и защиты информации;
- рекомендовать ограничение использования бинарного (объектного) кода для приложений в простых системах.

Стандарт состоит из четырёх основных разделов.

1. Основные определения — список основных определений и соглашений, используемых в спецификациях, и список заголовочных файлов

языка Си, которые должны быть предоставлены соответствующей стандартной системой.

2. Оболочка и утилиты — описание утилит и командной оболочки `sh`, стандарты регулярных выражений.

3. Системные интерфейсы — список системных вызовов языка Си.

4. Обоснование — объяснение принципов, используемых в стандарте.

POSIX нейтрален по отношению к системной архитектуре и разрядности процессора.

1.2 Практическая часть

Был создан shell скрипт `Script1.sh`, реализующий довольно простой сценарий:

- Если аргументы не поданы, то выводится ошибка "Не были введены аргументы" и передается в поток `stderr`.
- Если аргумент является строкой "Погода", то погода Москвы на сегодняшний день выводится в поток `stdout`.
- Во всех остальных случаях выводится ошибка "Введена нераспознанная команда" и передается в поток `stderr`.

Описание программы:

`if [$# -eq 0]` - программа зайдет внутрь этого блока, если количество поданных аргументов равно нулю.

`echo "Не были введены аргументы" >&2` - вывод ошибки и передача ее в поток `stderr`.

`exit 1` - возврат кода завершения этой ошибки.

`elif [$1 != "Погода"] || [$# -gt 1];` - программа зайдет внутрь этого блока, если количество поданных аргументов больше одного или единственный поданный аргумент не равен строке "Погода".

`echo "Введена нераспознанная команда" >&2` - вывод ошибки и передача ее в поток `stderr`.

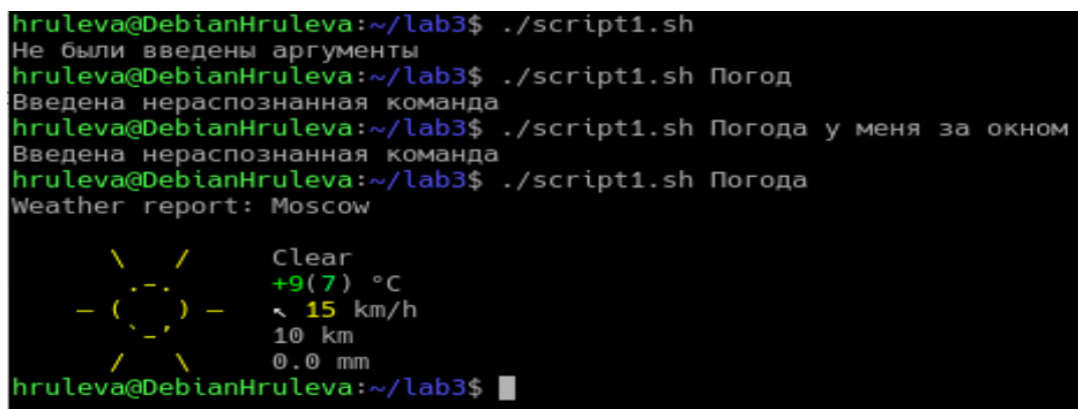
`exit 2` - возврат кода завершения этой ошибки.

Если программа не зашла ни в один из предыдущих блоков, то

`curl wttr.in/Moscow?0 >&1` - вывод погоды Москвы на сегодняшний день и передача ее в поток `stdout`.

`exit 0` - возврат успешного кода завершения.

Код shell скрипта `Script1.sh` представлен в приложении 1.



```
hruleva@DebianHruleva:~/lab3$ ./script1.sh
Не были введены аргументы
hruleva@DebianHruleva:~/lab3$ ./script1.sh Погод
Введена нераспознанная команда
hruleva@DebianHruleva:~/lab3$ ./script1.sh Погода у меня за окном
Введена нераспознанная команда
hruleva@DebianHruleva:~/lab3$ ./script1.sh Погода
Weather report: Moscow

  \  /      Clear
 - ( - )  +9(7) °C
  /  \    ^ 15 km/h
         10 km
         0.0 mm
hruleva@DebianHruleva:~/lab3$ █
```

Рисунок 1 - Демонстрация работы скрипта `Script1.sh`

Был создан shell скрипт `Script2.sh`, запускающий приложения. Поток вывода приложения переходит в `stdout.txt`, поток ошибок переходит в `stderr.txt`, а коды завершения помещаются в текстовый файл `exit.txt`.

Описание программы:

`if [$# -gt 0]` - программа зайдет внутрь этого блока, если количество поданных аргументов не равно нулю.

`path=$1` - в переменную `path` кладется название(путь) запускаемого приложения.

`$path 1>stdout.txt 2>stderr.txt` - перенаправление потока `stdout` в текстовый файл `stdout.txt`, а потока `stderr` в текстовый файл `stderr.txt`.

`echo $?>exit.txt` - получение кода завершения программы и передача его в текстовый файл `exit.txt`.

`echo "Приложение отработало. Проверьте результат"` - вывод информации о том, что приложение отработало.

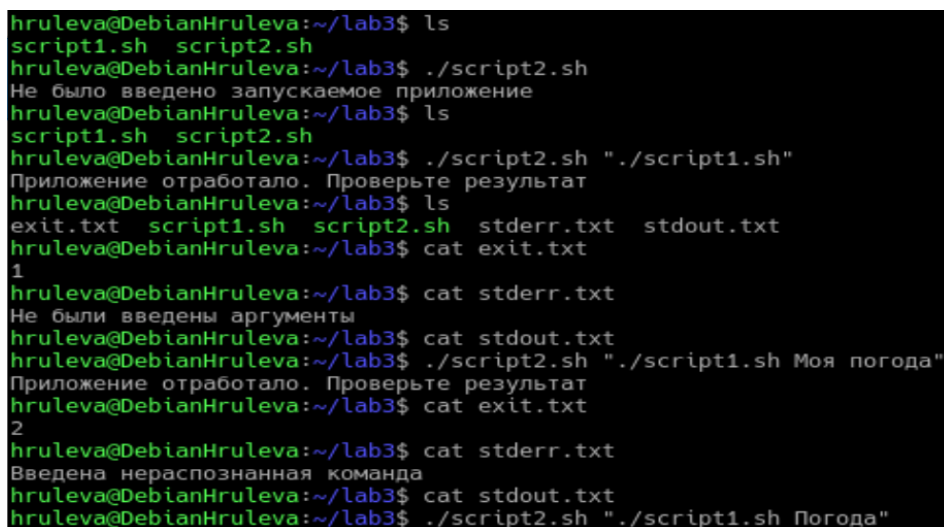
`exit 0` - возврат успешного кода завершения.

Если количество поданных аргументов было равно нулю, то выведется:

`echo "Не было введено запускаемое приложение"`

`exit 1` - возврат кода завершения ошибки.

Код shell скрипта `Script2.sh` представлен в приложении 2.



```
hruleva@DebianHruleva:~/lab3$ ls
script1.sh  script2.sh
hruleva@DebianHruleva:~/lab3$ ./script2.sh
Не было введено запускаемое приложение
hruleva@DebianHruleva:~/lab3$ ls
script1.sh  script2.sh
hruleva@DebianHruleva:~/lab3$ ./script2.sh "./script1.sh"
Приложение отработало. Проверьте результат
hruleva@DebianHruleva:~/lab3$ ls
exit.txt  script1.sh  script2.sh  stderr.txt  stdout.txt
hruleva@DebianHruleva:~/lab3$ cat exit.txt
1
hruleva@DebianHruleva:~/lab3$ cat stderr.txt
Не были введены аргументы
hruleva@DebianHruleva:~/lab3$ cat stdout.txt
hruleva@DebianHruleva:~/lab3$ ./script2.sh "./script1.sh Моя погода"
Приложение отработало. Проверьте результат
hruleva@DebianHruleva:~/lab3$ cat exit.txt
2
hruleva@DebianHruleva:~/lab3$ cat stderr.txt
Введена нераспознанная команда
hruleva@DebianHruleva:~/lab3$ cat stdout.txt
hruleva@DebianHruleva:~/lab3$ ./script2.sh "./script1.sh Погода"
```

Рисунок 2 - Демонстрация работы скрипта `Script2.sh`

```

hruleva@DebianHruleva:~/lab3$ ./script2.sh "./script1.sh Погода"
Приложение отработало. Проверьте результат
hruleva@DebianHruleva:~/lab3$ cat exit.txt
0
hruleva@DebianHruleva:~/lab3$ cat stdout.txt
Weather report: Moscow

  .-.      Light rain shower
 (  ) .    +9(6) °C
(  _)(  _ ) ~ 19 km/h
  ' ' ' '  9 km
  ' ' ' '  3.7 mm
hruleva@DebianHruleva:~/lab3$ █

```

Рисунок 3 - Демонстрация работы скрипта Script2.sh

ВЫВОДЫ

Таким образом, в ходе лабораторной работы был реализован POSIX-совместимый shell-скрипт для получения потоков запускаемого приложения (stderr/stdout).