|  |
| --- |
| Университет итмо, кафедра вт |
| Лабораторная работа №1 по дисциплине «Основы системного программирования» |
| Вариант 1 |
| Группа Р3302 |
| **Выполнила: Орлова Кристина Александровна** |
| **Преподаватель: Жмылёв Сергей Александрович** |

|  |
| --- |
| *21.09.18* |

Задание

Написать интерактивную программу (скрипт) на языке shell, которая выводит список действий с номерами и ожидает ввода номера пункта, после чего начинает выполнение заданных в этом пункте команд. Скрипт должен корректно обрабатывать ситуацию окончания входного потока данных и позволять работать с относительными и абсолютными именами файлов и каталогов, которые могут содержать: пробелы и символы табуляции; символы '\*', '?', '[', ']', '-', '~', '$' или состоять только из них.   
  
Для команд удаления необходимо запрашивать собственное подтверждение дополнительно. Текст запроса должен соответствовать системному (выводимому командой rm). Ожидание ввода ответа происходит на новой строке.   
  
Организовать обработку ошибок, например, при отсутствии удаляемого файла, для чего перенаправить вывод системных сообщений об ошибках в файл-журнал с именем lab1\_err, расположенном в домашней директории, и выдать свое сообщение на стандартный поток ошибок.   
  
Каждый запрос на ввод должен сопровождаться сообщением на отдельной строке, содержащим информацию о том, какие данные должны быть введены. Ожидание ввода происходит с новой строки.   
  
Выход из скрипта должен осуществляться только при выборе соответствующего пункта меню или окончании стандартного потока ввода.   
  
Каждый аргумент должен подаваться программе отдельной строкой. Например, нужно скопировать файл, тогда действия пользователя должны быть следующие:

1. Ввод числа, который соответствует пункту меню "Скопировать файл". Enter.
2. Ввод имени файла, который нужно скопировать. Enter.
3. Ввод пути, куда этот файл должен быть скопирован. Enter.

Меню должно выводиться при запуске скрипта и после каждого выполнения пункта.   
  
Исходный код лабораторной работы должен находиться в системе контроля версий, развёрнутой на helios. (`svn help`, `git help`).

**Вариант #1**

1. Напечатать имя текущего каталога
2. Сменить текущий каталог
3. Напечатать содержимое текущего каталога
4. Создать файл
5. Удалить файл
6. Выйти из программы

Исходный код

|  |
| --- |
| #!/usr/bin/ksh |
|  |  |
|  | menu="1. Print the name of the current directory |
|  | 2. Change current directory |
|  | 3. Print the contents of the current directory |
|  | 4. Create the file |
|  | 5. Delete the file |
|  | 6. Exit the program" |
|  |  |
|  | errorFile=$HOME/lab1\_err |
|  | errorFileSize=`cat $errorFile | wc -c` |
|  |  |
|  | function outputError { |
|  | echo "\033[31;40m$1\033[0m" >&2 |
|  | } |
|  |  |
|  | function output { |
|  | echo "\033[36;40m$1\033[0m" |
|  | } |
|  |  |
|  | function checkErrorPresence { |
|  | currentErrorFileSize=`cat $errorFile | wc -c` |
|  |  |
|  | if [ "$currentErrorFileSize" -gt "$errorFileSize" ] |
|  | then |
|  | outputError "$@" |
|  | fi |
|  |  |
|  | errorFileSize=$currentErrorFileSize |
|  | } |
|  |  |
|  | function checkOnExist { |
|  | if [ -"$2" "$1" ] |
|  | then |
|  | echo 0 |
|  | else |
|  | echo 1 |
|  | fi |
|  | } |
|  |  |
|  | function startNewLoop { |
|  | echo "\033[36;40m\n $menu\n\033[0m" |
|  |  |
|  | read commandNumber |
|  |  |
|  | case $commandNumber in |
|  | 1) |
|  | pwd |
|  | startNewLoop;; |
|  |  |
|  | 2) |
|  | output "Input the path to directory" |
|  | IFS= read -r path |
|  |  |
|  | if [ `echo "$path" | egrep "^$" | wc -l` -eq 1 ] |
|  | then |
|  | outputError "Empty path" |
|  | else |
|  | if [ `echo "$path" | egrep "^-" | wc -l` -eq 1 ] |
|  | then |
|  | path="./$path" |
|  | fi |
|  |  |
|  | resultExist=$(checkOnExist "$path" "d") |
|  |  |
|  | if [ "$resultExist" -eq 1 ] |
|  | then |
|  | outputError "There is no such path to directory" |
|  | else |
|  | cd -- "$path" 2>> $errorFile |
|  |  |
|  | checkErrorPresence "Permission denied" |
|  | fi |
|  | fi |
|  |  |
|  | startNewLoop;; |
|  |  |
|  | 3) |
|  | ls 2>> $errorFile |
|  |  |
|  | checkErrorPresence "Permission denied" |
|  |  |
|  | startNewLoop;; |
|  |  |
|  | 4) |
|  | output "Input the path to file" |
|  | IFS= read -r path |
|  |  |
|  | if [ `echo "$path" | egrep "^$" | wc -l` -eq 1 ] |
|  | then |
|  | outputError "Empty path" |
|  | else |
|  | touch -- "$path" 2>> $errorFile |
|  |  |
|  | checkErrorPresence "There is no such path" |
|  | fi |
|  |  |
|  | startNewLoop;; |
|  |  |
|  | 5) |
|  | output "Input the path to file" |
|  | IFS= read -r path |
|  |  |
|  | if [ `echo "$path" | egrep "^$" | wc -l` -eq 1 ] |
|  | then |
|  | outputError "Empty path" |
|  | else |
|  | resultExist=$(checkOnExist "$path" "f") |
|  |  |
|  | if [ "$resultExist" -eq 1 ] |
|  | then |
|  | outputError "There is no such file" |
|  | else |
|  | output "rm: remove $path (yes/no)?" |
|  | read answer |
|  |  |
|  | if [ `echo "$answer" | egrep "^y(e|es)?$" | wc -l` -eq 1 ] |
|  | then |
|  | rm -- "$path" 2>> $errorFile |
|  |  |
|  | checkErrorPresence "Permission denied" |
|  | fi |
|  | fi |
|  | fi |
|  |  |
|  | startNewLoop;; |
|  |  |
|  | 6) |
|  | return 0;; |
|  |  |
|  | \*) |
|  | output "Wrong. Try again (exit by the 6)" |
|  |  |
|  | startNewLoop;; |
|  | esac |
|  | } |
|  |  |
|  | startNewLoop |

Выводы

Я вспомнила, как писать примитивные bash скрипты; некоторые команды, которые мне пришлось использовать при написании данного скрипта.

В этот раз я писала скрипт даже с использованием функций, передачи им аргументов и возврата значения.

Также я вспомнила всю ~~боль~~ радость от написания bash скриптов в vim и даже открыла для себя некоторые новые возможности работы с ним (копирование и вставку).

Более того, я получила незаменимый опыт чтения чужих bash скриптов, а именно lcheck и chech\_1 (скрипты тестировщика), и дебага их с помощью echo в журнал логов.