Лабораторная работа № 7. Оболочка и скрипты

6 января 2024 г.

Семён Чайкин, ИУ9-11Б

Цель работы

Получение навыков написания сценариев на «скриптовых» языках.

Задание 1

На `Bash` напишите скрипт, который будет запускать долго выполняющуюся программу (напишите скрипт, имитирующий такую программу, скажем, просто ожидающий несколько минут и завершающийся) строго каждые `t` минут, но так, чтобы одновременно выполнялось не более `1` экземпляра этой программы. Путь к программе и периодичность запуска передавайте в виде аргументов командной строки. Вывод и ошибки запускаемой программы направляйте в файлы, имена этих файлов формируйте автоматически. Запускаемую программу запрещается убивать.

Реализация

```
#!/bin/bash

[[ $# = 2 ]] || { echo "usage: ${0} <delay> <command>"; exit 0; }

cmd="${2}"
[[ -x $cmd ]] || { echo "error: can't execute $cmd"; exit 1; }

filename=$(basename $cmd)
filename="${filename%.*}"
errfile="$filename-err.txt"
outfile="$filename-out.txt"

while true; do
  ( [[ `ps ax | grep -v grep | grep -v "${0}" | grep $cmd | wc -l` -gt 0 ]] &&
    { echo "Program is still running... Can't restart now"; } ) ||
    { echo "Restarting program..."; ./$cmd 2&>> $errfile 1&>> $outfile & }
  sleep $1
done
```

Тестирование

Тестовый файл `test.sh`:

```
#!/bin/bash

echo "Hello, world!"
sleep 5
```

Результат в консоли:

```
[trololo@trololo 1]$ ls
1.sh test.sh
```

```
[trololo@trololo 1]$ chmod +x 1.sh
[trololo@trololo 1]$ chmod +x test.sh
[trololo@trololo 1]$ ./1.sh
usage: ./1.sh <delay> <command>
[trololo@trololo 1]$ ./1.sh 3 ./test.sh
Restarting program...
Program is still running... Can't restart now
Restarting program...
Program is still running... Can't restart now
Restarting program...
Program is still running... Can't restart now
^C
[trololo@trololo 1]$ ls
1.sh test-err.txt test-out.txt test.sh
[trololo@trololo 1]$ cat test-out.txt
Hello, world!
Hello, world!
Hello, world!
```

Задание 2

На Bash напишите скрипт, который принимает путь к проекту на языке C и выводит общее число непустых строк во всех файлах .c и .h указанного проекта. Предусмотрите рекурсивный обход вложенных папок.

Реализация

```bash
#!/bin/bash

rec() {
  local result=0
  if [ -d "$1" ]; then
    while read name; do
      [[ $name != "" ]] && { result=$(($result+`rec "$1/$name"`)); }
    done <<< $(ls "$1")
    echo $result
  else
    name=$(basename "$1")
    ext="${name##*.}"
    if [ "$ext" = "c" ] || [ "$ext" = "h" ]; then
      echo `cat "$1" | sed '/^\s*$/d' | wc -l`
    else
      echo 0
    fi
  fi
}

[[ "$1" = "" ]] && rec "$PWD" || rec "$1"
```

Тестирование

```
[trololo@trololo 2]$ chmod +x 2.sh
[trololo@trololo 2]$ ls
2.sh
[trololo@trololo 2]$ ./2.sh
0
[trololo@trololo 2]$ mkdir -p {a,b,c}/{d,e,f}/{g,h,i}
```

```
[trololo@trololo 2]$ tee {a,b,c}/{d,e,f}/{g,h,i}/test.txt << EOF >> /dev/null
> First line
> Second line
> Third line
> EOF
[trololo@trololo 2]$ tee {a,b,c}/{d,e,f}/{g,h,i}/test.h << EOF >> /dev/null
> Как с йети быть?
> Ети? Надо чаще мыть.
> Да нет, я про снежного человека.
> А-а! А это надо с контр-адмиралом посоветоваться, он Атлантиду видел.
> EOF
[trololo@trololo 2]$ tee {a,b,c}/{d,e,f}/{g,h,i}/test.c << EOF >> /dev/null
> Мне всегда задают три вопроса:
> почему я в армии, сколько мне лет
> и отчего у меня волосы на груди окрасились.
>
> Начну с последнего: волосы у меня на груди окрасились,
> потому что я пролил на них ракетный окислитель.
>
> Лет мне двадцать девять, скоро юбилей. А в армии я потому,
> что меня жена с тёщей хотели в сумасшедший дом отдать — за убеждения.
> EOF
[trololo@trololo 2]$ ls
2.sh a b c
[trololo@trololo 2]$ ./2.sh
297
```

Задание 3

На выбранном скриптовом языке напишите программу, которая выводит в консоль указанное число строк заданной длины, состоящих из латинских букв, цифр и печатных знаков, присутствующих на клавиатуре. Длину строки и число строк передавайте как аргументы командой строки. Для каких целей можно использовать такую программу? Оформите логику приложения в виде отдельной функции и поместите её в отдельный модуль.

Реализация

Файл `generate_strings.rb`:

```ruby
def generate_strings(n, length)
  if n <= 0
    STDERR.puts "error: invalid number of strings (expected positive number, got: #{n})"
    exit -1
  end

  if length <= 0
    STDERR.puts "error: invalid string length (expected positive number, got: #{length})"
    exit -1
  end

  @printable ||= (?!..?~).reduce(:+)
  n.times.map{ length.times.map{ @printable[rand(@printable.size)] }.join }
end
```

Тестирование

Тестовый файл `3.rb`

```ruby
#!/usr/bin/env ruby

require_relative "generate_strings"

if ARGV.size != 2
  puts "usage: #{$0} <number of strings> <string length>"
  exit
end

n = ARGV[0].to_i
length = ARGV[1].to_i

puts generate_strings n, length
```

Результат в консоли:

```
[trololo@trololo 3]$ ls
3.rb generate_strings.rb
[trololo@trololo 3]$ chmod +x 3.rb
[trololo@trololo 3]$ ./3.rb
usage: ./3.rb <number of strings> <string length>
[trololo@trololo 3]$ ./3.rb 3 17
f%?`LmNqgDsWUu=,!
d[j<|G]s8M*x_)1O'
ay*i/0]KJU==:D8`e
[trololo@trololo 3]$ ./3.rb 2 5
.:Fj}
siyS'
```

Задание 4

На выбранном скриптовом языке напишите функцию, которая принимает произвольную чистую функцию с переменным числом аргументов и возвращает мемоизованную версию этой функции. Для запоминания результатов вычислений выберете подходящую структуру данных из числа встроенных классов выбранного языка.

Реализация

Файл `memoization.rb`:

```ruby
def memoize(name)
  @lookup ||= Hash.new { |h, k| h[k] = {} }

  f = singleton_class.instance_method(name)

  define_singleton_method name do |*args|
    @lookup[name][args] =
      @lookup[name].has_key?(args) ?
        @lookup[name][args] : f.bind(self).call(*args)
  end
end
```

Тестирование

Тестовый файл `4.rb`:

```ruby
#!/usr/bin/env ruby

require_relative "memoization"
```

```ruby
memoize def fib(n)
  n <= 1 ? n : fib(n-1) + fib(n-2)
end

memoize def fact(n)
  n <= 1 ? 1 : n * fact(n-1)
end

n = gets.to_i

n.times.each {|i| [fib(i), fact(i)] }

a = fib n
b = fact n

puts "#{a} [#{a.to_s.chars.map(&:to_i).sum}]"
puts "#{b} [#{b.to_s.chars.map(&:to_i).sum}]"
```

Результат в консоли:

```
[trololo@trololo 4]$ ls
4.rb memoization.rb
[trololo@trololo 4]$ chmod +x 4.rb
[trololo@trololo 4]$ ./4.rb
50
12586269025 [46]
30414093201713378043612608166064768844377641568960512000000000000 [216]
[trololo@trololo 4]$ time ./4.rb <<< 10000
3364476487...366875 [9123]
2846259680...000000 [149346]

real    0m0.291s
user    0m0.199s
sys     0m0.088s
```

Вывод

Ознакомился со скриптовыми языками `Bash` и `Ruby`, а также научился их применять для написания необходимых скриптов.