NC State University
ECE Department ECE 720 W. Rhett Davis

Homework #1 Solution

1. The commands and return values for each problem are given below
   a. the number of files that do not have the filename extension ".h"

      soln:
      ```
      find /afs/eos/dist/ds5-2013.06/FastModelsTools_8.2/OSCI/SystemC
             -type f | grep -v '\.h$' | wc
      ```
      ans: 194

   b. the number of occurrences of the string "class" in all files with the extension ".h"

      soln:
      ```
      find /afs/eos/dist/ds5-2013.06/FastModelsTools_8.2/OSCI/SystemC
                -name "*.h" -exec grep -o class {} \; | wc
      ```
      ans: 2301

2. This can be done with a four-line script, as follows:

   ```
   import os

   if not
   os.access('subdir',os.F_OK):
     os.mkdir('subdir')
   ```

3. The following mymonths.py script satisfies the constraints:

   ```
   import months

   class monthlist(months.monthlist):
       def __init__(self):
           months.monthlist.__init__(self)

       def listall(self):
           keys = self.dict.keys()
           keys.sort()
           for k in keys:
               print self.dict[k]
   ```

   The output of the mytop.py script is shown below:

   ```
    4     April 30
    8    August 31
   12  December 31
    2  February 28
    1   January 31
    7      July 31
    6      June 30
    3     March 31
    5       May 31
   11  November 30
   10   October 31
    9 September 30
   ```

4. The script below finds the average time of the address-phase for each transaction, separated by the initiator ID:

```
import re

f=file("sim.out")

start_time = {}
summed_time = {}
txn_count = {}

for line in f:
    m=re.search(r"select_initiator.cpp:\s+(\d+)",line)
    if m:
        time=m.group(1)
        continue
    m=re.search(r"(\d+)\s+starting new transaction",line)
    if m:
        id=m.group(1)
        start_time[id]=int(time)
    m=re.search(r"(\d+)\s+transaction waiting begin-response",line)
    if m:
        id=m.group(1)
        if id not in summed_time:
            summed_time[id]=int(time)-start_time[id]
        else:
            summed_time[id]=summed_time[id]+int(time)-start_time[id]
        if id not in txn_count:
            txn_count[id]=1
        else:
            txn_count[id]=txn_count[id]+1

for id in summed_time:
  print id, float(summed_time[id])/float(txn_count[id])
```

The average time for initiator 101 is 4.6 ns, and for initiator 102 is 5.1 ns. The system apparently gives transactions from initiator 101 a higher priority.

5. The script below determines the depth of the tree in the tree.txt file:

```
import re

f=file('tree2.txt')

verts={}

for line in f:
    m=re.search(r"^(\w):(.*)$",line)
    if m:
      lastvert=m.group(1).strip()
      verts[lastvert]= m.group(2).strip().split()
    else:
      print line

def recursion(v):
    depth=1
    for child in verts[v]:
        depth=max(depth,recursion(child)+1)
    return depth

print 'Tree Depth:',recursion(lastvert)
```