COMP 312 Assignment 5

Daniel Braithwaite

April 11, 2016

1 Python

1.1 Program 1

```
from SimPy.Simulation import *
import random
import numpy
import math

## Useful extras ----------
def conf(L):
    """confidence interval"""
    lower = numpy.mean(L) - 1.96*numpy.std(L)/math.sqrt(len(L))
    upper = numpy.mean(L) + 1.96*numpy.std(L)/math.sqrt(len(L))
    return (lower,upper)

## Model ----------
class Source(Process):
    """generate random arrivals"""
    def run(self, N, lamb, mu):
        for i in range(N):
            a = Arrival(str(i))
            activate(a, a.run(mu))
            t = random.expovariate(lamb)
            yield hold, self, t

class Arrival(Process):
    n = 0

    """an arrival"""
    def run(self, mu):
        arrivetime = now()

        Arrival.n += 1
        if Arrival.n < len(G.custmon):
            G.custmon[Arrival.n].observe(now())

        yield request, self, G.server
        t = random.expovariate(mu)
        yield hold, self, t
        yield release, self, G.server
        delay = now()-arrivetime
        G.delaymon.observe(delay)

        Arrival.n -= 1
```

```
        if Arrival.n < len(G.custmon):
            G.custmon[Arrival.n].observe(now())



class G:
    server = 'dummy'
    delaymon = 'Monitor'
    custmon = 'Monitor'

def model(c, N, K, lamb, mu, maxtime, rvseed):
    # setup
    initialize()
    random.seed(rvseed)
    G.server = Resource(c, monitored=True)
    G.delaymon = Monitor()
    G.custmon = []
    for i in range(K):
        G.custmon.append(Monitor())

    # simulate
    s = Source('Source')
    activate(s, s.run(N, lamb, mu))
    simulate(until=maxtime)

    # gather performance measures
    W = G.delaymon.mean()
    #L = G.server.waitMon.timeAverage() + G.server.actMon.timeAverage()
    C = []
    for mon in G.custmon:
        C.append(mon.timeAverage())

    return(W, C)

## Experiment ----------
allW = []
allPI = {}
for i in range(10):
    allPI[i] = []

for k in range(50):
   seed = 123*k
   result = model(c=1, N=10000, K=10, lamb=2, mu=3, maxtime=2000000, rvseed=seed
   allW.append(result[0])

   s = sum(result[1])
```

```
    for i in range(10):
        allPI[i].append(result[1][i] / s)
    #allWait.append(result[2])
    #print result[1]
    #print "\n"

#print allW
print ""
print "Estimate of W:",numpy.mean(allW)
print "Conf int of W:",conf(allW)

#print allW
print ""
print "Estimate of PI 0:",numpy.mean(allPI[0])
print "Conf int of PI 0:",conf(allPI[0])
```

For this problem i added the monitors but wasnt sure how to find the values

1.2 Program 2

```
import numpy
import matplotlib.pyplot as pl

def graph(func, l, m, n):
    x = numpy.linspace(l, m, n, endpoint=True)
    y = [func(z) for z in x]

    pl.plot(x, y)
    pl.savefig('graph.png')

def f(x):
    if x == 0:
        return 1

    return numpy.sin(numpy.pi * x) / numpy.pi * x

graph(f, -numpy.pi, numpy.pi, 256)
```