import os,sys

import statistics

import subprocess

import time

performance\_measurement\_overhead = 34

BUILD\_FOLDER = "../build/"

SECONDS\_PER\_CYCLE = 0.556

def measure\_ram\_access():

for i in [1024, 2048, 16\*1024, 32\*1024, 64\*1024, 128\*1024, 256\*1024, 512\*1024, 1024\*1024, 1024\*1024\*2, 1024\*1024\*3, 1024\*1024\*4, 1024\*1024\*8, 1024\*1024\*64, 1024\*1024\*128, 1024\*1024\*256]:

clock\_cycles = []

for j in range(0,10):

output = subprocess.getoutput(BUILD\_FOLDER + 'ram\_access\_measurement ' + str(i))

if len(output.split(' ')) >= 3:

clock\_cycles.append(float(output.split(' ')[2]))

print ("------------------------------------------------------------")

print ("RAM Access Time",i)

print ("RAM Latency(Mean):",statistics.mean(clock\_cycles))

print( "RAM Latency(Variance):",statistics.pvariance(clock\_cycles))

print ("RAM Latency Overhead(Standard Deviation):",statistics.pstdev(clock\_cycles))

print ("------------------------------------------------------------")

def measure\_performance\_measure\_overhead():

output = subprocess.getoutput(BUILD\_FOLDER + 'performance\_measurement\_overhead' )

print (output)

output = subprocess.getoutput(BUILD\_FOLDER + 'loop\_overhead\_measurement' )

print (output)

def measure\_system\_call():

clock\_cycles = []

for i in range(0,10000):

output = subprocess.getoutput(BUILD\_FOLDER + 'system\_call\_measurement' )

if len(output.split(' ')) >= 3:

clock\_cycles.append(int(output.split(' ')[2]))

print ("System Call Overhead(Mean):",statistics.mean(clock\_cycles))

print ("System Call Overhead(Variance):",statistics.pvariance(clock\_cycles))

print ("System Call Overhead(Standard Deviation):",statistics.pstdev(clock\_cycles))

def measure\_procedure\_call():

for i in range(0,8):

output = subprocess.getoutput(BUILD\_FOLDER + 'procedure\_call' + str(i) + '\_measurement' )

print ("Arguments " + str(i))

print (output)

time.sleep(1)

def measure\_task\_creation():

output = subprocess.getoutput(BUILD\_FOLDER + 'task\_creation\_fork\_measurement' )

print (output)

output = subprocess.getoutput(BUILD\_FOLDER + 'task\_creation\_pthread\_measurement' )

print (output)

def measure\_context\_switch():

output = subprocess.getoutput(BUILD\_FOLDER + 'context\_switch\_fork\_measurement' )

print (output)

output = subprocess.getoutput(BUILD\_FOLDER + 'context\_switch\_pthread\_measurement' )

print (output)

def measure\_ram\_bandwidth():

bandwidth = []

for i in range(0,10):

output = subprocess.getoutput(BUILD\_FOLDER + 'memory\_read\_bandwidth\_measure' )

clock\_cycles = int(output.strip())

result = 1.0/(clock\_cycles\*SECONDS\_PER\_CYCLE\*pow(10,-9))

bandwidth.append(result)

print ("Memory Read Bandwidth(Mean):",statistics.mean(bandwidth),"GB/s")

print ("Memory Read Bandwidth(Variance):",statistics.pvariance(bandwidth))

print ("Memory Read Bandwidth(Standard Deviation):",statistics.pstdev(bandwidth),"GB/s")

print ("------------------------------------------------------------")

bandwidth = []

for i in range(0,10):

output = subprocess.getoutput(BUILD\_FOLDER + 'memory\_write\_bandwidth\_measure' )

clock\_cycles = int(output.strip())

result = 1.0/(clock\_cycles\*SECONDS\_PER\_CYCLE\*pow(10,-9))

bandwidth.append(result)

print ("Memory Write Bandwidth(Mean):",statistics.mean(bandwidth),"GB/s")

print ("Memory Write Bandwidth(Variance):",statistics.pvariance(bandwidth))

print ("Memory Write Bandwidth(Standard Deviation):",statistics.pstdev(bandwidth),"GB/s"

)

def measure\_page\_fault():

output = subprocess.getoutput(BUILD\_FOLDER + 'page\_fault\_measurement' )

print (output)