НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»

ФАКУЛЬТЕТ ІНФОРМАТИКИ І ОБЧИСЛЮВАЛЬНОЇ ТЕХНІКИ

КАФЕДРА ОБЧИСЛЮВАЛЬНОЇ ТЕХНІКИ

**Лабораторна робота №8**

з дисципліни **«**Паралельні та розподілені обчислення**»**

**Процеси у мові Python**

Виконав:

студент 3 курсу гр. ІО-31

Долинний Олександр

№ ЗК 3110

Перевірив:

Корочкін О. В.

Київ 2015 р.

***Завдання:***

1.1 3.21 3.11

F1: A = sort(B) \* (MB \* MC)

F2: W = sort(R \* MT) \* (MX \* MS)

F3: R = sort(S + T) \* trans(MS \* MR)

***Лістинг програми:***

"""

Parallel and distributed computing

Lab 8: Processes in Python

Dolinniy Alexandr

IO-31

22.11.2015

Func1: A = sort(B)\*(MB\*MC)

Func2: W = sort(R\*MT)\*(MX\*MS)

Func3: R = sort(S+T)\*trans(MS\*MR)

"""

from multiprocessing import Process

N=4

def InputVector():

vector = [1 for col in range(N)]

return vector

def InputMatrix():

matrix = [[1 for row in range(N)] for col in range(N)]

return matrix

def OutputVector(vector):

if N <= 5:

print(vector)

def OutputMatrix(matrix):

if N <= 5:

for i in range(N):

print(matrix[i])

def AddVectors(vector1, vector2):

result = InputVector()

for i in range(N):

result[i] = vector1[i] + vector2[i]

return result

def SortVector(vector):

for i in range(N):

for j in range(i, N):

if (vector[i] > vector[j]):

buffer = vector[i]

vector[i] = vector[j]

vector[j] = buffer

return vector

def MultiplyVectorOnMatrix(vector, matrix):

result = InputVector()

for i in range(N):

result[i] = 0

for j in range(N):

result[i] = result[i] + vector[j]\*matrix[j][i]

return result

def MultiplyMatrices(matrix1, matrix2):

result = [[0 for row in range(len(matrix1))] for col in range(len(matrix2[0]))]

for i in range(len(matrix1)):

for j in range(len(matrix2[0])):

for k in range(len(matrix2)):

result[i][j] += matrix1[i][k]\*matrix2[k][j]

return result

def TransMatrix(matrix):

for i in range(N):

for j in range(i, N):

buf = matrix[i][j]

matrix[i][j] = matrix[j][i]

matrix[j][i] = buf

return matrix

def F1():

print("Function 1 started.")

B = InputVector()

MB = InputMatrix()

MC = InputMatrix()

SortVector(B)

MD = MultiplyMatrices(MB, MC)

A = MultiplyVectorOnMatrix(B, MD)

OutputVector(A)

print("Function 1 ended.")

def F2():

print("Function 2 started.")

R = InputVector()

MT = InputMatrix()

MX = InputMatrix()

MS = InputMatrix()

K = MultiplyVectorOnMatrix(R, MT)

SortVector(K)

ML = MultiplyMatrices(MX, MS)

W = MultiplyVectorOnMatrix(K, ML)

OutputVector(W)

print("Function 2 ended.")

def F3():

print("Function 3 started.")

S = InputVector()

T = InputVector()

MS = InputMatrix()

MR = InputMatrix()

K = AddVectors(S, T)

SortVector(K)

ML = MultiplyMatrices(MS, MR)

TransMatrix(ML)

R = MultiplyVectorOnMatrix(K, ML)

print (R)

print("Function 3 ended.")

if \_\_name\_\_ == "\_\_main\_\_":

print("Main started.")

process1 = Process(target=F1)

process2 = Process(target=F2)

process3 = Process(target=F3)

process1.start()

process2.start()

process3.start()

process1.join()

process2.join()

process3.join()

print("Main ended.")

input()