МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

НИЖЕГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ

УНИВЕРСИТЕТ им. Р.Е.АЛЕКСЕЕВА

Институт радиоэлектроники и информационных технологий

ОТЧЕТ

по практической работе

по дисциплине

Параллельные вычисления

РУКОВОДИТЕЛЬ:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_ Мартынов Д.С.\_\_

(подпись) (фамилия, и.,о.)

СТУДЕНТ:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_ Сухоруков В.А.\_\_\_

(подпись) (фамилия, и.,о.)

\_\_\_\_\_\_\_19-В-2\_\_\_\_\_\_\_\_

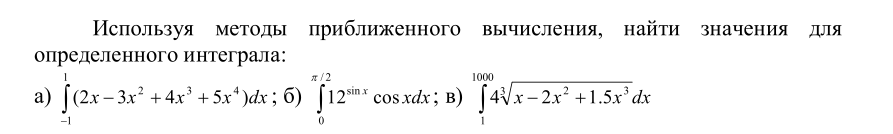
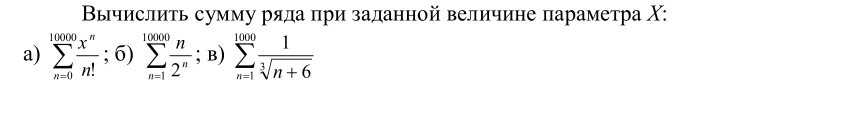
(шифр группы)

Работа защищена «\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_

С оценкой \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Нижний Новгород 2021

# Задание

1. 
2. 

# Решение задачи №1

Функции для вычисления интегралов

import math

def integral\_first(a,b):

h=(b-a)/100

x=a

summa=0.0

for i in range(100):

y=2\*x-3\*x\*x+4\*x\*x\*x+5\*x\*x\*x\*x

summa=summa+y

x=x+h

res=h\*summa

print('интеграл 2\*x-3\*x\*\*2+4\*x\*\*3+5\*x\*\*4 dx с пределами a=%d b=%d'%(a,b),

' равен %.5f\n' % (res))

# END integral\_first ------------------------------------------------

def integral\_second(a,b):

h=(b-a)/100

x=a

summa=0

for i in range(100):

y=12\*\*(math.sin(x))\*math.cos(x)

summa=summa+y

x=x+h

res=h\*summa

print('интеграл 12\*\*sin(x)\*cos(x) dx с пределами a=%d'%(a),

'b=%.5f равен %.5f\n' % (b,res))

# END integral\_second ------------------------------------------------

def integral\_third(a,b):

h=(b-a)/100

x=a

summa=0

for i in range(100):

y=4\*((x-2\*x\*x+1.5\*x\*\*3)\*\*(1/3))

summa=summa+y

x=x+h

res=h\*summa

print('интеграл 4\*((x-2\*x\*x+1.5\*x\*\*3)\*\*(1/3)) dx с пределами a=%d b=%d'%(a,b),

' равен %.5f\n' % (res))

# END integral\_third ------------------------------------------------

Решение с использование потоков

import threading

import time

#Замер времени исполнения программы

start\_time = time.time()

#Вывод информации о программе

print("\tПрограмма считает три интеграла методом левых прямоугольников.")

print("\tВычисления производятся в трёх потоках.\n")

# Запуск функций в отдельных потоках

t1 = threading.Thread(target=integral\_first, args=(-1,1,))

t1.start()

t2 = threading.Thread(target=integral\_second, args=(0,math.pi/2,))

t2.start()

t3 = threading.Thread(target=integral\_third, args=(1,1000,))

t3.start()

print("\n --- Программа исполнена за %s секунд ---" % (time.time() - start\_time))

# END ---------------------------------------------------------------

Решение без использования потоков

import time

#Замер времени исполнения программы

start\_time = time.time()

#Вывод информации о программе

print("\tПрограмма считает три интеграла методом прямоугольников.")

print("\tВычисления производятся в одном потоке.\n")

# Запуск функций

integral\_first(-1,1)

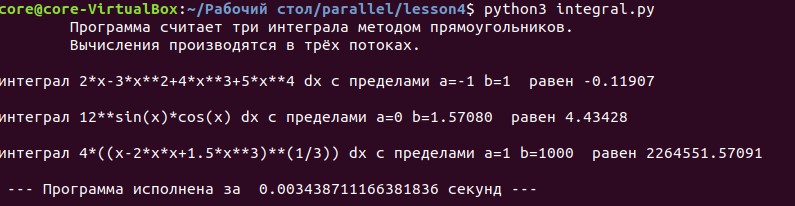
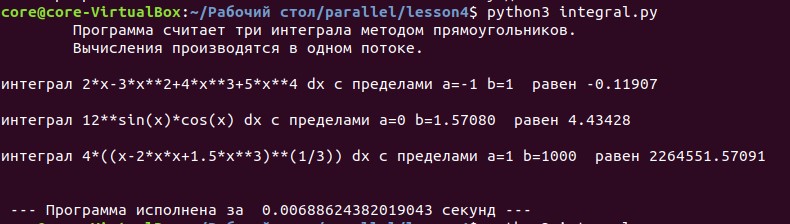
integral\_second(0,math.pi/2)

integral\_third(1,1000)

print("\n --- Программа исполнена за %s секунд ---" % (time.time() - start\_time))

# END ---------------------------------------------------------------

Результат работы



При использовании потоков время, затраченное на выполнение в 2 раза меньше.

# Решение задачи №2

Функции для вычисления суммы рядов

import math

def series\_first(n,x):

summa=0

for i in range(0,10000):

tmp=x\*\*i/math.factorial(i)

summa=summa+tmp

print('Сумма ряда x\*\*n/fact(n) при n от 0 до %d'%(n),

'x=%.2f равна %.5f\n' % (x,summa))

# END series\_first ------------------------------------------------

def series\_second(n):

summa=0

for i in range(1,10000):

tmp=i/2\*\*i

summa=summa+tmp

print('Сумма ряда n/(2\*\*n) при n от 1 до %d'%(n),' равна %.5f\n' % (summa))

# END series\_second ------------------------------------------------

def series\_third(n):

summa=0

for i in range(1,1000):

tmp=(i+6)\*\*(-1/3)

summa=summa+tmp

print('Сумма ряда (n+6)\*\*(-1/3) при n от 1 до %d'%(n),' равна %.5f\n' % (summa))

# END series\_third ------------------------------------------------

Решение с использование потоков

import threading

import time

#Вывод информации о программе

print("\tПрограмма суммы трёх рядов.")

print("\tВычисления производятся в трёх потоках.\n")

#Получение исходных данных

print("Введите Х ")

x=input()

x=int(x)

#Замер времени исполнения программы

start\_time = time.time()

# Запуск функций в отдельных потоках

t2 = threading.Thread(target=series\_second, args=(10000,))

t2.start()

t3 = threading.Thread(target=series\_third, args=(1000,))

t3.start()

series\_first(10000,x)

print("\n --- Программа исполнена за %s секунд ---" % (time.time() - start\_time))

# END ---------------------------------------------------------------

Решение без использования потоков

import time

#Получение исходных данных

print("Введите Х ")

x=input()

x=int(x)

#Замер времени исполнения программы

start\_time = time.time()

# Запуск функций

series\_first(10000,x)

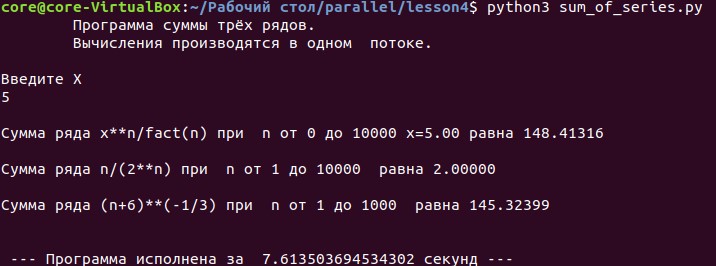
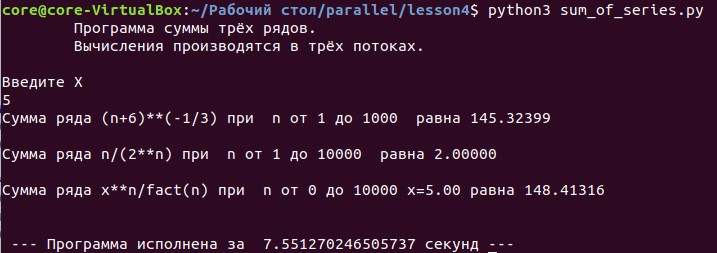
series\_second(10000)

series\_third(1000)

print("\n --- Программа исполнена за %s секунд ---" % (time.time() - start\_time))

# END ---------------------------------------------------------------

Результат работы



Использование потоков не дало существенного преимущества поскольку бо́льшую часть времени занимает подсчёт суммы ряда , две другие суммы вычисляются в разы быстрее и незначительно влияют на скорость выполнения программы.