Национальный исследовательский университет

«МЭИ»

Институт радиотехники и электроники  
Кафедра РТС

КУРСОВАЯ РАБОТА

Тема курсовой работы:

"Разработка модуля расчёта координат спутника GPS"

Часть 3

**Выполнил:**

Студент гр. ЭР-15-14

Мокшин А.С.

**Проверил:**  
 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Москва 2019

**Листинг программы:**

#include <iostream>

#include <fstream>

#include <gpssvpos.h>

#include <kepler.h>

#include <ctime>

using namespace std;

int **main**()

{

time\_t start, end;

double t = 396018;

double delt = 0.1;

double \*koord = new double[3];

double \*koord\_matlab = new double[3];

double max\_del = 0;

int i\_max = 0;

std::ofstream out;

out.open("Е:\\cpp.txt");

std::ifstream in("Е:\\mat.txt");

if (!in)

{

std::cout << "File not open!" << std::endl;

} else {

std::cout << "File open!" << std::endl;

}

time(&start);

for (int i = 0; i < (12\*3600/delt); i++)

{

gps\_coord(t,koord);

t += delt;

std::string koord\_str1 = std::to\_string(koord[0]);

std::string koord\_str2 = std::to\_string(koord[1]);

std::string koord\_str3 = std::to\_string(koord[2]);

out << koord\_str1 << " " << koord\_str2 << " " << koord\_str3 << std::endl;

in >> koord\_matlab[0] >> koord\_matlab[1] >> koord\_matlab[2];

for (int j = 0; j < 3; j++)

{

if (abs(koord[j]-koord\_matlab[j]) > max\_del)

{

max\_del = abs(koord[j]-koord\_matlab[j]);

i\_max = i;

}

}

}

time(&end);

in.close();

delete[] koord;

koord = nullptr;

delete[] koord\_matlab;

koord\_matlab = nullptr;

double seconds = difftime(end, start);

std::string seconds1 = std::to\_string(seconds\*1000000/(12\*3600/delt));

cout << "Srednee vremia raschota, mcs: " << seconds1 << std::endl;

std::string max\_del1 = std::to\_string(max\_del);

cout << "Maximalnaia raznost koordinat: " << max\_del1 << std::endl;

std::string imax = std::to\_string(i\_max);

cout << "Nomer otcheta s max raznostiy: " << imax << std::endl;

out.close();

in.close();

}

#include <gpssvpos.h>

#include <kepler.h>

#include <iostream>

#include <cmath>

#include <ostream>

using namespace std;

void **gps\_coord**(double t, double \*koord)

{

double mu = 3.986005E+14;

double we = 7.292115E-05;

double toe = .309618000000E+06;

double a\_sqr = sqrt(.26560593206E+06);

double e = .427016000000E-02;

double M0 = -.17927475E+03\*M\_PI/180;

double omega = -.2013571E+02\*M\_PI/180;

double i0 = .55581618E+02\*M\_PI/180;

double omega0 = -.12417322E+03\*M\_PI/180;

double del\_n = .24458E-08\*M\_PI/180;

double i\_dot = -.31719E-08\*M\_PI/180;

double omega\_dot = -.46211E-06\*M\_PI/180;

double cuc = -.20713E-05;

double cus = .61411E-05;

double crc = .26525E+03;

double crs = -.41188E+02;

double cic = .46566E-07;

double cis = .37253E-07;

double tk = t - toe;

double Mk = M0 + (sqrt(mu)/pow(a\_sqr,3) + del\_n)\*tk;

double Ek = kepler(Mk,e);

double Vk = atan2(sqrt(1-pow(e,2))\*sin(Ek),cos(Ek)-e);

double Uk = omega + Vk + cuc\*cos(2\*(omega + Vk)) + cus\*sin(2\*(omega + Vk));

double rk = a\_sqr\*a\_sqr\*(1-e\*cos(Ek)) + crc\*cos(2\*(omega+Vk))+crs\*sin(2\*(omega+Vk));

double ik = i0 + i\_dot\*tk + cic\*cos(2\*(omega + Vk)) + cis\*sin(2\*(omega + Vk));

double lambk = omega0 + (omega\_dot - we)\*tk - we\*toe;

koord[0] = (cos(-lambk)\*cos(-Uk)-sin(-lambk)\*cos(-ik)\*sin(-Uk))\*rk;

koord[1] = (-sin(-lambk)\*cos(-Uk)-cos(-lambk)\*cos(-ik)\*sin(-Uk))\*rk;

koord[2] = (-sin(-ik)\*sin(-Uk))\*rk;

}

#include <kepler.h>

#include <cmath>

double **kepler**(double Mk, double e){

double Ek = Mk;

double Ek1 = Ek;

do{

Ek1 = Ek;

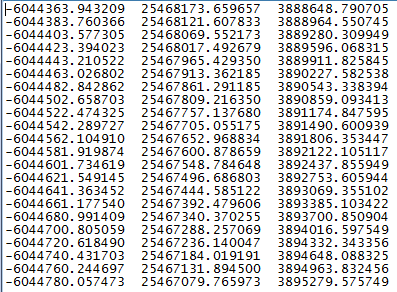
Ek = Mk + e\*sin(Ek);

}while(fabs(Ek1-Ek)/fabs(Ek) > 0.0001);

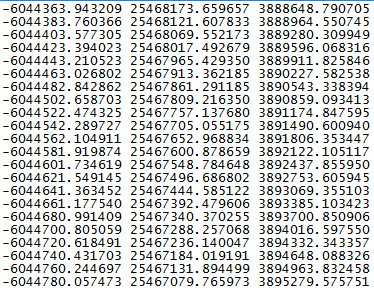
return Ek;

}

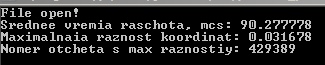
**Результаты программы на С++:**



**Результаты программы в Matlab:**

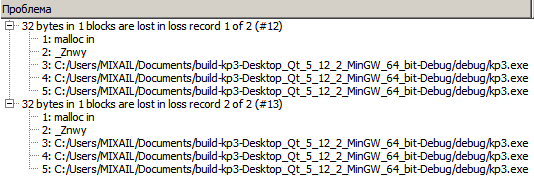


**Максимальная разница координат:**



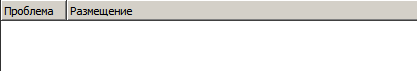
Видно из результатов, максимальное различие координат, полученных с помощью C++ и Matlab составляет 0.031678 м.

**Результаты теста при наличии ошибок памяти:**



Наблюдаются утечки памяти. Они вызваны из-за выделения памяти под хранение координат и не очисткой этой памяти после использования.

**Результаты теста без ошибок:**



Утечек памяти не обнаружено.

**Результаты теста на время выполнения:**

Замерять время производилась с помощью времени выполнения цикла и делением на количество итераций (оценка приближенная).

