

**НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ
«МЭИ»**

ИНСТИТУТ РАДИОТЕХНИКИ И ЭЛЕКТРОНИКИ

КАФЕДРА РАДИОТЕХНИЧЕСКИХ СИСТЕМ

КУРСОВАЯ РАБОТА

по дисциплине

**«АППАРАТУРА ПОТРЕБИТЕЛЕЙ СПУТНИКОВЫХ РАДИОНАВИГАЦИОННЫХ
СИСТЕМ»**

ФИО СТУДЕНТА: ЛЕВАШОВ А.В.

Группа: ЭР-15-16

ВАРИАНТ №: 11

Дата: _____

Подпись: _____

ФИО ПРЕПОДАВАТЕЛЯ: КОРОГОДИН И.В.

Оценка: _____

МОСКВА, 2021

ВВЕДЕНИЕ

На сегодняшний день трудно представить жизнь современного человека без спутниковых радионавигационных систем. Возможность определения собственных координат, а также точного времени сильно улучшила уровень жизни современных людей. Бумажные карты остались в прошлом, определить свое местоположение, необходимый маршрут и продолжительность времени в пути можно просто зайдя в мобильное приложение.

Но большинство пользователей совершенно не задумывается насколько в действительности сложные технические решения предприняты для того, чтобы предоставить настолько высокоточный сервис неограниченному числу пользователей. Когда-то о таком не могли мечтать даже военные, а ведь первые радионавигационные технологии были доступны только им.

Нам, как инженерам-радиотехникам, в отличие от пользователей интересен не только готовый продукт в виде координат и маршрута движения, но и то по средствам каких научных и технических изысканий решается навигационная задача.

Поэтому целью курсового проекта является укрепление знаний, полученных в курсах, посвящённых глобальным навигационным системам, исследование навигационной системы Beidou, ознакомление с рядом инструментов и техник, используемых при разработке навигационных приемников.

АННОТАЦИЯ

Цель проекта - добавление в программное обеспечение приемника функции расчета положения спутника Beidou на заданное время по данным его эфемерид.

Требования к разрабатываемому программному модулю:

- 1) требования назначения;
- 2) отсутствие утечек памяти;
- 3) малое время выполнения;
- 4) низкий расход памяти;
- 5) корректное выполнение при аномальных входных данных.

Для достижения цели выполняется ряд задач, соответствующих этапам проекта и контрольным мероприятиям:

- 1) обработка данных от приемника, работа со сторонними сервисами для подготовки входных и проверочных данных для разрабатываемого модуля;
- 2) моделирование модуля в Matlab/Python;
- 3) реализация программного модуля на C/C++, включая юнит-тестирование в Check.

Этапы курсовой работы отличаются осваиваемыми инструментами.

ЭТАП 1. ИСПОЛЬЗОВАНИЕ СТОРОННИХ СРЕДСТВ

1.1 Общие сведения о системе Beidou

BeiDou Navigation Satellite System является самостоятельно реализуемой и развиваемой Китайской Народной Республикой навигационной спутниковой системой. Целью создания системы являются: создание независимой, открытой, высокотехнологичной, стабильной и надежной Глобальной навигационной спутниковой системы. Кроме того, создание BDS призвано ускорить развитие космической отрасли Китая и промышленности в целом. Вследствие чего Китай уделяет большое внимание развитию BDS и, в частности, обладанию правами на интеллектуальную собственность касательно всех элементов системы. В соответствии с вышесказанным сформулированы основные принципы построения BDS:

- 1) Открытость. Строительство и развитие BDS ведётся открыто, система должна предоставлять высокое качество бесплатных услуг пользователям по всему миру.
- 2) Автономность. BDS должна обеспечивать работу по всему миру в независимость от других навигационных систем.
- 3) Совместимость. BDS должна быть пригодна для совместного использования с другими ГНСС.
- 4) Прогрессивность. В процессе строительства и эксплуатации системы должно вестись постоянное планомерное повышение качества и внедрение новейших технологий.

Состав орбитальной группировки космической навигационной системы Бэйдоу на 10 марта 2020 года:

- 1) Всего в составе ОГ : 48 КА
- 2) Используются по целевому назначению: 43 КА

3) Не используется по целевому назначению: 5 КА

Орбитальная группировка системы BeiDou Navigation Satellite System представлена на рисунке 1:

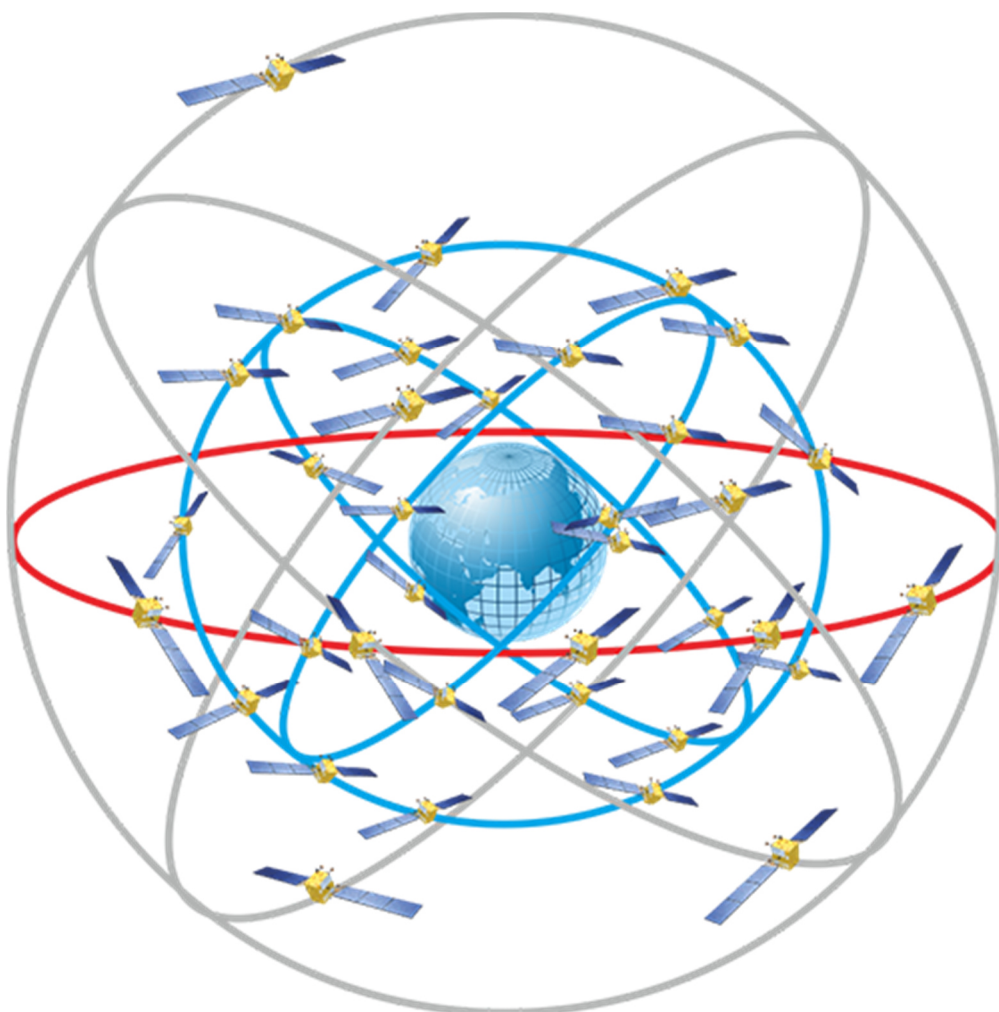


Рисунок 1 – орбитальная группировка системы BeiDou Navigation Satellite System

Приведем параметры эфемерид системы в таблице 1.

Таблица 1 – Параметры эфемерид системы

Параметр	Описание
t_{oe}	Опорная эпоха эфемерид
\sqrt{A}	Корень из большой полуоси орбиты
e	Эксцентриситет орбиты
ω	Аргумент перигея
Δn	Поправка в среднее движение
M_0	Средняя аномалия на опорную эпоху
Ω_0	Долгота восходящего угла орбиты на опорную эпоху
Ω	Скорость прямого восхождения
i_0	Угол наклона орбиты на опорную эпоху
IDOT	Скорость изменения наклона орбиты
Cuc	Амплитуда косинусной поправки к аргументу широты
Cus	Амплитуда синусной поправки к аргументу широты
Crc	Амплитуда косинусной поправки к радиусу орбиты
Crs	Амплитуда синусной поправки к радиусу орбиты
Cic	Амплитуда косинусной поправки к углу наклона
Cis	Амплитуда синусной поправки к углу наклона

1.2 Использование входных данных и определение номера спутника

На крыше корпуса Е МЭИ установлена трехдиапазонная антенна Harxon HX-CSX601A. Она через 50-метровый кабель, сплиттер, bias-tee и усилитель подключена к трем навигационным приемникам:

- 1) Javad Lexon LGDD,
- 2) SwiftNavigation Piksi Multi,
- 3) Clonicus разработки ЛНС МЭИ.

Эти приемники осуществляют первичную обработку сигналов Beidou B1I, выдавая по интерфейсам соответствующие потоки данных - наблюдения псевдодальностей и эфемериды спутников. Данные от приемника Clonicus, записанные вечером 16 февраля 2021 года, доступны в рабочем репозитории (директория logs) в нескольких форматах.

Воспользуемся этими данными и сведем их в таблицу для конкретного варианта. Данные сведены в таблицу 2.

Таблица 2 – Значения эфемерид спутника С11

Параметр	Обозначение параметра	Значение
Satnum	PRN	11
toe (мс)	t_{oe}	237600000.000
Crs (рад)	-	-1.4750000000000000e+01
Dn (рад/мс)	Δn	3.08584299606173840e-12
M0 (рад)	M_0	2.05369950341053498e+00
Cuc (рад)	-	-8.32602381706237793e-07
e	e	2.12729186750948429e-03
Cus (рад)	-	1.11386179924011230e-05
sqrtA ($m^{1/2}$)	\sqrt{A}	5.28260794258117676e+03
Cic (рад)	-	3.35276126861572266e-08
Omega0 (рад)	Ω_0	-2.34989438813352569e+00
Cis (рад)	-	-2.60770320892333984e-08
i0 (рад)	i_0	9.90627871535710081e-01
Crc (рад)	-	1.4590625000000000e+02
Omega (рад)	ω	-1.91916766236221403e+00
OmegaDot (рад/мс)	$\dot{\Omega}$	-6.45062583731548538e-12
iDot (рад/сек)	i_{DOT}	-1.25719522437021839e-13
Tgd (мс)	T_{gd}	4.1000000000000000e+04
Toc (мс)	T_{oc}	2.3760000000000000e+08
af2 (мс/мс ²)	-	2.16840444796468865e-22
af1 (мс/мс)	-	2.26592078433895949e-11
af0 (мс)	-	-9.19481515884399414e-01
URA	-	0
IODE	-	2827
IODC	-	10
codeL2	-	0
L2P	-	0
WN	-	789

Проверку корректности этих данных можно осуществить сравнив их с данными на сайте (https://www.glonass-iac.ru/BEIDOU/beidou_almanac.php). Приведем скриншот таблицы эфемерид с сайта glonass-iac (рисунок 2).

PRN	H	e	t	δ_1	Ω	A	Ω_0	ω	m	af_0	af_1	week
C01	000	0.00071239	430080	0.00743022	1.38291475e-9	6493.3501	2.98905894	-1.58469541	1.06331646	-0.0007095337	3.63798e-11	789
C02	000	0.00096846	430080	0.02975084	9.25752847e-10	6493.3682	-1.02751434	-1.63411872	-2.20927314	0.0005016327	-2.91038e-11	789
C03	000	0.00091696	430080	0.03526358	1.21147903e-9	6493.5327	-1.29044860	-0.02036682	-3.09766448	-0.0004491806	2.54659e-11	789
C04	000	0.00042629	430080	0.01647232	1.46291808e-9	6493.4438	-1.55592091	-2.85548951	0.86799906	-0.0004148483	-8.00355e-11	789
C05	000	0.00081635	602112	0.02306963	1.54292141e-9	6493.5288	-1.38334432	-1.17875413	-2.77005621	-0.0007972717	-7.63976e-11	789
C06	000	0.01062536	430080	0.00345146	-2.01151236e-9	6492.7988	0.63641665	-2.20170060	-2.74867860	0.0001039505	3.27418e-11	789
C07	000	0.00814486	430080	-0.05126851	-2.18294807e-9	6492.9468	2.63927761	-2.52518816	1.88863220	-0.0003395081	-3.63798e-11	789
C08	000	0.00533342	430080	0.09738381	-1.88579284e-9	6493.1177	-1.48581469	-2.68672953	-0.08744552	-0.0009117126	7.27596e-12	789
C09	000	0.00773954	430080	0.00841293	-1.96579617e-9	6492.9893	0.67875279	-2.38421885	-2.99440952	0.0007238388	-2.54659e-11	789
C10	000	0.00687933	430080	-0.04953679	-2.16008998e-9	6493.2471	2.63247394	-2.55366680	1.54080693	-0.00008392334	7.27596e-12	789
C11	000	0.00211430	417792	0.04812865	-6.75456707e-9	5282.5640	-2.35106374	-1.91359160	1.31983804	-0.0009155273	2.18279e-11	789

Рисунок 2 – Таблицы эфемерид с сайта glonass-iac

Как видно, что данные предоставленные преподавателем действительно сходятся со спутником C11. Но сложность заключается в том что PRN не отражает номера спутника, который используется в различных сторонних средствах. Номер спутника можно определить с помощью таблицы приведенной на Википедии (рисунок 3).

№	Спутник	PRN	Дата (UTC)	Ракета	NSSDC ID	SCN	Орбита	Статус	Система
12	Компас М3	C11	29.04.2012 20:50	CZ-3B/E	2012-018A	38250	СОО, ~21 500 км	действующий	

Рисунок 3 – Таблица спутников BeiDou на Википедии

Из предоставленной таблицы видно, что спутнику с PRN C11 соответствует спутник №12, это обстоятельство стоит учитывать при выполнении следующих пунктов этапа.

1.3 Определение формы орбиты и положения спутника на ней на начало рассматриваемого интервала времени по данным сервиса CelesTrak

Для выполнения этого пункта перейдем на сайт CelesTrak (<https://celestrak.com>). Настроим параметры и выберем необходимый спутник, после чего будет построена Земля и орбита спутника вокруг нее (рисунок 4).

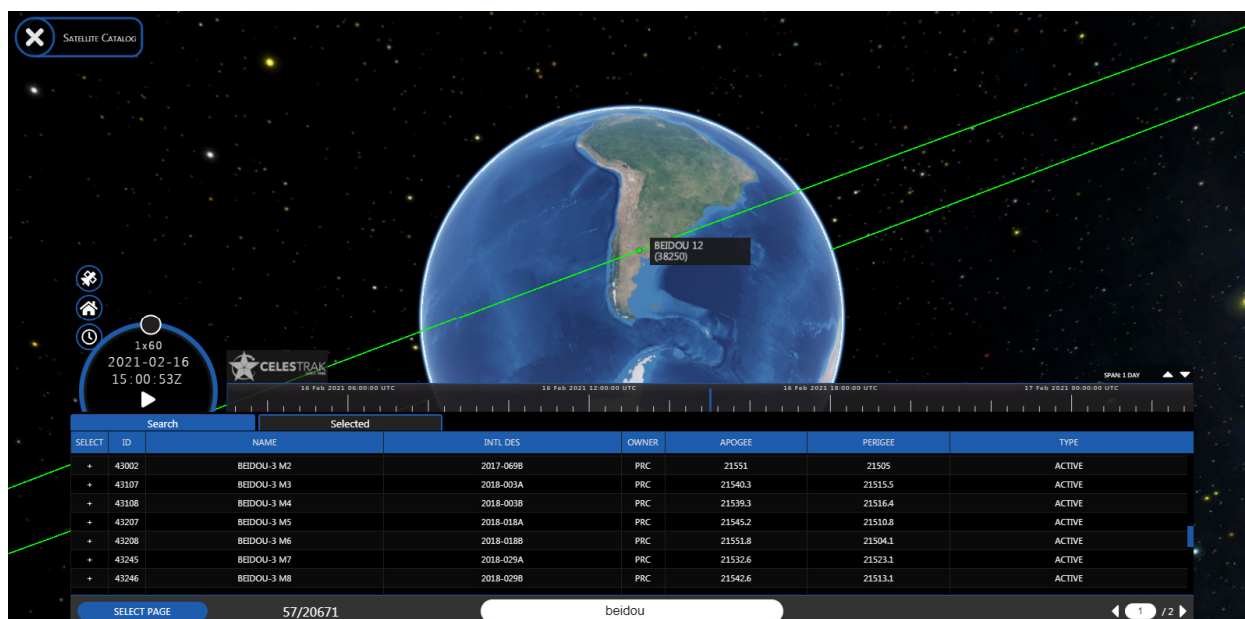


Рисунок 4 – Результат моделирования на CelesTrak

Учтем, что в задании просится построить модель на момент 18:00 по МСК. Это говорит о том, что на сайте нужно время установить 15:00. На данный момент времени спутник еще не находится в зоне видимости антенны на корпусе Е, что сходится с данными, полученными при использовании сервиса Trimble GNSS Planning.

1.4 Расчет графика угла места собственного спутника от времени по данным Trimble GNSS Planning Online

Нужно построить график угла места от времени и SkyView собственного спутника на заданный интервал времени. Для этого воспользуемся веб-сайтом Trimble GNSS Planning (<https://www.gnssplanning.com>). Во вкладке настроек (Settings) указываем координаты корпуса «Е» МЭИ и время проведения записи (Рисунок 5). Во вкладке библиотеки спутников (Satellite Library) отключаем отображение всех спутников, кроме заданного (Рисунок 6).

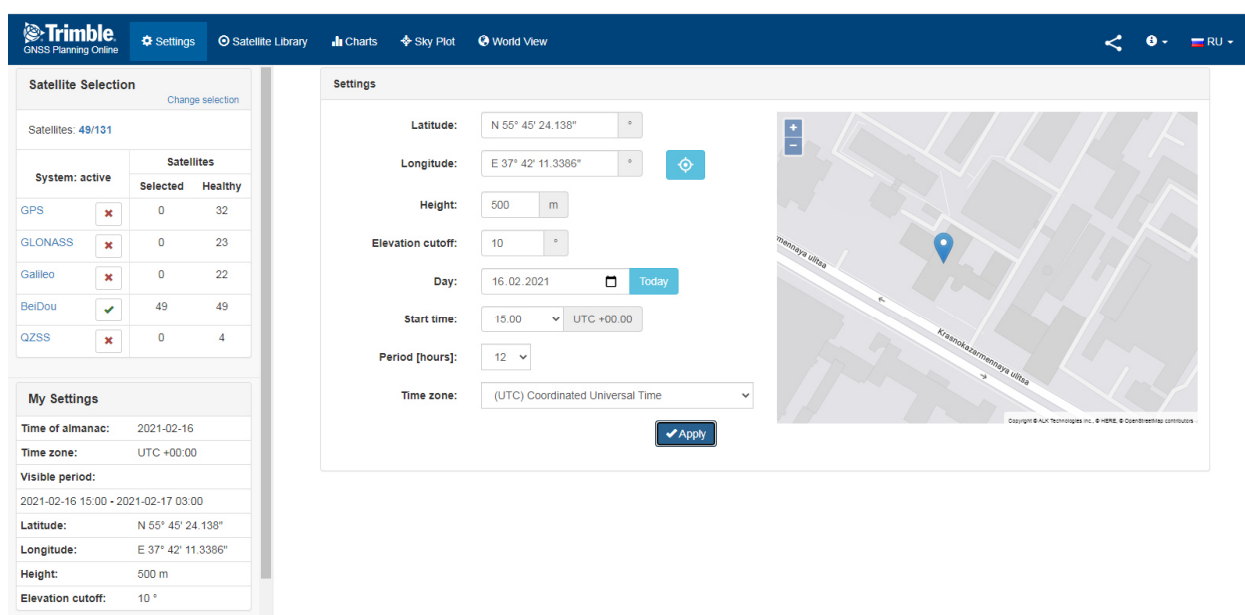


Рисунок 5 – Вкладка настроек (Settings) Trimble GNSS Planning

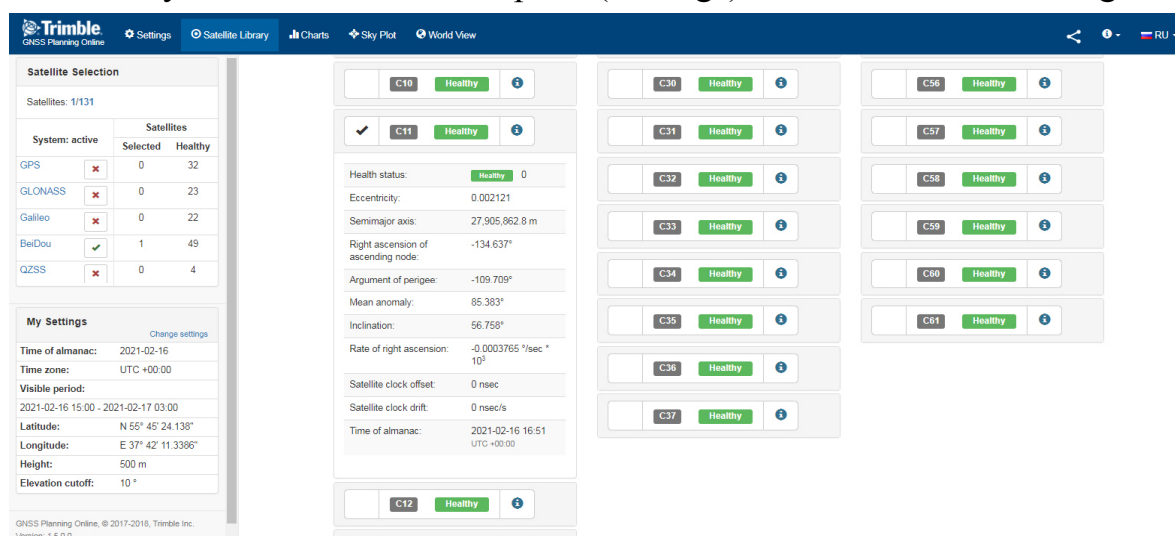


Рисунок 6 – Вкладка библиотека спутников (Satellite Library) Trimble GNSS Planning

Чтобы получить график угла места, нажимаем вкладку графики (Charts). По полученным данным, спутник был виден 1 раз (Рисунок 7). Появление наблюдается с 18:40 до 00:30. Время указано по UTC +00:00.

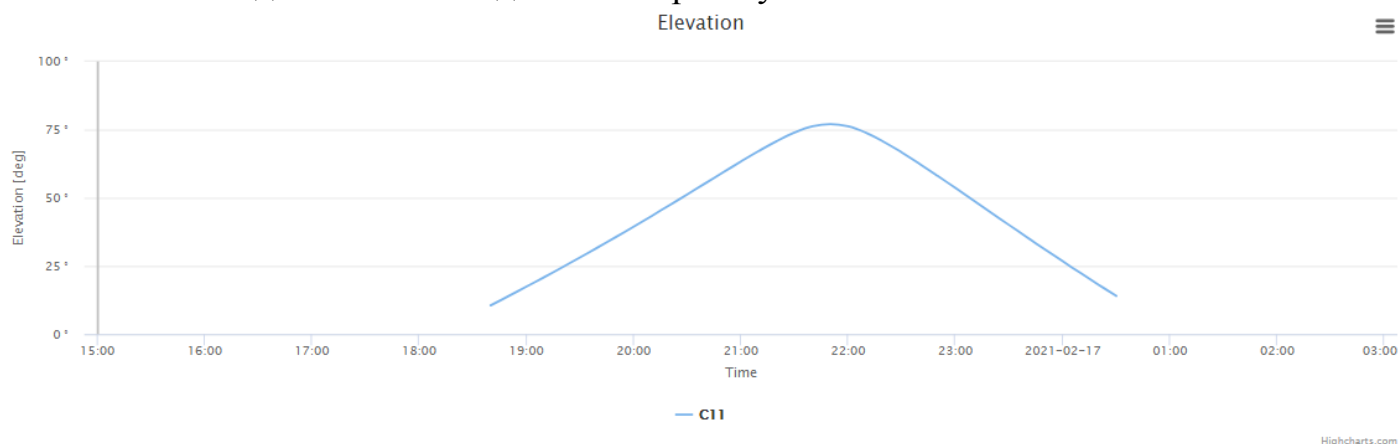


Рисунок 7 – График угла места спутника Beidou C11

Во вкладке «Sky Plot» можно получить карту небосвода (SkyView) (рисунок 8).

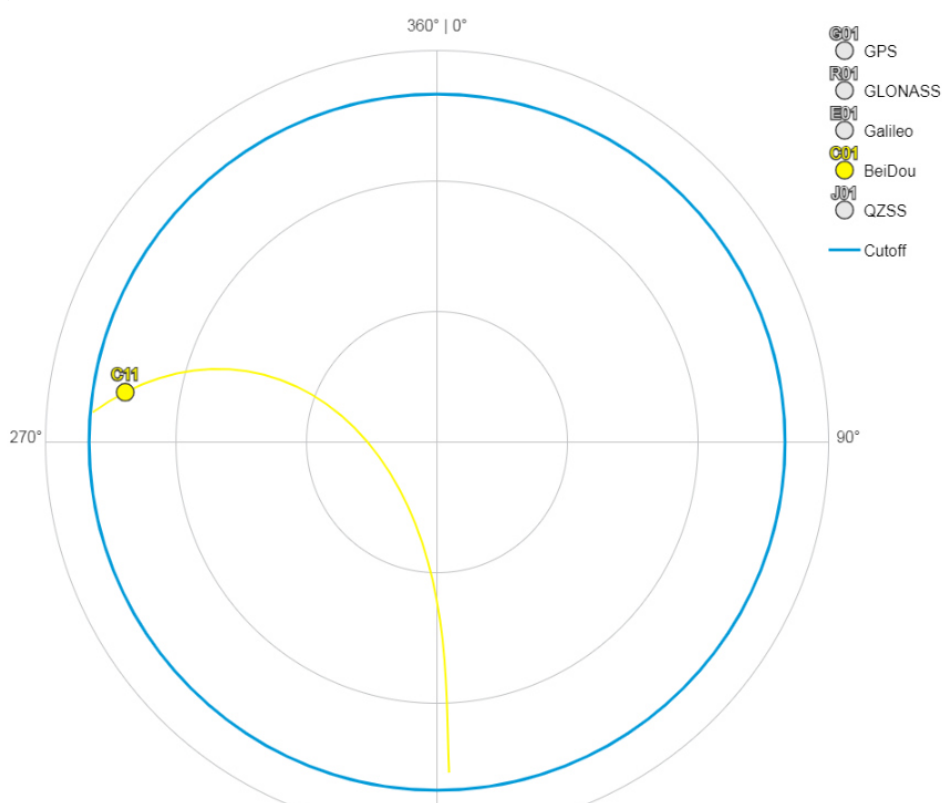


Рисунок 8 – SkyView спутника Beidou C11

Траектория отражает единственное появление спутника в заданном промежутке времени.

ЭТАП 2. МОДЕЛИРОВАНИЕ

Задание на этап:

Требуется реализовать на языке Matlab или Python функцию расчета положения спутника Beidou на заданный момент по шкале времени UTC. В качестве эфемерид использовать данные, полученные на предыдущем этапе.

Построить трехмерные графики множества положений спутника Beidou с системным номером, соответствующим номеру студента по списку. Графики в двух вариантах: в СК ECEF WGS84 и соответствующей ей инерциальной СК. Положения должны соответствовать временному интервалу с 18:00 МСК 16 февраля до 06:00 МСК 17 февраля 2021 года. Допускается использовать одни и те же эфемериды на весь рассматриваемый интервал.

Построить SkyView за указанный временной интервал (напоминаю, антенна на крыше корпуса Е) и сравнить результат с Trimble GNSS Planning Online, полученный на прошлом этапе.

Моделирование.

Моделирование проводилось с помощью алгоритма приведенного на сайте Навипедия. Для выполнения данного этапа был использован пакет математического моделирования Matlab.

2.1 Результаты моделирования

Алгоритм расчёта реализован в программе MATLAB R2015a, код приведён в приложении 1.

Рассчитанные траектории движения спутника приведены на рисунке 9.

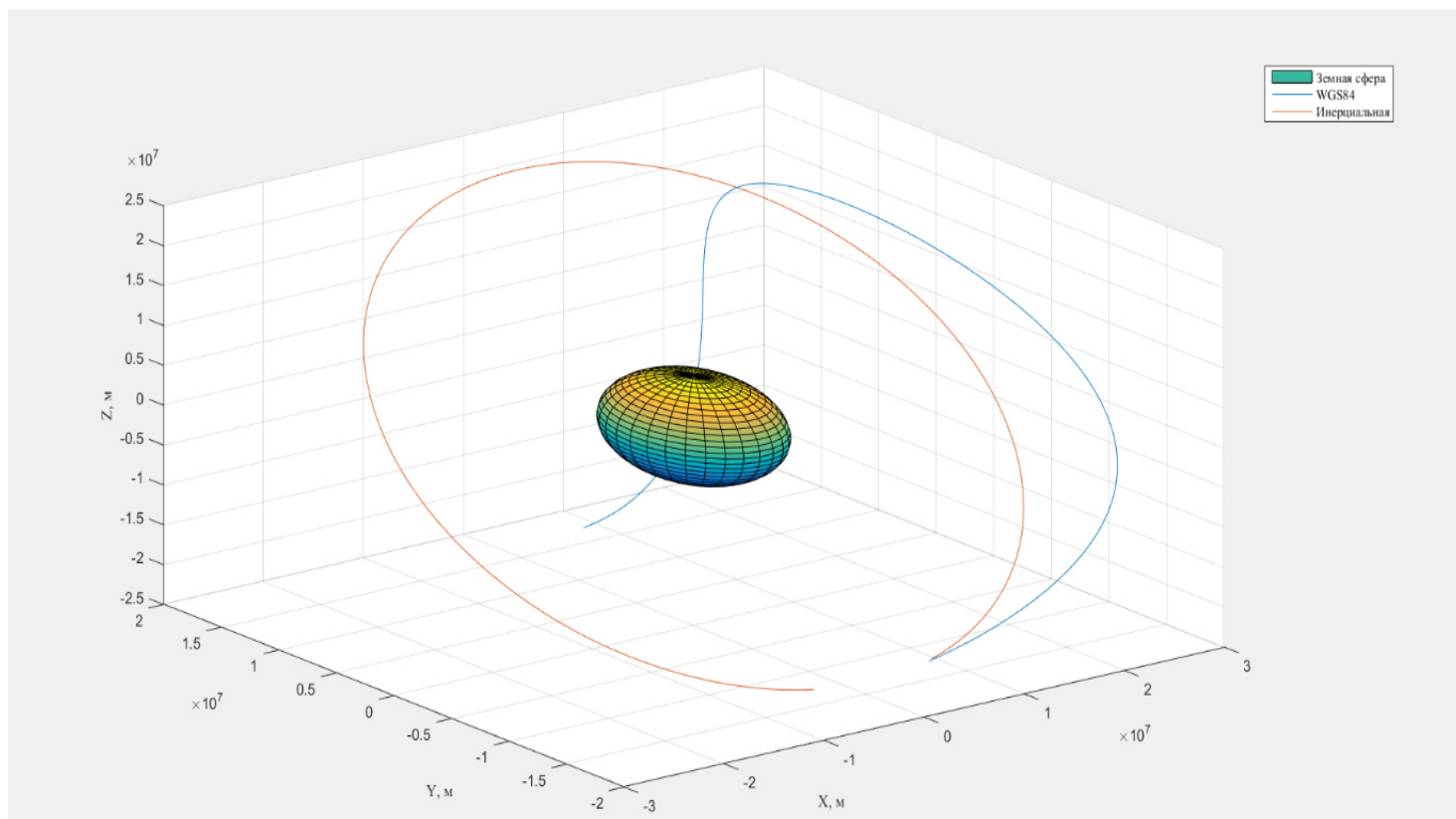


Рисунок 9 – Траектория движения спутника Beidou PRN №11 в системе координат WGS-84 (синяя линия) и инерциальной системе координат (красная линия)

Необходимо построить SkyView за указанный временной интервал и сравнить результат с Trimble GNSS Planning Online, полученный на прошлом этапе.

Для этого необходимо декартовы координаты пересчитать в сферические, а затем по полученным углам построить графики в полярной системе координат (рисунок 10) и угла места в зависимости от времени (рисунок 11). Полученные результаты соответствуют, полученным на Trimble GNSS Planning.

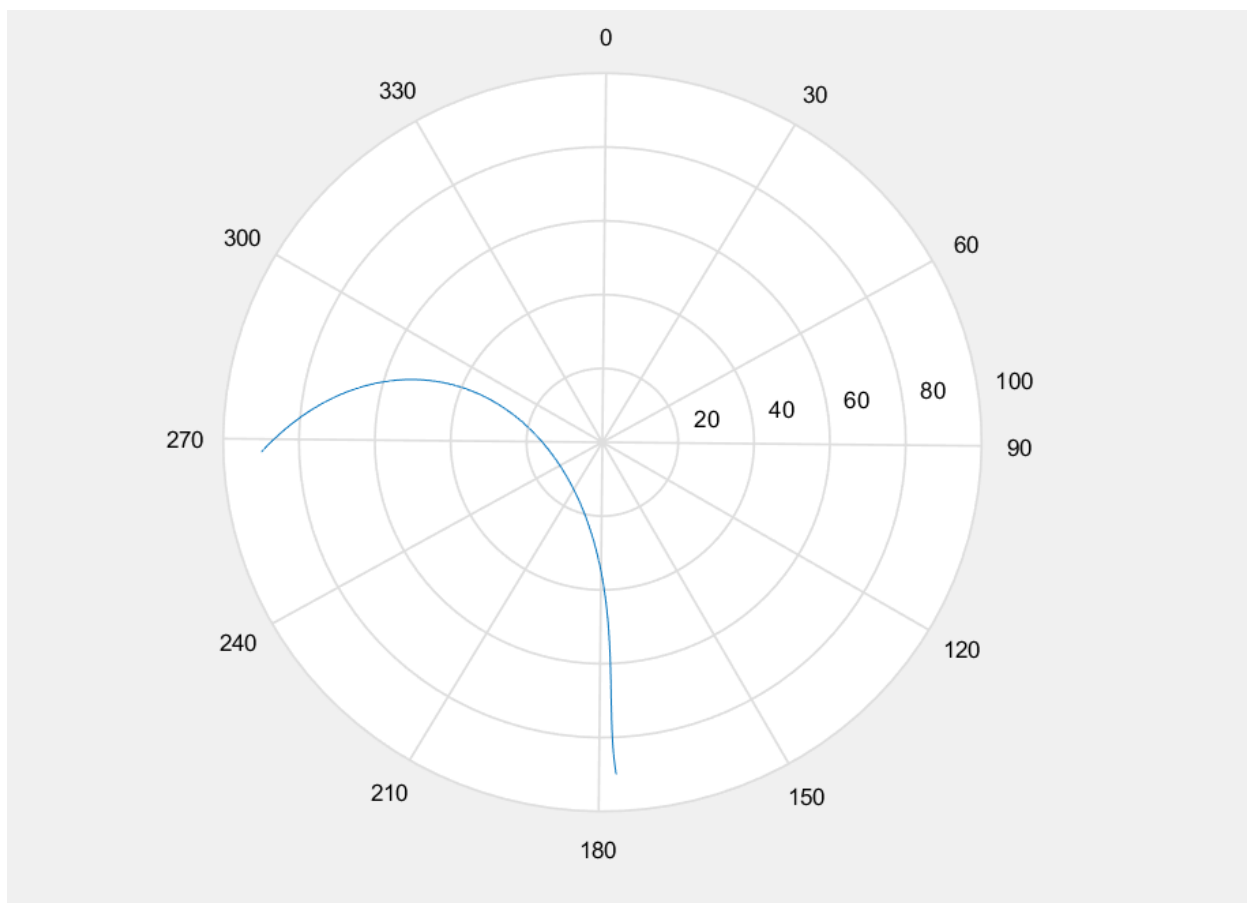


Рисунок 10 – SkyView спутника Beidou PRN №11

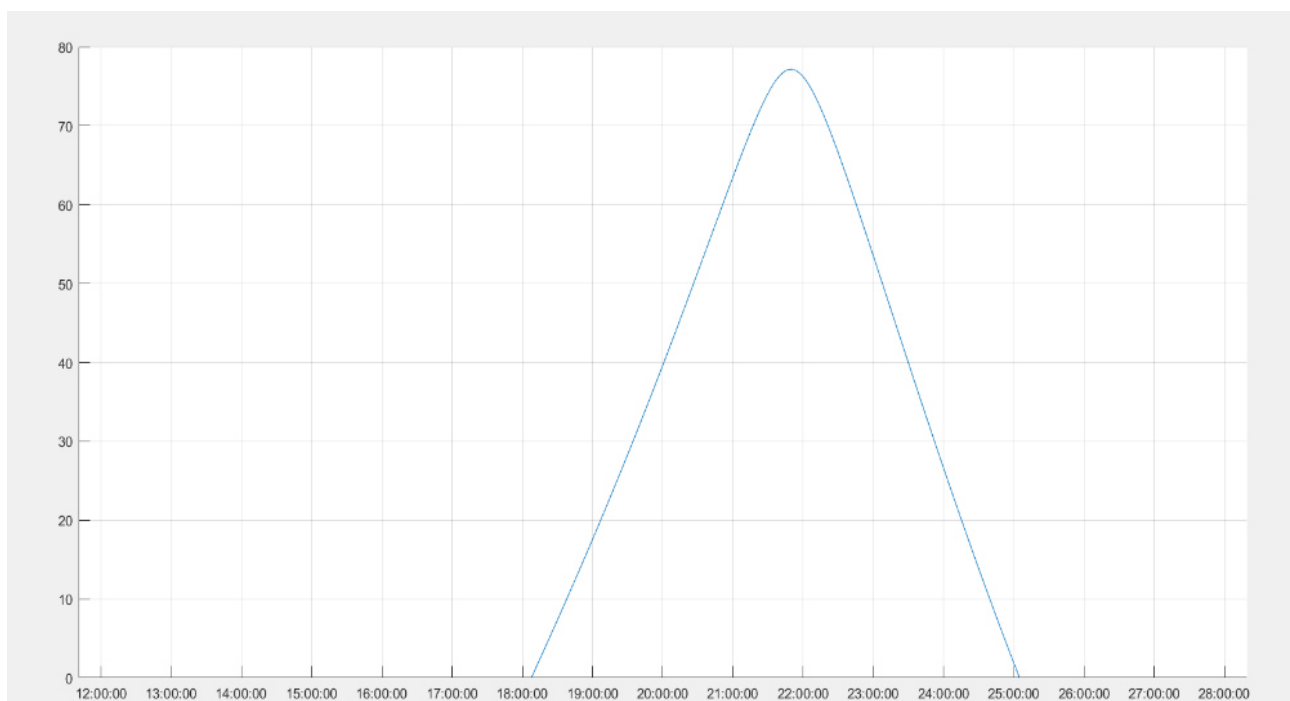


Рисунок 11 – График угла места спутника Beidou PRN №11

ЭТАП 3. РЕАЛИЗАЦИЯ

Требуется разработать на языке C/C++ функцию расчета положения спутника Beidou на заданное время по шкале UTC, минимизируя время её исполнения и количество затрачиваемой оперативной памяти. Вызов функции не должен приводить к выбросу исключений или утечкам памяти при любом наборе входных данных.

Функция расчета положения спутника в Matlab/Python относительно проста, т.к. доступны библиотеки линейной алгебры и решения уравнений. Но при разработке встраиваемого ПО приходится сохранять лицензионную частоту, минимизировать вычислительную нагрузку и затраты памяти. Поэтому отобразить модель из Matlab/Python в прошивку приемника дословно, как правило, не получается. В рассматриваемом примере потребуется, как минимум, выполнить свою реализацию решения трансцендентного уравнения.

Реализацию программы будем производить в Microsoft Visual Studio 2019, используя при этом язык программирования C++.

Иерархия проекта будет следующей:

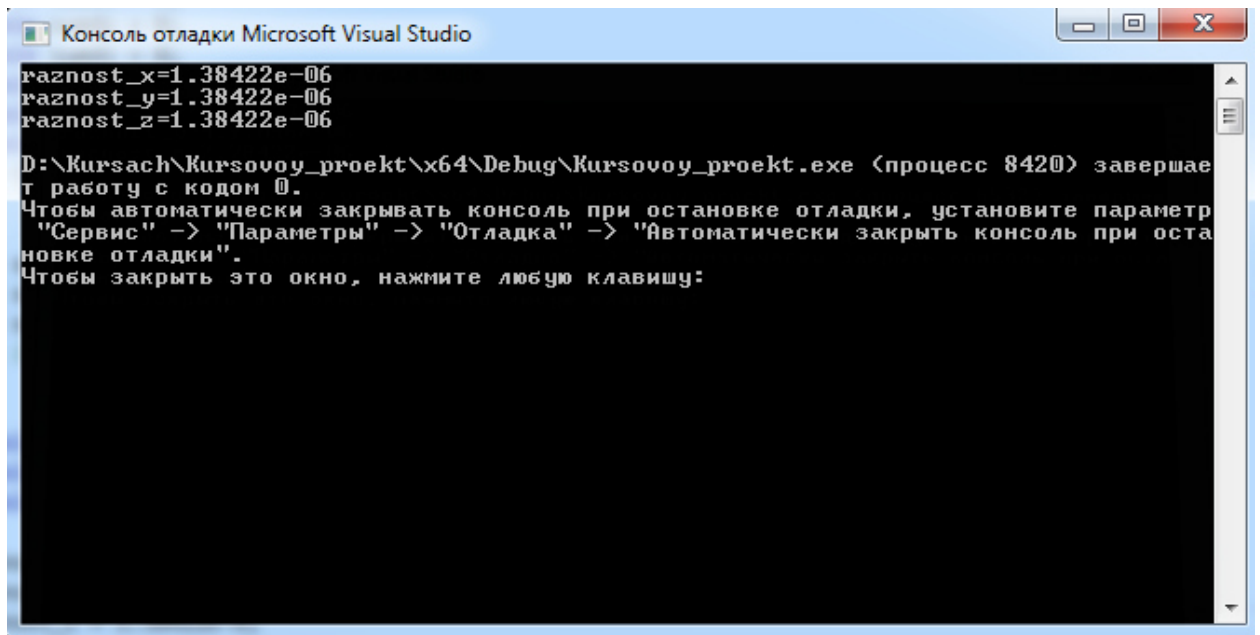
1) заглавным выступает файл `Kursovoy_proekt.cpp`, в нем будут задаваться исходные данные, текстовый файл с рассчитанными координатами НКА и расчет разности этих координат с рассчитанными на втором этапе. Теоретические ожидания таковы, что расхождения между расчетами будут минимальны и стремиться к нулю;

2) далее будут созданы два заголовочных файла, таких как `Kepler.h` (для функции Кеплера) и `Rasch_coord.h` (для функции расчета координат в каждый такт времени);

3) создание файла расчета координат в каждый такт времени `Rasch_coord.cpp`;

4) создание файла расчета функции Кеплера Kepler.cpp

Результат запуска программы приведен на рисунке 12:



```
Консоль отладки Microsoft Visual Studio
raznost_x=1.38422e-06
raznost_y=1.38422e-06
raznost_z=1.38422e-06
D:\Kursach\Kursovoy_proekt\x64\Debug\Kursovoy_proekt.exe (процесс 8420) завершае
т работу с кодом 0.
Чтобы автоматически закрывать консоль при остановке отладки, установите параметр
"Сервис" -> "Параметры" -> "Отладка" -> "Автоматически закрыть консоль при оста
новке отладки".
Чтобы закрыть это окно, нажмите любую клавишу:
```

Рисунок 12 – Результат запуска программы Kursovoy_proekt

Стоит заметить, что теоретические ожидания подтвердились и средняя разница в расчете C++ и Matlab очень мала и равна $1.38422 \cdot 10^{-6}$. Вероятно разница ненулевая из-за представления значений в файле в формате long double, что позволяет получить большую разрядность чем представление числа в Matlab.

После этого можно переходить к анализу написанного кода с помощью встроенных инструментов Microsoft Visual Studio 2019. Перейдя во вкладку Анализ запустим профилировщик производительности.

В общем случае профилирование – это сбор характеристик программы во время ее выполнения. При профилировании замеряется время выполнения и количество вызовов отдельных функций и строк в коде программы.

Изначально запустим такой инструмент, как инструментирование, который даст подробный отчет о загрузке ЦП во время работы программы, а также о количестве вызове, используемых функций. График загрузки ЦП представлен на рисунке 13.

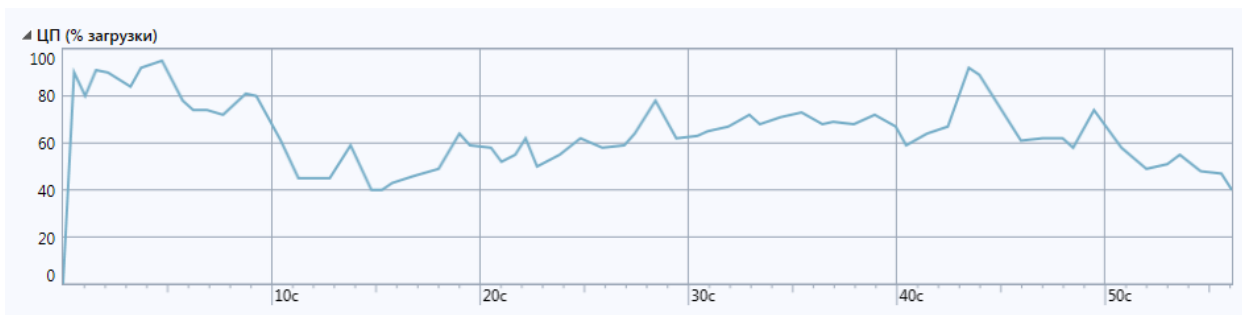


Рисунок 13 – График загрузки ЦП

Информация о критическом пути и о функциях, выполняющих максимальную индивидуальную работу приведена на рисунке 14.

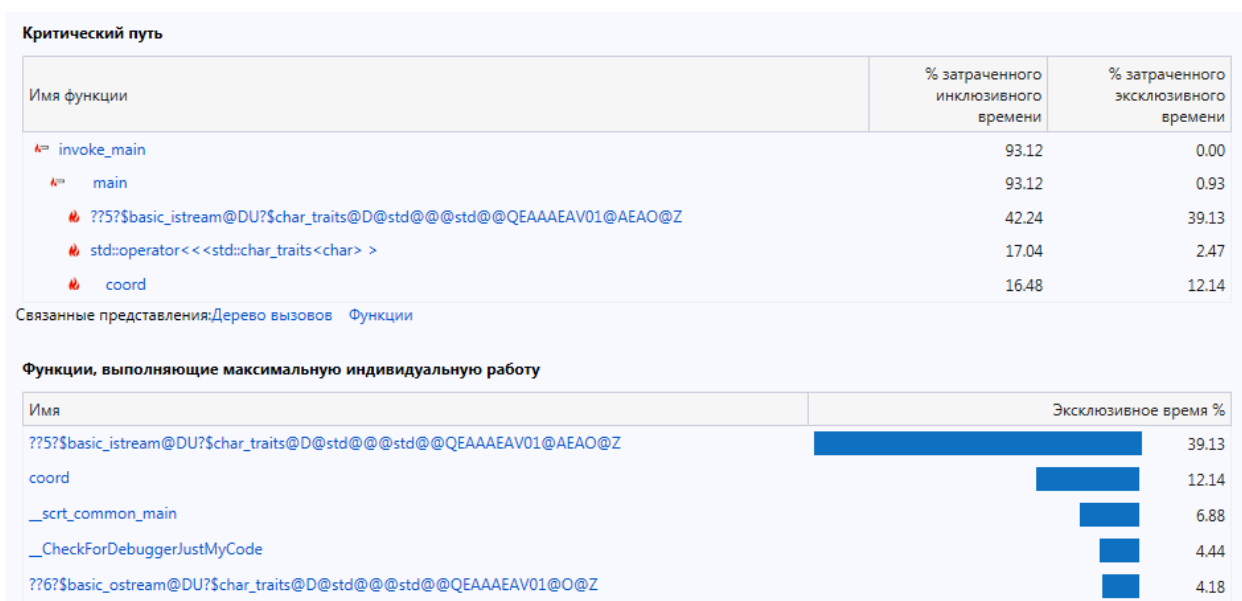


Рисунок 14 – Результаты инструментирования

Результаты подробного профилирования будут приведены в приложении.

Следующим важным инструментом, которым стоит воспользоваться – это использование памяти, чтобы определить такую проблему как утечки памяти.

В общем случае под утечкой памяти (memory leak) понимается – неконтролируемое уменьшение свободной оперативной или виртуальной памяти компьютера. Причиной утечек становятся ошибки в программном коде.

В итоге был получен график использования памяти при проведении сеанса диагностики. Зависимость представлена на рисунке 15.

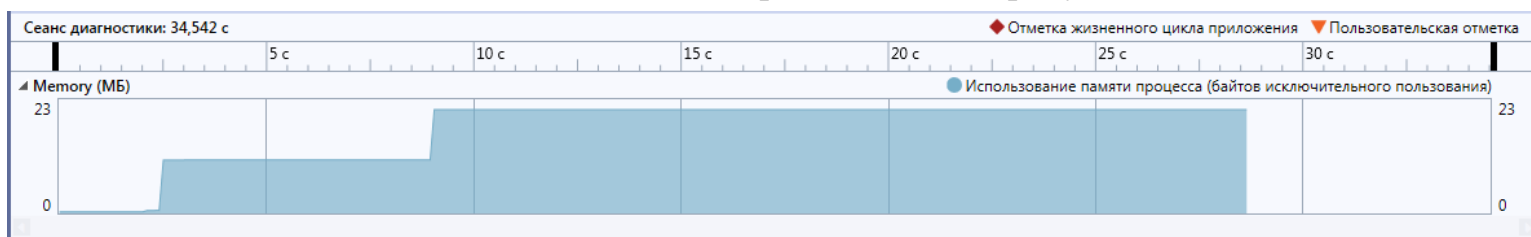


Рисунок 15 – Использование памяти при проведении сеанса диагностики

Также предоставим отчет из отладки на рисунке 16.

```

Выход
Показать выходные данные из: Отладка
"Kursovoy_proekt.exe" (Win32). Загружено "D:\Kursach\Kursovoy_proekt\x64\Debug\Kursovoy_proekt.exe". Символы загружены.
"Kursovoy_proekt.exe" (Win32). Загружено "C:\Windows\System32\ntdll.dll".
"Kursovoy_proekt.exe" (Win32). Загружено "C:\Windows\System32\kernel32.dll".
"Kursovoy_proekt.exe" (Win32). Загружено "C:\Windows\System32\KernelBase.dll".
"Kursovoy_proekt.exe" (Win32). Загружено "C:\Windows\System32\msvcrt140d.dll".
"Kursovoy_proekt.exe" (Win32). Загружено "C:\Windows\System32\user32.dll".
"Kursovoy_proekt.exe" (Win32). Загружено "C:\Windows\System32\api-ms-win-core-localization-l1-2-0.dll".
"Kursovoy_proekt.exe" (Win32). Загружено "C:\Windows\System32\api-ms-win-core-processthreads-l1-1-1.dll".
"Kursovoy_proekt.exe" (Win32). Загружено "C:\Windows\System32\api-ms-win-core-file-l1-2-0.dll".
"Kursovoy_proekt.exe" (Win32). Загружено "C:\Windows\System32\api-ms-win-core-timezone-l1-1-0.dll".
"Kursovoy_proekt.exe" (Win32). Загружено "C:\Windows\System32\api-ms-win-core-file-l2-1-0.dll".
"Kursovoy_proekt.exe" (Win32). Загружено "C:\Windows\System32\api-ms-win-core-synch-l1-2-0.dll".
Поток 0x29a4 завершился с кодом 0 (0x0).
Программа "[8420] Kursovoy_proekt.exe" завершилась с кодом 0 (0x0).

```

Рисунок 16 – Отчет отладки

В ходе выполнения данного этапа была написана программа на языке C++ реализующая алгоритм из этапа 2. Средняя разница в полученных значениях получилась пренебрежительно малой, что говорит о корректности, написанной программы. Также код был тщательно проанализирован с помощью инструментов Microsoft Visual Studio 2019 на предмет загрузки ЦП и утечек памяти. Последних обнаружено не было, что указывает на отсутствие серьезных ошибок в программе.

ЗАКЛЮЧЕНИЕ

В учебном плане радиотехнического факультета присутствует ряд дисциплин, которые посвящены радионавигационным системам. Эти дисциплины рассматривают их с различных сторон, таких как: общие сведения о навигации и радионавигации, техническая реализация НКА, формирование радионавигационных сигналов, алгоритмы оптимального приема сигналов, реализация приемных устройств, алгоритмы получения координат потребителя и т.д. Само собой эти дисциплины ведутся в различные периоды времени и разными преподавателями, что часто приводит к разрозненности знаний в умах студентов. Но данная курсовая работа позволяет собрать, полученные знания воедино и использовать их на практике.

Курсовой проект состоит из 3-х этапов, причем выполнение каждого является довольно трудоемкой задачей, которая требует приложения солидных усилий. Но такая работа позволяет еще больше расширить свое представление о ГНСС.

Первый этап стал ознакомительным с исследуемой системой Beidou. Пришлось исследовать ИКД этой системы, работать с альманахом и сторонними средствами, для получения некоторых референсных значений с которыми можно будет сравнивать индивидуальные расчеты.

Второй этап предполагал работу с уже знакомой средой математического моделирования Matlab, необходимо было реализовать алгоритм, который могу бы построить орбиту спутника и SkyView, и график угла места, заданного НКА. Результаты, полученные на этом этапе, совпали с референсными из предыдущего этапа. Этот факт устанавливает верность реализованного алгоритма.

Третий этап стал самым трудоемким, ввиду того, что для его выполнения пришлось изучать стороннюю литературу, посвященную языку C++ и программе Microsoft Visual Studio 2019. Но по преодолению этих

трудностей была написана программа, состоящая из нескольких файлов, которую можно считать непосредственной реализацией второго этапа для переноса в «железо». Анализ кода показал, что он является работоспособным и не обладающим утечками памяти. Значения рассчитанные в C++ практически не отличаются от того, что было рассчитано в Matlab. А то небольшое различие, которое все же присутствует вызвано исключительно типом данным C++ и Matlab.

Подводя общий итог можно сказать, что данный проект помог отточить не только радиотехнические навыки, но и навыки программирования и работы в среде Matlab и Microsoft Visual Studio, и даже знание английского языка, ведь вся техническая литература, описывающая систему Beidou, приведена исключительно на английском.

ПРИЛОЖЕНИЕ

Приложение 1.

```
close all;
clear all;
clc;
format long

%% Введем исходные данные
SatNum = 11;
Toe = 237600.000000;
Crs = -1.4750000000000000e+01;
Dn = 3.08584299606173840e-12;
M0 = 2.05369950341053498e+00;
Cuc = -8.32602381706237793e-07;
e = 2.12729186750948429e-03;
Cus = 1.11386179924011230e-05;
sqrtA = 5.28260794258117676e+03;
A = sqrtA^2;
Cic = 3.35276126861572266e-08;
Omega0 = -2.34989438813352569e+00;
Cis = -2.60770320892333984e-08;
i0 = 9.90627871535710081e-01;
Crc = 1.45906250000000000e+02;
omega = -1.91916766236221403e+00;
OmegaDot = -6.45062583731548538e-12;
iDot = -1.25719522437021839e-13;
Tgd = 4.1000000000000000e+04;
Toc = 2.3760000000000000e+08;
af2 = 2.16840444796468865e-22;
af1 = 2.26592078433895949e-11;
af0 = -9.19481515884399414e-01;
URA = 0;
IODE = 2827;
IODC = 10;
codeL2 = 0;
L2P = 0;
WN = 789;
mu = 3.98600442e14;
Om_Earth = 7.2921151467e-5;
R_Earth = 6371e3;

% время на начало расчета 18:00 МСК 16 февраля
t_begin = (24*2 + 15)*3600;
% время на момент окончания расчета 6:00 МСК 17 февраля
t_finish = (24*3 + 3)*3600;

%Длительность расчета
Time = t_begin:1:t_finish;

% Среднее движение КА
n = sqrt(mu/A^3) + Dn;
```

```

%Рассчитаем координаты КА
for k = 1:length(Time)
    T(k) = Time(k) - Toe;
    %Исходя из предоставленного алгоритма
    if T(k) > 302400
        T(k) = T(k) - 604800;
    end
    if T(k) < -302400
        T(k) = T(k) + 604800;
    end
    % Средняя аномалия
    M(k) = M0 + n*T(k);
    % Уравнение Кеплера
    E(k) = M(k);
    Epre(k) = M(k)+1;
    Ostanova = 1e-7;
    while abs(E(k) - Epre(k)) > Ostanova
        Epre(k) = E(k);
        E(k) = M(k) + e*sin(Epre(k));
    end
    % Истинная аномалия
    v(k) = atan2( sqrt(1 - e^2) * sin(E(k)) , cos(E(k)) - e );
    % Коэффициенты коррекции
    Correction_cos(k) = cos(2*(omega + v(k)));
    Correction_sin(k) = sin(2*(omega + v(k)));
    % Аргумент широты
    u(k) = omega + v(k) + Cuc*Correction_cos(k) + Cus*Correction_sin(k);
    % Радиус орбиты
    r(k) = A * (1 - e * cos(E(k))) + Crc*Correction_cos(k) + Crs*Correction_sin(k);
    % Наклон орбиты КА
    i(k) = i0 + iDot * T(k) + Cic*Correction_cos(k) + Cis*Correction_sin(k);
    % Долгота восходящего угла
    E(k) = Omega0 + (OmegaDot - Om_Earth) * T(k) - Om_Earth*Toe;
    % Положение на орбите
    x = r(k) * cos(u(k));
    y = r(k) * sin(u(k));
    % Координаты
    Xcord(k) = x * cos(E(k)) - y * cos(i(k)) * sin(E(k));
    Ycord(k) = x * sin(E(k)) + y * cos(i(k)) * cos(E(k));
    Zcord(k) = y * sin(i(k));
    % Координаты в инерциальной
    X(k) = Xcord(k) * cos(E(k)) + Ycord(k) * sin(E(k));
    Y(k) = -Xcord(k) * sin(E(k)) + Ycord(k) * cos(E(k));
    Z(k) = Zcord(k);
end

%% Пересчет координат в WGS-84
ppb = 1e-9;
mas = 1e-3/206264.8; % [рад]

WGS_84 = [-3*ppb -353*mas -4*mas;353*mas -3*ppb 19*mas;4*mas -19*mas -3*ppb];
CordWGS_84 = [Xcord; Ycord; Zcord];

for i = 1:length(CordWGS_84(1,:))
    CordWGS_84(:,i) = CordWGS_84(:,i) + WGS_84 * CordWGS_84(:,i) + [0.07; -0; -0.77];
end

CordWGS_84 = CordWGS_84.';

```

```

% Построение графических зависимостей
[XE,YE,ZE] = sphere(30);
figure (1)
surf(XE*R_Earth,YE*R_Earth,ZE*R_Earth)
hold on
grid on
plot3(CordWGS_84(:,1), CordWGS_84(:,2), CordWGS_84(:,3))
plot3(X, Y, Z)
xlabel('X, м', 'FontName', 'Times New Roman')
ylabel('Y, м', 'FontName', 'Times New Roman')
zlabel('Z, м', 'FontName', 'Times New Roman')
hold off
lgd = legend('Земная сфера','WGS84','Инерциальная СК');
lgd.FontName = 'Times New Roman';

%% Географические координаты корпуса E
% Широта
N_gr = 55;
N_min = 45;
N_sec = 23.8178;
N = N_gr*pi/180 + N_min/3437.747 + N_sec/206264.8;

% Долгота
E_gr = 37;
E_min = 42;
E_sec = 12.2608;
E = E_gr*pi/180 + E_min/3437.747 + E_sec/206264.8;

H = 500; % высота [м]

a = 6378137.0000;
b = 6356752.3142;
e = sqrt(1-(b/a).^2);
x = (a*cos(E))/(sqrt(1 + (1 - e*e)*(tan(N))^2)) + H*cos(E)*cos(N);
y = (a*cos(N))/(sqrt(1 + (1 - e*e)*(tan(N))^2)) + H*cos(N)*cos(N);
z = (a*(1 - e*e)*sin(N))/(sqrt(1 - e*e*sin(N)*sin(N))) + H*sin(N);

Coordinates_E(1) = x;
Coordinates_E(2) = y;
Coordinates_E(3) = z;

%% SkyPlot
for i = 1:length(CordWGS_84(:,1))

    [X(i) Y(i) Z(i)] = ecef2enu(CordWGS_84(i,1),CordWGS_84(i,2), ...
    ... CordWGS_84(i,3),N,E,H,wgs84Ellipsoid,'radians');
    if Z(i) > 0
        r(i) = sqrt(X(i)^2 + Y(i)^2 + Z(i)^2);
        theta(i) = acos(Z(i)/r(i));
        if X(i) > 0
            N(i) = -atan(Y(i)/X(i))+pi/2;
        elseif (X(i)<0)&&(Y(i)>0)
            N(i) = -atan(Y(i)/X(i))+3*pi/2;
        elseif (X(i)<0)&&(Y(i)<0)
            N(i) = -atan(Y(i)/X(i))-pi/2;
        end
    else theta(i) = NaN;
        r(i) = NaN;
        N(i) = NaN;
    end
end
end

```



```

% SkyPlot
figure (1)
polar(N,theta*180/pi)
ax.ThetaDir = 'clockwise';
ax.ThetaZeroLocation = 'top';

% Angle
T_hours = hours(Time/3600 - 2*24); % Перевод временной оси в формат hh:mm:ss
figure (2)
grid on
hold on
plot(T_hours, (-theta)*180/pi+90, 'DurationTickFormat', 'hh:mm:ss')

```

ПРИЛОЖЕНИЕ 2.

Kursovoy_proekt.cpp

```

#include <iostream>
#include <fstream>
#include <cmath>
#include "Kepler.h"
#include "Rasch_coord.h"
using namespace std;
int main()
{
    long double e, mu, Omega_e, toe, A, n0, i0, Omega0, omega, M0, Dn, n, OmegaDot,
    IDOT, Crs, Cuc, Cus, Cic, Cis, Crc;
    long double t;
    e = 2.12729186750948429e-03;
    mu = 3.98600442E+14;
    Omega_e = 7.2921151467E-5;
    toe = 237600;
    A = pow(5.282607942E+3, 2);
    n0 = pow(mu / (pow(A, 3)), 0.5);
    i0 = 0.99062787153;
    Omega0 = -2.34989438813352569;
    omega = -1.91916766236221403;
    M0 = 2.05369950341053498;
    Dn = 3.08584299606173840E-9;
    n = n0 + Dn;
    OmegaDot = -6.45062583731548538E-12;
    IDOT = -1.25719522437021839E-13;
    Crs = -14.75000;
    Cuc = -8.32602381706237793E-7;
    Cus = 5.74858859E-6;
    Cic = 3.35276126861572266E-8;
    Cis = -2.60770320892333984E-8;
    Crc = 1.4590625000000000E+02;

    long double t_begin = (24 * 2 + 15) * 3600;
    long double t_finish = (24 * 3 + 3) * 3600;
    int N_stop = 432000;
    long double *x = new long double[N_stop];
    long double *y = new long double[N_stop];
    long double *z = new long double[N_stop];
    long double xx, yy, zz;

```

```

int k = 0;
for (long double i = t_begin; i < t_finish; i+=0.1)
{
    coord(e, mu, Omega_e, toe,
          A, n0, i0, Omega0, omega, M0,
          Dn, n, OmegaDot, IDOT, Crs, Cuc, Cus,
          Cic, Cis, Crc, i, &xx, &yy, &zz);
    x[k] = xx;
    y[k] = yy;
    z[k] = zz;
    k++;
}

ofstream f;
f.open("Znach.txt");
k = 0;
for (int i = t_begin; i < t_finish; i += 1)
{
    f << "k = " << k << "\t" << "t ="<< i << "\t" << x[k] << "\t" << y[k] << "\t" <<
    z[k]<< endl;
    k++;
}
f.close();

long double* x_e2 = new long double[N_stop];
long double* y_e2 = new long double[N_stop];
long double* z_e2 = new long double[N_stop];
int i = 0;

ifstream fin("cord.txt");
if (!fin.is_open())
    cout << "Файл не открылся!" << endl;
else
{
    while (fin >> x_e2[i] >> y_e2[i] >> z_e2[i])
    {
        i++;
    }
    fin.close();
}

double sumdx = 0;
double sumdy = 0;
double sumdz = 0;
double dxmax = 0;
double dymax = 0;
double dzmax = 0;
for (int i = 0; i < N_stop; i++)
{
    sumdx += (fabs(x[i] - x_e2[i]));
    sumdy += (fabs(y[i] - y_e2[i]));
    sumdz += (fabs(z[i] - z_e2[i]));
}
double raznost_x = sumdx / N_stop;
double raznost_y = sumdy / N_stop;
double raznost_z = sumdz / N_stop;

cout << "raznost_x=" << raznost_x << endl;

```

```

cout << "raznost_y=" << raznost_y << endl;
cout << "raznost_z=" << raznost_z << endl;

delete[] x;
delete[] y;
delete[] z;
return 0;
}

```

// Kepler.h

```

#ifndef Kepler_H
#define Kepler_H

long double kepler(long double e, long double Mk);

#endif

```

//Rasch_coord.h

```

#ifndef Rasch_coord_H
#define Rasch_coord_H

void coord(long double e, long double mu, long double Omega_e, long double toe,
           long double A, long double n0, long double i0, long double Omega0, long double
omega, long double M0,
           long double Dn, long double n, long double OmegaDot, long double IDOT, long double
Crs, long double Cuc, long double Cus,
           long double Cic, long double Cis, long double Crc, long double t, long double* x,
           long double* y, long double* z);

#endif

```

//Rasch_coord.cpp

```

#include <math.h>
#include "Rasch_coord.h"
#include "Kepler.h"
void coord(long double e, long double mu, long double Omega_e, long double toe,
           long double A, long double n0, long double i0, long double Omega0, long double
omega, long double M0,
           long double Dn, long double n, long double OmegaDot, long double IDOT, long double
Crs, long double Cuc, long double Cus,
           long double Cic, long double Cis, long double Crc, long double t, long double *x,
           long double *y, long double *z)
{
    long double tk, Mk, E_k, nu_k, Phi_k, delta_u_k, delta_r_k, delta_i_k, u_k, r_k,
i_k;
    long double x_k, y_k, Omega_k, x_k1, y_k1, z_k1;
    tk = t - toe;
    if (tk > 302400)
    {
        tk = 604800 - tk;
    }
    else if (tk < -302400)
    {
        tk = 604800 + tk;
    }
}

```

```

    }
    Mk = M0 + n * tk;
    E_k = kepler(e, Mk);
    nu_k = atan2((sqrt(1 - pow(e, 2)) * sin(E_k)) / (1 - e * cos(E_k)), (cos(E_k) - e) / (1 - e * cos(E_k)));
    Phi_k = nu_k + omega;
    delta_u_k = Cus * sin(2 * Phi_k) + Cuc * cos(2 * Phi_k);
    delta_r_k = Crs * sin(2 * Phi_k) + Crc * cos(2 * Phi_k);
    delta_i_k = Cis * sin(2 * Phi_k) + Cic * cos(2 * Phi_k);
    u_k = Phi_k + delta_u_k;
    r_k = A * (1 - e * cos(E_k)) + delta_r_k;
    i_k = i0 + delta_i_k + IDOT * tk;
    x_k = r_k * cos(u_k);
    y_k = r_k * sin(u_k);
    Omega_k = Omega0 + (OmegaDot - Omega_e) * tk - Omega_e * toe;
    // earth-fixed
    x_k1 = x_k * cos(Omega_k) - y_k * cos(i_k) * sin(Omega_k);
    y_k1 = x_k * sin(Omega_k) + y_k * cos(i_k) * cos(Omega_k);
    z_k1 = y_k * sin(i_k);
    *x = x_k1;
    *y = y_k1;
    *z = z_k1;
}

```

// Kepler.cpp

```

#include <math.h>
#include "Kepler.h"
long double kepler(long double e, long double Mk)
{
    long double E[] = { 0, 0, 0, 0 };
    E[0] = Mk;
    for (int j = 1; j < 4; j++)
    {
        E[j] = E[j - 1] + (Mk - E[j - 1] + e * sin(E[j - 1])) / (1 - e * cos(E[j - 1]));
        E[j - 1] = E[j];
    }
    return E[3];
}

```

ПРИЛОЖЕНИЕ 3.

Имя функции	Число вызовов	% затраченного инклюзии...	% затраченного эксклю...	Среднее затраченное ин...	Среднее затраченное экс...	Имя модуля
??5?\$basic_istream@...	1 296 003	42,24%	39,13%	39 550,00	36 636,00	MSVCP140D.dll
coord	432 000	16,48%	12,14%	46 290,00	34 089,00	Kursovoy_proekt.exe
_scrt_common_main	1	100,00%	6,88%	121 347 933 805,00	8 346 391 277,00	Kursovoy_proekt.exe
_CheckForDebugger...	10 054 741	4,44%	4,44%	535,00	535,00	Kursovoy_proekt.exe
??6?\$basic_ostream@...	129 600	4,36%	4,18%	40 864,00	39 112,00	MSVCP140D.dll
fabs	1 296 000	7,71%	4,13%	7 223,00	3 864,00	Kursovoy_proekt.exe
fabsl	1 296 000	3,06%	2,93%	2 866,00	2 739,00	Kursovoy_proekt.exe
kepler	432 000	4,28%	2,86%	12 022,00	8 029,00	Kursovoy_proekt.exe
std::basic_ostream<c...	259 203	4,22%	2,80%	19 759,00	13 089,00	Kursovoy_proekt.exe
std::operator<<<std::...	259 203	17,04%	2,47%	79 785,00	11 569,00	Kursovoy_proekt.exe
std::basic_filebuf<cha...	1 857 603	2,45%	1,76%	1 602,00	1 151,00	Kursovoy_proekt.exe
std::basic_ostream<c...	259 203	6,80%	1,71%	31 851,00	8 008,00	Kursovoy_proekt.exe
??6?\$basic_ostream@...	86 400	1,77%	1,65%	24 790,00	23 129,00	MSVCP140D.dll
std::basic_filebuf<cha...	1 857 603	2,26%	1,09%	1 478,00	714,00	Kursovoy_proekt.exe
fgetc	8 476	0,97%	0,97%	139 150,00	139 150,00	ucrtbased.dll
std::basic_ostream<c...	259 203	3,46%	0,97%	16 209,00	4 527,00	Kursovoy_proekt.exe
_unlock_file	1 857 603	0,96%	0,96%	629,00	629,00	ucrtbased.dll
main	1	93,12%	0,93%	113 000 890 120,00	1 129 933 731,00	Kursovoy_proekt.exe
?rdbuf@?\$basic_ios@...	1 296 015	0,93%	0,93%	871,00	871,00	MSVCP140D.dll
fflush	43 200	0,89%	0,89%	25 079,00	25 079,00	ucrtbased.dll
?flags@ios_base@std...	259 203	0,75%	0,75%	3 497,00	3 497,00	MSVCP140D.dll
?_Osfx@?\$basic_ostre...	259 203	0,57%	0,57%	2 691,00	2 691,00	MSVCP140D.dll
std::char_traits<char...	216 000	0,58%	0,56%	3 266,00	3 151,00	Kursovoy_proekt.exe
_security_check_cookie	319 367	0,56%	0,56%	2 130,00	2 130,00	Kursovoy_proekt.exe
_RTC_CheckStackVars	751 366	0,52%	0,52%	836,00	836,00	Kursovoy_proekt.exe
_lock_file	1 857 603	0,51%	0,51%	335,00	335,00	ucrtbased.dll
std::basic_ostream<c...	259 203	1,89%	0,49%	8 859,00	2 314,00	Kursovoy_proekt.exe
std::basic_filebuf<cha...	259 200	1,26%	0,36%	5 880,00	1 694,00	Kursovoy_proekt.exe
Имя функции	Число вызовов	% затраченного инклюзии...	% затраченного эксклю...	Среднее затраченное ин...	Среднее затраченное экс...	Имя модуля
std::basic_ostream<c...	259 203	1,03%	0,43%	6 639,00	2 514,00	Kursovoy_proekt.exe
std::basic_filebuf<cha...	259 200	1,26%	0,36%	5 880,00	1 694,00	Kursovoy_proekt.exe
?good@ios_base@std...	518 406	0,30%	0,30%	697,00	697,00	MSVCP140D.dll
fwrite	43 200	0,14%	0,14%	3 859,00	3 859,00	ucrtbased.dll
std::char_traits<char...	259 203	1,76%	0,14%	8 230,00	634,00	Kursovoy_proekt.exe
?sputn@?\$basic_strea...	259 203	1,36%	0,10%	6 371,00	490,00	MSVCP140D.dll
??Bios_base@std@...	432 001	0,09%	0,09%	266,00	266,00	MSVCP140D.dll
std::char_traits<char...	241 427	0,11%	0,08%	529,00	418,00	Kursovoy_proekt.exe
?widen@?\$basic_ios...	43 203	0,08%	0,08%	2 272,00	2 272,00	MSVCP140D.dll
std::basic_ostream<c...	259 203	0,10%	0,08%	471,00	371,00	Kursovoy_proekt.exe
?flush@?\$basic_ostre...	43 203	1,80%	0,07%	50 578,00	1 966,00	MSVCP140D.dll
?setstate@?\$basic_ios...	259 203	0,06%	0,06%	303,00	303,00	MSVCP140D.dll
?width@ios_base@st...	259 203	0,06%	0,06%	269,00	269,00	MSVCP140D.dll
?_Pnavail@?\$basic_str...	259 200	0,06%	0,06%	259,00	259,00	MSVCP140D.dll
?width@ios_base@st...	259 203	0,05%	0,05%	251,00	251,00	MSVCP140D.dll
?tie@?\$basic_ios@DU...	259 203	0,05%	0,05%	250,00	250,00	MSVCP140D.dll
?put@?\$basic_ostrea...	43 203	0,52%	0,05%	14 721,00	1 399,00	MSVCP140D.dll
?pptr@?\$basic_strea...	216 000	0,05%	0,05%	273,00	273,00	MSVCP140D.dll
?pbump@?\$basic_str...	216 000	0,04%	0,04%	252,00	252,00	MSVCP140D.dll
std::basic_filebuf<cha...	43 200	1,66%	0,04%	46 614,00	1 191,00	Kursovoy_proekt.exe
std::char_traits<char...	111 826	0,05%	0,04%	559,00	422,00	Kursovoy_proekt.exe
std::basic_filebuf<cha...	43 200	0,67%	0,04%	18 693,00	1 081,00	Kursovoy_proekt.exe
std::char_traits<char...	51 675	0,08%	0,03%	1 788,00	673,00	Kursovoy_proekt.exe
operator new	6	0,03%	0,03%	5 131 662,00	5 131 662,00	Kursovoy_proekt.exe
??6?\$basic_ostream@...	43 203	2,45%	0,02%	68 913,00	685,00	MSVCP140D.dll
std::basic_filebuf<cha...	8 476	1,04%	0,02%	148 504,00	3 409,00	Kursovoy_proekt.exe
std::basic_filebuf<cha...	8 476	1,14%	0,02%	163 627,00	3 066,00	Kursovoy_proekt.exe
?gptr@?\$basic_strea...	59 327	0,02%	0,02%	407,00	407,00	MSVCP140D.dll

std::endl<char,std::ch...	43 203	2,43%	0,02%	68 228,00	552,00	Kursovoy_proekt.exe
std::basic_filebuf<cha...	8 475	0,06%	0,02%	8 579,00	2 594,00	Kursovoy_proekt.exe
std::_Fgetc<char>	8 476	0,98%	0,01%	140 324,00	1 042,00	Kursovoy_proekt.exe
std::char_traits<char...	16 950	0,01%	0,01%	610,00	452,00	Kursovoy_proekt.exe
std::basic_filebuf<cha...	8 476	0,01%	0,01%	1 308,00	798,00	Kursovoy_proekt.exe
?egptr@?\$basic_strea...	16 950	0,01%	0,01%	393,00	393,00	MSVCP140D.dll
?_Gndec@?\$basic_str...	8 475	0,00%	0,00%	702,00	702,00	MSVCP140D.dll
?eback@?\$basic_strea...	16 951	0,00%	0,00%	351,00	351,00	MSVCP140D.dll
operator delete	3	0,00%	0,00%	1 896 472,00	1 896 472,00	Kursovoy_proekt.exe
??6?\$basic_ostream@...	3	0,00%	0,00%	1 102 405,00	1 102 405,00	MSVCP140D.dll
std::basic_filebuf<cha...	2	0,00%	0,00%	1 175 496,00	975 918,00	Kursovoy_proekt.exe
fclose	2	0,00%	0,00%	359 952,00	359 952,00	ucrtbased.dll
?_Getcat@?\$codecvt...	1	0,00%	0,00%	176 934,00	176 934,00	MSVCP140D.dll
pre_cpp_initialization	1	0,00%	0,00%	81 236,00	81 236,00	Kursovoy_proekt.exe
__srt_narrow_argv_p...	1	0,00%	0,00%	79 946,00	79 946,00	Kursovoy_proekt.exe
powl	2	0,00%	0,00%	36 524,00	36 373,00	Kursovoy_proekt.exe
__srt_common_main...	1	93,12%	0,00%	113 001 451 020,00	72 619,00	Kursovoy_proekt.exe
std::basic_filebuf<cha...	2	0,00%	0,00%	402 277,00	35 879,00	Kursovoy_proekt.exe
_onexit	2	0,00%	0,00%	34 172,00	33 600,00	Kursovoy_proekt.exe
??0?\$basic_ostream@...	1	0,00%	0,00%	63 858,00	63 858,00	MSVCP140D.dll
std::use_facet<std::co...	2	0,00%	0,00%	159 348,00	31 156,00	Kursovoy_proekt.exe
__security_init_cookie	1	0,00%	0,00%	91 508,00	59 065,00	Kursovoy_proekt.exe
__srt_initialize_crt	1	0,00%	0,00%	97 215,00	46 167,00	Kursovoy_proekt.exe
std::basic_ofstream<c...	1	0,00%	0,00%	161 603,00	39 682,00	Kursovoy_proekt.exe
std::basic_filebuf<cha...	2	0,00%	0,00%	31 716,00	19 757,00	Kursovoy_proekt.exe
pre_c_initialization	1	0,00%	0,00%	220 191,00	38 302,00	Kursovoy_proekt.exe
std::basic_filebuf<cha...	2	0,00%	0,00%	20 318,00	15 602,00	Kursovoy_proekt.exe
??0?\$basic_istream@...	1	0,00%	0,00%	30 314,00	30 314,00	MSVCP140D.dll
invoke_main	1	93,12%	0,00%	113 000 918 417,00	28 297,00	Kursovoy_proekt.exe
__get_entropy	1	0,00%	0,00%	32 443,00	27 544,00	Kursovoy_proekt.exe
mainCRTStartup	1	100,00%	0,00%	121 347 960 750,00	26 945,00	Kursovoy_proekt.exe
__isa_available_init	1	0,00%	0,00%	51 048,00	26 558,00	Kursovoy_proekt.exe
__srt_initialize_defau...	1	0,00%	0,00%	23 793,00	23 793,00	Kursovoy_proekt.exe
pow	1	0,00%	0,00%	93 393,00	22 042,00	Kursovoy_proekt.exe
std::basic_ifstream<c...	1	0,00%	0,00%	39 638,00	20 665,00	Kursovoy_proekt.exe
__srt_is_managed_app	1	0,00%	0,00%	23 696,00	20 277,00	Kursovoy_proekt.exe
std::locale::~~locale	2	0,00%	0,00%	10 705,00	9 989,00	Kursovoy_proekt.exe
std::_Unique_ptr_base...	2	0,00%	0,00%	10 427,00	9 771,00	Kursovoy_proekt.exe
std::unique_ptr<std::...	1	0,00%	0,00%	59 101,00	19 254,00	Kursovoy_proekt.exe
std::unique_ptr<std::...	2	0,00%	0,00%	19 975,00	9 498,00	Kursovoy_proekt.exe
std::basic_filebuf<cha...	6	0,00%	0,00%	5 111,00	2 528,00	Kursovoy_proekt.exe
__srt_file_policy::set...	1	0,00%	0,00%	14 135,00	14 135,00	Kursovoy_proekt.exe
??1?\$basic_streambuf...	2	0,00%	0,00%	6 212,00	6 212,00	MSVCP140D.dll
__srt_main_policy::se...	1	0,00%	0,00%	10 274,00	10 274,00	Kursovoy_proekt.exe
??0?\$basic_streambuf...	2	0,00%	0,00%	4 978,00	4 978,00	MSVCP140D.dll
?getloc@?\$basic_stre...	2	0,00%	0,00%	4 660,00	4 660,00	MSVCP140D.dll
?always_noconv@cod...	2	0,00%	0,00%	4 623,00	4 623,00	MSVCP140D.dll
__srt_file_policy::set...	1	0,00%	0,00%	7 051,00	7 051,00	Kursovoy_proekt.exe
??0?\$basic_ios@DU?\$...	2	0,00%	0,00%	2 885,00	2 885,00	MSVCP140D.dll
??1?\$basic_ios@DU?\$...	2	0,00%	0,00%	2 845,00	2 845,00	MSVCP140D.dll
??0_Lockit@std@@@Q...	2	0,00%	0,00%	2 742,00	2 742,00	MSVCP140D.dll
?_Init@?\$basic_strea...	6	0,00%	0,00%	870,00	870,00	MSVCP140D.dll
[MSVCP140D.dll]	3	0,00%	0,00%	1 599,00	1 599,00	MSVCP140D.dll

operator new[]	6	0,03%	0,00%	5 132 441,00	779,00	Kursovoy_proekt.exe
std::basic_ifstream<c...	1	0,00%	0,00%	396 197,00	3 499,00	Kursovoy_proekt.exe
GetModuleHandleW	1	0,00%	0,00%	3 419,00	3 419,00	kernel32.dll
std::basic_filebuf<cha...	2	0,00%	0,00%	2 350,00	1 352,00	Kursovoy_proekt.exe
_guard_dispatch_icall...	2	0,00%	0,00%	1 276,00	1 276,00	Kursovoy_proekt.exe
??Bid@locale@std@...	2	0,00%	0,00%	1 228,00	1 228,00	MSVCP140D.dll
operator delete[]	3	0,00%	0,00%	1 897 272,00	800,00	Kursovoy_proekt.exe
_get_stream_buffer_p...	2	0,00%	0,00%	1 110,00	1 110,00	ucrtbased.dll
std::basic_ifstream<c...	1	0,00%	0,00%	15 012,00	2 160,00	Kursovoy_proekt.exe
GetSystemTimeAsFile...	1	0,00%	0,00%	2 037,00	2 037,00	kernel32.dll
[MSVCP140D.dll]	3	0,00%	0,00%	597,00	597,00	MSVCP140D.dll
??1_Lockit@std@@Q...	2	0,00%	0,00%	823,00	823,00	MSVCP140D.dll
_sclr_narrow_enviro...	1	0,00%	0,00%	1 526,00	1 526,00	Kursovoy_proekt.exe
pow<long double,do...	1	0,00%	0,00%	4 249,00	1 429,00	Kursovoy_proekt.exe
std::locale::_Getfacet	2	0,00%	0,00%	781,00	697,00	Kursovoy_proekt.exe
std::basic_ofstream<c...	1	0,00%	0,00%	2 002 857,00	1 381,00	Kursovoy_proekt.exe
_sclr_initialize_onexi...	1	0,00%	0,00%	2 165,00	1 294,00	Kursovoy_proekt.exe
std::basic_filebuf<cha...	2	0,00%	0,00%	6 959,00	646,00	Kursovoy_proekt.exe
std::basic_ofstream<c...	1	0,00%	0,00%	5 975,00	1 217,00	Kursovoy_proekt.exe
std::dynamic initializ...	1	0,00%	0,00%	5 102,00	1 174,00	Kursovoy_proekt.exe
GetCurrentProcessId	1	0,00%	0,00%	1 173,00	1 173,00	kernel32.dll
?_Decref@facet@loca...	2	0,00%	0,00%	531,00	531,00	MSVCP140D.dll
_malloc_dbg	1	0,00%	0,00%	1 030,00	1 030,00	ucrtbased.dll
std::_Compressed_pai...	2	0,00%	0,00%	584,00	513,00	Kursovoy_proekt.exe
atexit	2	0,00%	0,00%	34 676,00	504,00	Kursovoy_proekt.exe
?_Init@\$basic_strea...	2	0,00%	0,00%	503,00	503,00	MSVCP140D.dll
QueryPerformanceCo...	1	0,00%	0,00%	1 004,00	1 004,00	kernel32.dll
pow	1	0,00%	0,00%	1 341,00	1 003,00	Kursovoy_proekt.exe
_sclr_acquire_startup...	1	0,00%	0,00%	1 655,00	990,00	Kursovoy_proekt.exe
_sclr_is_ucrt_dll_in_use	3	0,00%	0,00%	321,00	321,00	Kursovoy_proekt.exe
std::basic_ofstream<c...	1	0,00%	0,00%	702 581,00	934,00	Kursovoy_proekt.exe
_free_dbg	1	0,00%	0,00%	889,00	889,00	ucrtbased.dll
std::_Fac_node::_scala...	1	0,00%	0,00%	5 095,00	808,00	Kursovoy_proekt.exe
std::basic_ifstream<c...	1	0,00%	0,00%	1 437,00	801,00	Kursovoy_proekt.exe
_crt_fast_decode_poi...	2	0,00%	0,00%	571,00	394,00	Kursovoy_proekt.exe
std::basic_ofstream<c...	1	0,00%	0,00%	3 028,00	740,00	Kursovoy_proekt.exe
std::unique_ptr<std::...	1	0,00%	0,00%	2 685,00	730,00	Kursovoy_proekt.exe
_RTC_Initialize	1	0,00%	0,00%	724,00	724,00	Kursovoy_proekt.exe
std::_Facet_Register	1	0,00%	0,00%	2 295,00	706,00	Kursovoy_proekt.exe
std::basic_ifstream<c...	1	0,00%	0,00%	701,00	701,00	Kursovoy_proekt.exe
std::_Compressed_pai...	1	0,00%	0,00%	1 157,00	696,00	Kursovoy_proekt.exe
std::_Unique_ptr_base...	1	0,00%	0,00%	1 204,00	695,00	Kursovoy_proekt.exe
std::_Unique_ptr_base...	1	0,00%	0,00%	1 903,00	694,00	Kursovoy_proekt.exe
GetCurrentThreadId	1	0,00%	0,00%	685,00	685,00	kernel32.dll
std::basic_ifstream<c...	1	0,00%	0,00%	103 746,00	666,00	Kursovoy_proekt.exe
??1?\$basic_istream@...	1	0,00%	0,00%	626,00	626,00	MSVCP140D.dll
std::unique_ptr<std::...	1	0,00%	0,00%	2 080,00	615,00	Kursovoy_proekt.exe
pow	1	0,00%	0,00%	2 630,00	614,00	Kursovoy_proekt.exe
?clear@\$basic_ios@...	1	0,00%	0,00%	588,00	588,00	MSVCP140D.dll
std::_Fac_node::~_Fac...	1	0,00%	0,00%	3 121,00	568,00	Kursovoy_proekt.exe
_crt_rotate_pointer_v...	3	0,00%	0,00%	184,00	184,00	Kursovoy_proekt.exe
_crt_fast_encode_poi...	1	0,00%	0,00%	717,00	517,00	Kursovoy_proekt.exe
std::basic_ofstream<c...	1	0,00%	0,00%	479,00	479,00	Kursovoy_proekt.exe
std::_Fac_node::opera...	1	0,00%	0,00%	1 501,00	471,00	Kursovoy_proekt.exe
std::_Fac_tidy_reg_t::~...	1	0,00%	0,00%	5 564,00	469,00	Kursovoy_proekt.exe
std::_dynamic atexit d...	1	0,00%	0,00%	5 999,00	435,00	Kursovoy_proekt.exe
_sclr_release_startup...	1	0,00%	0,00%	569,00	425,00	Kursovoy_proekt.exe
std::basic_filebuf<cha...	1	0,00%	0,00%	523,00	412,00	Kursovoy_proekt.exe
std::_Compressed_pai...	1	0,00%	0,00%	465,00	408,00	Kursovoy_proekt.exe
post_pgo_initialization	1	0,00%	0,00%	24 190,00	397,00	Kursovoy_proekt.exe
_sclr_initialize_mta	1	0,00%	0,00%	384,00	384,00	Kursovoy_proekt.exe
std::forward<std::_Fa...	1	0,00%	0,00%	414,00	367,00	Kursovoy_proekt.exe
std::unique_ptr<std::...	1	0,00%	0,00%	364,00	364,00	Kursovoy_proekt.exe
?_Incref@facet@local...	1	0,00%	0,00%	299,00	299,00	MSVCP140D.dll
std::_Fac_node::opera...	1	0,00%	0,00%	1 166,00	277,00	Kursovoy_proekt.exe
_sclr_is_user_mather...	1	0,00%	0,00%	259,00	259,00	Kursovoy_proekt.exe
??1?\$basic_ostream@...	1	0,00%	0,00%	253,00	253,00	MSVCP140D.dll
_RTC_Terminate	1	0,00%	0,00%	131,00	131,00	Kursovoy_proekt.exe
std::_Fac_node::_Fac_n...	1	0,00%	0,00%	88,00	88,00	Kursovoy_proekt.exe