

# CS225 Assignment 1 - "Where have I been ?"

Evdzhan Mustafa

March 28, 2014

## 1 Introduction

This is my report explaining my solution for the CS22510(2013-2014) Assignment 1. It explains my design decisions and assumptions in order to tackle the problem, using C, C++ and JAVA.

## 2 Design, implementation and assumptions

In each of the languages, I created a function that reads a line from the specified files. If the line is RMC or GSV sentence, those lines get processed, and the data they yield is stored appropriately in the correct object/structures. If anything else is read, it is discarded. On returning the function returns integer that tells me what was read.

Having such function, I use while loops to get good satellite fix from both streams. An assumption I made here is that on reaching GSV sentences and processing it, every subsequent RMC sentence is regarded as valid or not valid based on that GSV sentence. To achieve that, I simply use a boolean flag, which is updated every time a set of GSV sentences are read.

Having valid satellite fix from both streams, I now synchronise the times. I achieve this by using my first function, on both streams, until RMC sentences are read. Those first RMC sentences tell me the current time for each stream. I compare those two times, and synchronise them by continuing to read from the earlier one, until both times equalise.

Having synchronised those two, I start my main operation of continually fetching RMC sentences from both streams, and processing any GSV lines on the way. Another assumption I made here is that each RMC sentence from the first stream, has a matching pair from the second one. I believe this is true, since I did not see any mistakes in my output files. However, my program will indeed fail if one RMC sentence is missing from one of the streams, but is present in the second.

My main operation consists of the following :

- 1 - Read from both streams the next pair of RMC sentences.
- 2 - Check if the current satellites in streams are OK.

- 3 - If both streams are OK, put the location of one of the streams into a container (array/linked list/vector) and calculate the offset.
- 4 - If only one stream is OK, store the location, and update the bad stream's location using the offset.
- 5 - If two streams fail, read the next pair of RMC sentences.

This is put in one big while loop, that exits only if one of the files ends. When it happens, I output everything from the container into a GPX file using appropriate GPX format.

### 3 Environment used

My JAVA, C and C++ programs were developed on Linux Mint 16 machine (one version greater than those in the Delphinium(C55)).

The JAVA program was written using the Eclipse Kepler IDE. Java Runtime Environment version - 1.7.0.50 .

The C program was developed using NetBeans 8.0 IDE. The compiler used was gcc version 4.8.1. The standard used was -std=gnu90 . I used some non-standard ANSI C functions such as strsep, as well as timediff.

The C++ was again developed using NetBeans 8.0 IDE. The compiler version was g++ 4.8.1. Standard used - -std=gnu++98 . Yet again I used some non-standard ANSI functions.

The compilation command for the C and C++ programs was -Werror -Wall. That is all warnings to errors, and show all warnings.