ENGG1410: Introductory Programming for Engineers

Mini Project #2: GPS Position Calculation

Mohamad Abou El Nasr
School of Engineering, University of Guelph
Fall 2022

Start Date: Week #10 2022
Duration: 2 Weeks
Report Due Date: Week #12 2022, In the Drop Box

1 Objectives:

In the previous Labs you learned several skills that can assist you in developing complex applications and projects by implementing an encryption/decryption utility. In this lab, you will be asked to develop a small program with functions dealing with structures and array of structures as well file input output.

The main objectives of this lab is to introduce you to:

- concepts of procedural and data abstraction .
- The use of libraries as tools for encapsulating data objects and applicable operators.
- Introduce storage classes extern, static, and typedef and structures.

2 Introduction

21 global positioning satellites (GPSs) are in orbit around the earth. they are distributed such that at least four satellites are visible from any point on earth at any time. each satellite continuously broadcasts its identity, its status, its position, and the current value of its onboard clock. since the clocks are very accurate and the speed of transmission of radio waves is very well known, a small handheld receiver can get signals from the satellites visible to it and, by subtracting the times of origination and receipt of each signal, can deduce its own position relative to the position broadcast by each satellite.

3 Main Topic of Mini Project #2

There are many applications for this reliable positioning information. You are developing a library of functions to be used in conjunction with a GPS handheld receiver to allow a user to perform his or her own special positioning applications.
Declare two variables of type structure *user_t*; the variable *our_user* in which to store your users own

position and current time, and an array of same structures type named *other_users* with the same data for each of several other users we need to be able to find.
A position is represented by type double components *latitude*, *longitude*, and *latitude*. component *longitude* represents the distance West of a reference longitude, in meters. For this problem assume all users are South and West of these references, and assume that distances are relatively small to avoid some second order effects that would occur if a significant part of the Earth's surface were involved. type double component *time* should represent elapsed nanoseconds since some reference time. String component *name* should hold the name of the user, This is an array of characters.

4 Requirements

Enter, compile and **execute** the program that implements the following functionality:

1. Function scan_user that scans position and time data into one user_t structure.
2. Function that calls scan_user to fill the *our_user* structure and the *other_users* array.
3. Function that calculates the difference in position between *our_user* and each of the *other_users* and stores these distances in an **array of structures**. This array of structures should consist of records each of which contains the name of another user and the distance relative to our user. because we assume small distances, the distance can be calculated approximately as

   $$\sqrt{(lat_1 - lat_2)^2 + (long_1 - long_2)^2 + (alt_1 - alt_2)^2}$$

   Where lat1, long1, and alt1 refer to the position of one user, lat2, long2, and alt2 Indicate the position of another user.
4. Function that searches through this relative position array of structures and finds the closest other user to our user, returning the structure representing this closest user.
5. Write a main program that calls these library functions to scan position and time data and to find the name and position of the nearest user.
6. Give your program the ability to work from an input text file that has the data of other users stored in it. The first line of this text file is an integer indicating the number of other users to work with followed by the records of these other users. Ask the user for the path of this file and then use it to scan_user_file to fill the *other_users* array for your calculations.

5 Deliverables

1. **Lab Final Report File**:
   - Name your file as follows:
     ENG1410_F22_MIniProject2_Section(Wed,Thur,..)_Group#_FinalReport.pdf
   - Check the format of the final report below in the next section.
   - Keep the first page only for the title page.
   - Only one submission is allowed.

2. **Zipped Project File**:

   - Name your file as follows:
     ENG1410_F22_MIniProject2_Section(Wed,Thur,..)_Group#_Project.zip
   - The project file should have a README file that explains the input and output of your program.

3. **DEMO**:

   - You will need to demonstrate to the Teaching Assistant that your program fulfills the requirements.
   - The DEMO will take place during the LAB hours and you will show what you have finished until the time of the lab, not necessary the full project, but be ready to answer conceptual question on the still unfinished parts if any.

6 Report

Below is the general format of the report required:

1. **Title Page**:

   - Course #, and Date
   - Lab # and name of experiment
   - Your Group #, and Names

2. **Start a new page** and explain how you implemented your design by providing the following:

   (a) **Problem Statement**,
      i. Briefly describe the problem solved in the lab.

   (b) **Assumptions and Constraints**.
      i. Constraints could be for example no optimization with compilation.

   (c) **How you solved the Problem**,
      i. Flowchart.
      ii. Pseudo Code.
      iii. Block Diagram.

   (d) **System Overview & Justification of Design**
      i. Give an overview of the system to be designed.
      ii. Briefly explain how the system works and reasons behind the design.

3. **Error Analysis**

   (a) Confirm no syntax errors are present.

   (b) Confirm no semantic and logical errors are present.

   (c) Describe any problems with your program.

   (d) If no problems in the final system, describe problems/errors encountered during the development and how they were resolved.