Homework 1 - Extracting Data from a CSV file

Michael McAlpin
Instructor - COP3502 - CS-1
Spring 2017
EECS-UCF
michael.mcalpin@ucf.edu

January 25, 2017

Abstract

This assignment is based on a class of problem solved in enterprise computing; extraction, transformation, and loading. This is often referred to as ETL. The inputs will be data extracted from a leading aviation industry data and consulting firm, GCR. (See GCR.com for additional data.) The data is in a well known format where each data element is separated from the previous and following data elements by using a comma. It should be noted that this method of data manipulation is extremely common. The explicit order of the data fields and the desired outputs are defined in the "Specifications".

1 Objectives

The objectives of this assignment are to demonstrate proficiency in file I/O, data structures, and data transformation using C language resources.

1.1 Inputs

There are two basic inputs, the input file name, passed via the command line, and the input file data defined below.

1.1.1 Command Line arguments

The input file name will be input as follows:

- `hw1etl filename.ext`
- In the event that the input file is not available or there is an error finding the file, an appropriate error message shall be displayed. Use the example below for guidance.
- `hw1etl ERROR: File "bogusFilename" not found.`

1.1.2 Input File fields

The CSV input file contains the following fields. Please note these fields may vary in size, content, and validity of the data. Also note that some of the data formats are a *melange* of types. Specifically, note that both latitude and longitude contain numbers, punctuation, and text. Likewise, the FAA Site number contains digits, letters, and punctuation. (*This assignment will treat all input data as character data.*)

Table 1: Airports Data Fields

Field Title	Description	Size
FAA Site Number	Contains leading digits followed by a decimal point and short text	Leading digits followed by a decimal point and zero to two digits and a letter
Loc ID	The airport's short name, i.e. MCO for Orlando	4 characters
Airport Name	The airport's full name, i.e. Orlando International	~30 characters
Associated City	The nearest city	~25 characters
State	State	2 characters
Region	FAA Region	3 characters
ADO	Airline Dispatch Office	3 characters
Use	Public or Private	2 characters
Latitude	DD-HH-MM.MASDirection	Degrees, hours, minutes.milliarcseconds followed by either N,S,E or W. Treated as a string, for now
Longitude	See Latitude above.	ditto
Airport Ownership	Public or Private	2 characters
Part 139	FAA Regulation	No data
NPIAS Service Level	National Plan Integrated Airport Systems Descriptor	~10 characters
NPIAS Hub Type	Intentionally left blank	n/a
Airport Control Tower	Y/N	one character
Fuel	Fuel types available	up to 6 characters
Other Services	Collections of tag indicating INSTRuction, etc.	12 characters
Based Aircraft Total	Number of aircraft (may be blank)	Integer number
Total Operations	Takeoffs/Landings/etc (may be blank)	Integer number

2 Outputs

The outputs of the program will be populated `Struct airPdata` data. This data will be formatted so as to provide output define in the following sections.

2.1 Data Structure

The structure `struct airPdata` is described below. Please note the correlation with the data file’s *Field Names* refer to Table 1 on page 2 for more information.

```
typedef struct airPdata{
  char *siteNumber; //FAA Site Number
  char *LocID;   //Airport’s ``Short Name’’, ie MCO
  char *fieldName; //Airport Name
  char *city;    //Associated City
  char *state;   //State
  char *latitude; //Latitude
  char *longitude; //Longitude
  char controlTower;//Control Tower (Y/N)
} airPdata;
```

2.2 File output

The file output for this assignment is *stdout*, aka the console. Make sure there is a headline that names each column. For example:

```
FAA Site#    Short Name Airport Name                    City    ST      Latitude        Longitude      Tower
==========   ========== ================                ======= ==      ==============  =============== =
03406.20*H      2FD7    AIR ORLANDO                     ORLANDO FL      28-26-08.0210N  081-28-23.2590W N
03406.31*H      3FD5    ARNOLD PALMER HOSPITAL          ORLANDO FL      28-31-21.0090N  081-22-49.2520W N
03406.36*H      2FL5    BROOKSVILLE INTL AIRWAYS- INC   ORLANDO FL      28-25-26.0000N  081-27-35.0000W N
03406.24*H      FD99    DR P PHILLIPS HOSPITAL          ORLANDO FL      28-25-43.0220N  081-28-38.2590W N
03408.*A        ORL     EXECUTIVE                       ORLANDO FL      28-32-43.7000N  081-19-58.5000W Y
03406.11*H      37FA    FLORIDA HOSPITAL                ORLANDO FL      28-34-32.0020N  081-22-06.2490W N
03406.22*H      FD36    FLORIDA HOSPITAL EAST ORLANDO   ORLANDO FL      28-32-26.7000N  081-16-51.0000W N
03406.40*H      FL76    HELI-PARTNERS I-DRIVE           ORLANDO FL      27-23-04.0000N  081-29-07.0000W N
03406.39*H      97FD    HELICOPTERS INTL                ORLANDO FL      28-27-51.8300N  081-27-35.8800W N
03407.2*A       ISM     KISSIMMEE GATEWAY               ORLANDO FL      28-17-23.3000N  081-26-13.5000W Y
03406.*C        91FL    LAKE CONWAY NORTH               ORLANDO FL      28-28-45.0140N  081-22-03.2510W N
03406.33*C      89FL    LAKE HIAWASSEE                  ORLANDO FL      28-31-45.0100N  081-28-51.2600W N
03407.15*A      54FD    LM-ETS                          ORLANDO FL      28-22-03.0000N  081-04-34.0000W N
03407.09*H      82FD    LOCKHEED MARTIN                 ORLANDO FL      28-26-48.4900N  081-27-03.6900W N
03406.18*H      32FL    MEYER                           ORLANDO FL      28-30-05.0120N  081-26-39.2560W N
03408.4*H       27FA    ORANGE COUNTY SHERIFF’S OFFICE  ORLANDO FL      28-30-27.0110N  081-24-48.2540W N
03407.*A        MCO     ORLANDO INTL                    ORLANDO FL      28-25-45.8000N  081-18-32.4000W Y
03406.21*H      FD28    ORLANDO RGNL MEDICAL CENTER     ORLANDO FL      28-31-31.0090N  081-22-37.2510W N
03407.1*A       SFB     ORLANDO SANFORD INTL            ORLANDO FL      28-46-37.1000N  081-14-05.7000W Y
03406.29*H      7FA5    PREMIUM                         ORLANDO FL      28-23-21.0000N  081-29-19.0000W N
03406.113*H     26FA    PRINCETON HOSPITAL              ORLANDO FL      28-34-06.0040N  081-26-02.2550W N
03406.14*A      01FA    RYBOLT RANCH                    ORLANDO FL      28-35-21.9970N  081-08-39.2290W N
03406.38*C      12FL    TIMBERLACHEN                    ORLANDO FL      28-35-34.0000N  081-24-14.0000W N
03406.34*H      0FL7    WKMG-TV                         ORLANDO FL      28-35-38.7000N  081-25-11.6000W N
03406.3*H       13FD    YELVINGTON                      ORLANDO FL      28-31-07.0090N  081-22-59.2520W N
```

3 Processing

The primary goal is to provide programmatic access to the data from the input CSV file. This must be accomplished using standard C file IO techniques. Also note that it is vital to utilize the *stuct airPdata* for all data retrieval/extraction. Likewise, use of the *stuct airPdata* is required for the file output.

3.1 Reading the input

There are several approaches to read the input. Perhaps the most important consideration is reading the line in for each airport. Please note that there is one line per airport. Also note, that once the line is read into the input buffer it might be advantageous to parse the input buffer based on the *comma* delimiter.

There are several approaches possible. Make sure to test on *Eustis* as line termination characters/behaviors vary amongst operating systems.

3.2 Displaying the data structure

There are no data conversions for this assignment, therefore it is to your advantage to deal with all data elements as *character data*.

3.3 Testing

There will be four (4) input files provided for program testing. They are described below.

Table 2: Test Files

Filename	Description
twolines.csv	Two lines of test data, where one line consists of lower case letters, one unique letter per field, the other line will consist of uppercase letters.
orlando5.csv	Five lines of Orlando airport data.
orlando.csv	All 26 of the Orlando airports.
florida.csv	All 877 of Florida's airports.

4 Grading

Scoring will be based on the following rubric:

Table 3: Grading Rubric

Percentage	Description
-100	Cannot compile on *Eustis*
-100	Cannot accept input filename as command line argument
- 30	Cannot read input file
- 30	Cannot initialize *struct airPdata* with input data
- 30	Cannot output *struct airPdata* data from the input file

5 Submission Instructions

The assignment shall be submitted via *WebCourses*. There should be three files in the submission.

- The main source file named `hw1etl.c` (HW1ETL in caps... to prevent misreading the filename.)
- The *struct airPdata* include file, named airPdata.h
- A readme.doc file containing the following statement -"Your statement that the program is entirely your own work and that you have neither developed your code together with any another person, nor copied program code from any other person, nor permitted your code to be copied or otherwise used by any other person, nor have you copied, modified, or otherwise used program code that you have found in any external source, including but not limited to, online sources"