Assignment 3: awk

CS 3424 - Systems Programming — Sam Silvestro

Description

For this assignment, you will only use the features of `awk` discussed in this class. You may not make use of *any* external utilities/programs, with the exception of your `bash` script, which should simply contain one line used to invoke `awk` with your `awk` script.

You will create a *single awk script* which will take the contents of a comma-separated values file containing "calls for service" blotter data and produce a formatted reporting consisting of numbers representing statistics computed over this input data.

The input data is composed of records which consist of lines, with five fields delimited by commas present in each. These fields correspond to the following values:

Call ID, *Timestamp*, *Problem Type*, *Address*, *Division*

A simple, single example

Input file *a3-test-input1.csv*:

```
1  SAPD-2023-0381069,"2023-03-19 16:38:02","Welfare Check","13000 VISTA DEL
       NORTE",PRUE
2  SAPD-2023-0381068,"2023-03-19 16:37:46","Ordinance Violation","100
       WHITEWOOD DR",SOUTH
3  SAPD-2023-0381067,"2023-03-19 16:37:44",Information,"200 E HUISACHE AVE",
       CENTRAL
4  SAPD-2023-0381060,"2023-03-19 16:35:06","Traffic Related","200
       FREDERICKSBURG RD",CENTRAL
5  SAPD-2023-0381053,"2023-03-19 16:31:47",Threats,"15400 RED ROBIN RD",PRUE
6  SAPD-2023-0381052,"2023-03-19 16:31:07","Welfare Check","5500 UNIVERSITY
       HTS",PRUE
7  SAPD-2023-0381051,"2023-03-19 16:31:03","Assist the Public","400 W
       SOUTHCROSS BLVD",SOUTH
8  SAPD-2023-0381050,"2023-03-19 16:30:59","Welfare Check","2800 VANCE
       JACKSON RD",CENTRAL
9  SAPD-2023-0381048,"2023-03-19 16:30:47","Animal Related","500 THEO PKWY",
       SOUTH
10 SAPD-2023-0381045,"2023-03-19 16:30:07","Assault In Progress","1800 LAMAR
        ST",EAST
11 SAPD-2023-0381044,"2023-03-19 16:30:07","Criminal Mischief","13000 O'
       CONNOR RD",NORTH
12 SAPD-2023-0381043,"2023-03-19 16:29:09",Disturbance,"7600 GUILBEAU RD",
       PRUE
13 SAPD-2023-0381042,"2023-03-19 16:29:01",Disturbance,"5000 HAMILTON WOLFE"
       ,PRUE
```

```
SAPD-2023-0381040,"2023-03-19 16:28:14","Traffic Related","PITLUK AVE / QUINTANA RD",SOUTH
SAPD-2023-0381039,"2023-03-19 16:27:42",Theft,"8700 FREDERICKSBURG RD",PRUE
SAPD-2023-0381037,"2023-03-19 16:27:19",Disturbance,"20 NE LOOP 410",NORTH
SAPD-2023-0381036,"2023-03-19 16:26:43","Shoplifting In Progress","2100 S LAREDO ST",CENTRAL
SAPD-2023-0381035,"2023-03-19 16:26:12",Disturbance,"1400 MIRA VISTA",WEST
SAPD-2023-0381034,"2023-03-19 16:26:03","Burglary Vehicle","12200 STATE HWY 16 S",SOUTH
SAPD-2023-0381029,"2023-03-19 16:23:33",Disturbance,"11900 PERRIN BEITEL",NORTH
```

Output of running: `./assign3.bash a3-test-input1.csv`

```
Total calls = 19

Date: 2023-03-19
       First call: SAPD-2023-0381029,"2023-03-19 16:23:33",Disturbance,"11900 PERRIN BEITEL",NORTH
        Last call: SAPD-2023-0381069,"2023-03-19 16:38:02","Welfare Check","13000 VISTA DEL NORTE",PRUE

Per-problem totals:
        Theft: 1
        "Assist the Public": 1
        "Traffic Related": 2
        "Ordinance Violation": 1
        "Shoplifting In Progress": 1
        "Burglary Vehicle": 1
        Disturbance: 5
        "Welfare Check": 3
        Threats: 1
        "Criminal Mischief": 1
        "Animal Related": 1
        Information: 1

Per-division totals:
        PRUE: 6
        SOUTH: 5
        WEST: 1
        NORTH: 3
        CENTRAL: 4
```

Requirements

For this part of the assignment, you will only use the program features covered in class so far, specifically pertaining to `awk` and `bash`. Do *not* make use of Python, `sed`, `grep`, `find`, `sort`, `cut`, `tr`, or any other programs/utilities other than `awk` and `bash`. Furthermore, you must utilize only those program features that appear in the lecture notes (sorry, ChatGPT).

After reading all input, your `awk` script will print the following statistics:

- Total number of service calls read from input
- First call for service for each date
- Last call for service for each date
- Per-problem-type totals (i.e., the number of calls dispatched under each problem type that occurs in the input)
- Per-division totals (i.e., the number of calls dispatched under each substation division that occurs in the input)

Note: You may assume there will be no tie-breaking necessary when determining the first and last service calls for a given day.

Script Execution

Your program should be capable of invocation through a single bash file with input taken from standard input as well as one or more filenames provided directly on the command line (do *not prompt* the user for these filenames; they are supplied *only* on the command line; this is to say, you *cannot* make use of the `read` builtin command). The resulting output should be printed directly to standard output.

```
$ assign3.bash < example1.txt
```
or
```
$ assign3.bash example1.txt example2.txt
```
or
```
$ assign3.bash example1.txt example2.txt .....  exampleN.txt
```

Assignment Data

Sample input files can be found in:
`/usr/local/courses/ssilvestro/cs3424/Spring24/assign3`.

Script Files

Your submission should consist of multiple files:

- **`assign3.bash`** - a bash script with a single line of code (i.e., one command) used to invoke `awk` with your script.
- **`assign3.awk`** - your `awk` script used to produce the reporting data as specified previously.

Verifying Your Programs

You may use the provided sample data files (named `a3-test-input[0-9].csv`) as input to your bash script (and thus, indirectly, will be passed to your `awk` script). You may then validate this output against the numbers shown above. Lastly, you may use multiple example files to demonstrate your script's ability to process them together. The invocation for these attempts will look like the following:

```
./assign3.bash a3-test-input1.csv
```

as well as:

```
./assign3.bash a3-test-input1.csv a3-test-input2.csv
```

and so forth and so on, until you are satisfied that the results produced are correct.

If you encounter a discrepancy, I would recommend that you reduce the size of these example input files to the smallest size possible that will still reproduce the problem. It is must simpler to troubleshoot the incorrect computation over an input file consisting of five lines as compared to another containing fifty.

Submission

As usual, you will submit your assignment via Blackboard. Your submission is a *ZIP file*, named **`a3-`*`abc123`*`.zip`**, which should contain *only* your bash and `awk` files. The *abc123* represents your individual myUTSA ID.