Simulating Voter Wait Times

User Manual

Introduction

This guide will walk you through everything you need to know about using the Voter Simulator program. We will show you what you need to type in so that you do not run into any errors when using the program. Make sure that you have the latest version of C++ installed on your computer so that the code will be able to run properly. Open up the terminal and navigate to the program by typing "cd [folder name]" then pressing enter. You can check where you currently are and what folders are available to enter by typing "Is" in the terminal (Figure 1).

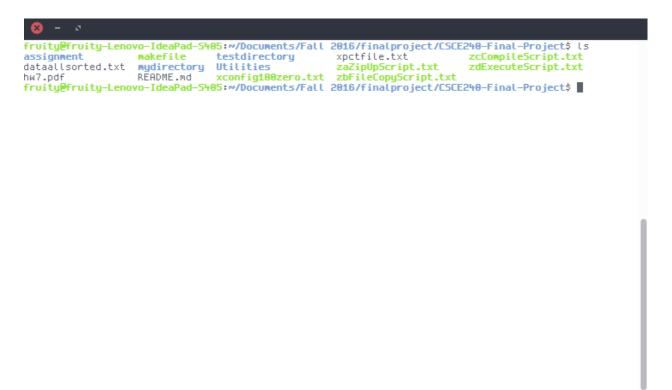


Figure 1 - Using 'Is' to see contents of the directory

Now you can see what files are in the main directory of the program. We will mainly be focusing on zaZipUpScript.txt, zbFileCopyScript.txt, zcCompileScript.txt, and zdExecuteScript.txt as scripts that run the program. Before we are able to use the program, we need a text file

containing the actual service times of the voting to the ones that the program calculates. This needs to be in the same directory as the four scripts listed above, and it should be named "dataallsorted.txt".

We also need to have a configuration file and precinct file in this same directory. If the configuration file does not already exist, create a text file named "xconfig100zero.txt". This file should have two lines: the first having seven integer values and the second having any number of doubles, which are just numbers with decimal places (Figure 2). The integer values should be separated by a space and are as follows for the first line:

- 1. The seed number for the random number generator
- 2. The number of hours in the election day
- 3. The mean time it takes to vote
- 4. The minimum number of voters per precinct
- 5. The maximum number of voters per precinct
- 6. The waiting time in minutes that is considered "too long"
- 7. The number of iterations to perform

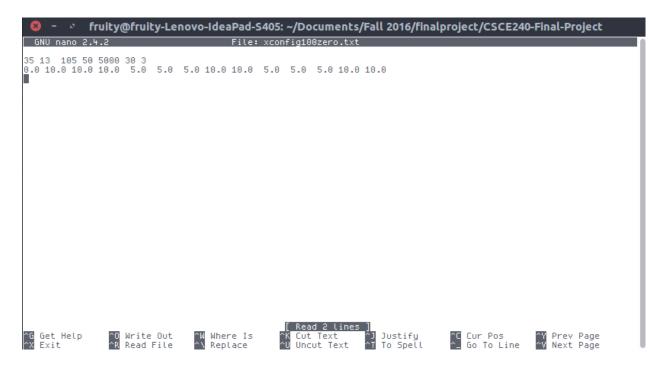


Figure 2 - The configuration file

The second line containing doubles should add up to 100, with the first number being the percentage of voters who voted early, and each one after that being the percentage of voters arriving for that hour. Now, if the precinct file does not already exist, create another text file called "xpctfile.txt" (Figure 3). There may be numerous lines in this file, but each one should be the same basic format, with each part separated by a space:

- 1. Precinct number (an integer)
- 2. Precinct name
- 3. Precinct turnout (a double)
- 4. Precinct number of voters (an integer)
- 5. Precinct expected voters (an integer)
- 6. Precinct expected per hour (an integer)
- 7. Precinct number of voting stations (an integer)

- 8. Precinct minority (a double)
- 9. Number of stations to histogram (an integer)
- 10. Number of stations to histogram (an integer)
- 11. Number of stations to histogram (an integer)

The first 8 values are self-explanatory values needed to do the calculation. The last 3 should only be above 0 if you want to make a histogram of voting stations. Make sure you have exactly eleven values separated by one space each for the code to work.

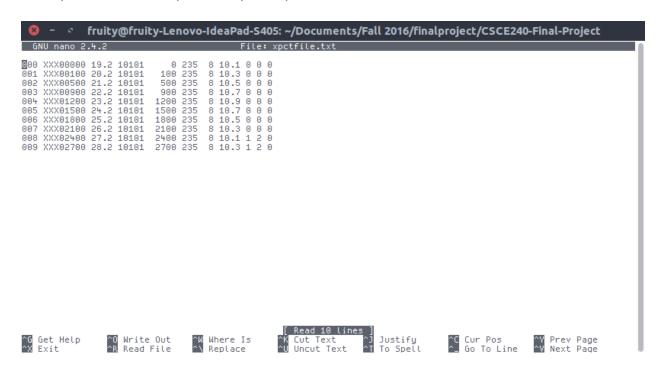


Figure 3 - The precinct file

We should make sure that the zdExecuteScript.txt has everything needed for it to run.

Open up the file in your favorite text editor and look at the line under "cd \$item". It should start with "./Aprog" and be followed by the configuration and precinct files that we made, each separated with a space. We need to make sure there are three more words separated by a

space: "zzout", "zzlog", and ">zzstdout0013". These create an output file, log file, and a record of the standard output respectively.

```
fruity@fruity-Lenovo-IdeaPad-S405: ~/Documents/Fall 2016/finalproject/CSCE240-Final-Project
GNU nano 2.4.2
                                            File: zdExecuteScript.txt
∰!/bin/bash
       Descend into "testdirectory" directory
cd testdirectory
for item in *
  echo " "
echo "EXECUTING" $item
  cd Sitem
  ./Aprog ../../xconfig100zero.txt ../../xpctfile.txt zzout zzlog >zzstdout0013
cd ..
echo "EXECUTION COMPLETE"
done
echo "Return from 'testdirectory' directory"
                                                          [ Read 16 lines ]
K Cut Text ^3 Justify
                                                                                              ^C Cur Pos
^_ Go To Line
^G Get Help
^X Exit
                   ^O Write Out
^R Read File
                                      ^W Where Is
^\ Replace
                                                         ^K Cut Text
^U Uncut Text
                                                                                                                 ^Y Prev Page
^V Next Page
```

Figure 4 - The zdExecuteScript.txt file

Now we have all the necessary files and we can run zaZipUpScript.txt, zbFileCopyScript.txt, zcCompileScript.txt, and zdExecuteScript.txt in order. To do this, type "./[filename]" to execute each file as a script. The names of the scripts explain exactly what they do: zip up the program, copy it to a different directory, compile it, and finally execute it.

Once you run those, we can see the output of the file by changing directories into testdirectory, and then group7_hw7. Open up zzout to see the final calculations the program made from the data.

```
8 - 8
  GNU nano 2.4.2
                                                                  File: zzout
MAIN: Beginning execution MAIN:
Wed Nov 30 23:38:02 2016
MAIN: outfile 'zzout'
MAIN: logfile 'zzlog'
MAIN:
CONFIG: RN seed: 35
CONFIG: Election Day length: 46800 = 13.00 ( 13.00
CONFIG: Time to vote mean: 105 = 1.75 minutes
CONFIG: Min and max expected voters for this simulation:
Mait time (minutes) that is 'too long': 30
Number of iterations to perform: 3
Max service time subscript: 12957
CONFIG: 0-0: 0.00
CONFIG: 6-7: 10.00
CONFIG: 7-8: 10.00
CONFIG: 8-9: 10.00
CONFIG: 9-10: 5.00
CONFIG: 10-11: 5.00
                                                                                    13.00) hours
                                                                                               50
                                                                                                                      5000
                            10.00
5.00
5.00
5.00
CONFIG: 10-11 :
CONFIG: 11-12 :
CONFIG: 12-13:

CONFIG: 13-14:

CONFIG: 14-15:

CONFIG: 15-16:

CONFIG: 16-17:
                            10.00
10.00
                            5.00
                               5.00
CONFIG: 17-18 : CONFIG: 18-19 :
                            10.00
SIM: RunSimulation for pct
                                                                        10101 100 235 8 10.30 HH 0 HH
10101 100 235 8 10.30 HH 0 HH
100 1 stations, mean/dev wait (mins) 0.45
100 1 stations, mean/dev wait (mins) 1.37
SIM: 1 XXX00100
OnePct: 1 XXX00100
OnePct: 0 1 XXX00100
OnePct: 1 1 XXX00100
                                                        20.20
                                                      20.20
                                                                                                                                                             1.05 toolong
                                                                                                                                                              2.82 toolong
                                                                                                                         ^O Write Out
^R Read File
                                                                         M-\ First Line
M-/ Last Line
^G Get Help
^X Exit
                                                ^W Where Is
^\ Replace
```

Figure 5 - Some of the output in zzout

This program is copyright of Dr. Buell. All rights reserved.