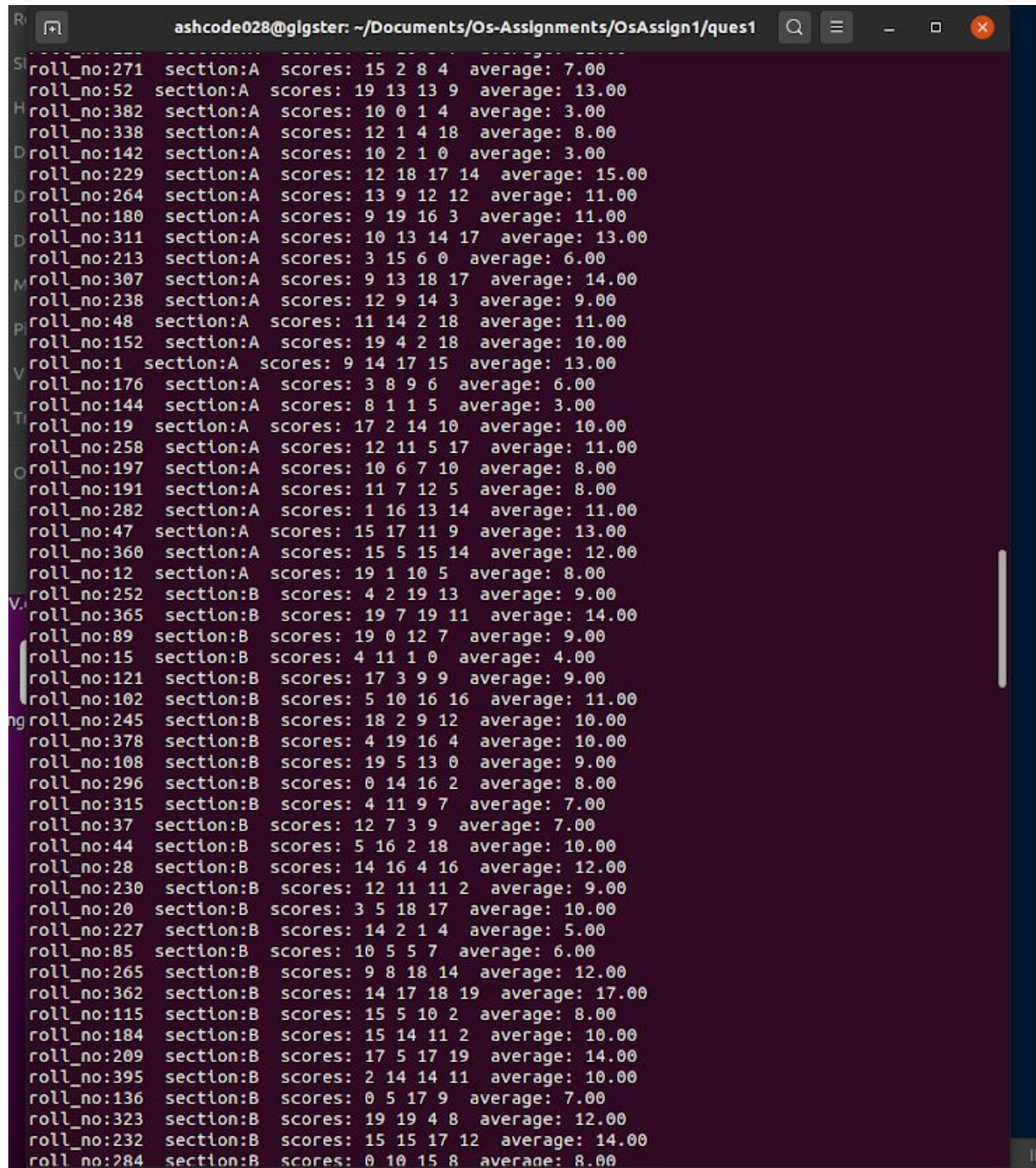


Os Assignment 1

Q1

In this ques , we were told to read csv files through syscalls.

Once we execute the make command in the terminal it prints the records of all students according the section and each records average as shown below.



```
ashcode028@glgster: ~/Documents/Os-Assignments/OsAssign1/ques1
roll_no:271 section:A scores: 15 2 8 4 average: 7.00
roll_no:52 section:A scores: 19 13 13 9 average: 13.00
roll_no:382 section:A scores: 10 0 1 4 average: 3.00
roll_no:338 section:A scores: 12 1 4 18 average: 8.00
roll_no:142 section:A scores: 10 2 1 0 average: 3.00
roll_no:229 section:A scores: 12 18 17 14 average: 15.00
roll_no:264 section:A scores: 13 9 12 12 average: 11.00
roll_no:180 section:A scores: 9 19 16 3 average: 11.00
roll_no:311 section:A scores: 10 13 14 17 average: 13.00
roll_no:213 section:A scores: 3 15 6 0 average: 6.00
roll_no:307 section:A scores: 9 13 18 17 average: 14.00
roll_no:238 section:A scores: 12 9 14 3 average: 9.00
roll_no:48 section:A scores: 11 14 2 18 average: 11.00
roll_no:152 section:A scores: 19 4 2 18 average: 10.00
roll_no:1 section:A scores: 9 14 17 15 average: 13.00
roll_no:176 section:A scores: 3 8 9 6 average: 6.00
roll_no:144 section:A scores: 8 1 1 5 average: 3.00
roll_no:19 section:A scores: 17 2 14 10 average: 10.00
roll_no:258 section:A scores: 12 11 5 17 average: 11.00
roll_no:197 section:A scores: 10 6 7 10 average: 8.00
roll_no:191 section:A scores: 11 7 12 5 average: 8.00
roll_no:282 section:A scores: 1 16 13 14 average: 11.00
roll_no:47 section:A scores: 15 17 11 9 average: 13.00
roll_no:360 section:A scores: 15 5 15 14 average: 12.00
roll_no:12 section:A scores: 19 1 10 5 average: 8.00
roll_no:252 section:B scores: 4 2 19 13 average: 9.00
roll_no:365 section:B scores: 19 7 19 11 average: 14.00
roll_no:89 section:B scores: 19 0 12 7 average: 9.00
roll_no:15 section:B scores: 4 11 1 0 average: 4.00
roll_no:121 section:B scores: 17 3 9 9 average: 9.00
roll_no:102 section:B scores: 5 10 16 16 average: 11.00
roll_no:245 section:B scores: 18 2 9 12 average: 10.00
roll_no:378 section:B scores: 4 19 16 4 average: 10.00
roll_no:108 section:B scores: 19 5 13 0 average: 9.00
roll_no:296 section:B scores: 0 14 16 2 average: 8.00
roll_no:315 section:B scores: 4 11 9 7 average: 7.00
roll_no:37 section:B scores: 12 7 3 9 average: 7.00
roll_no:44 section:B scores: 5 16 2 18 average: 10.00
roll_no:28 section:B scores: 14 16 4 16 average: 12.00
roll_no:230 section:B scores: 12 11 11 2 average: 9.00
roll_no:20 section:B scores: 3 5 18 17 average: 10.00
roll_no:227 section:B scores: 14 2 1 4 average: 5.00
roll_no:85 section:B scores: 10 5 5 7 average: 6.00
roll_no:265 section:B scores: 9 8 18 14 average: 12.00
roll_no:362 section:B scores: 14 17 18 19 average: 17.00
roll_no:115 section:B scores: 15 5 10 2 average: 8.00
roll_no:184 section:B scores: 15 14 11 2 average: 10.00
roll_no:209 section:B scores: 17 5 17 19 average: 14.00
roll_no:395 section:B scores: 2 14 14 11 average: 10.00
roll_no:136 section:B scores: 0 5 17 9 average: 7.00
roll_no:323 section:B scores: 19 19 4 8 average: 12.00
roll_no:232 section:B scores: 15 15 17 12 average: 14.00
roll_no:284 section:B scores: 0 10 15 8 average: 8.00
```

As the main function is evoked we create a process using fork().

Parent function reads the csv file ,computes the average of each student in section B.

Child process does the same thing but for section A students.

I used a struct called student to store roll no,section,scores in 4 assignments and its average.

I made an array of struct students and computed average differently for both processes.

I removed the storage of the first line i.e label so that the values are stored accordingly.

Error handling-

- While forking the current process -checked whether fork is being done properly or not($\text{pid} < 0$)
- In the child process ,after all the tasks are done I used `exit(0)` to avoid abrupt ending and signalling the parent process about the exit.
- In the parent process,I used the `waitpid()` command to wait till the child process exited then checked it using `WIFEXITED(status)` command.

Couldn't use read syscall instead used fgets-

- While using read syscall -i required file descriptor,buff to store ,and its size to be mentioned so that it can read that amount of bytes from the file.
- But when i used it i couldn't get the correct amount of size to be sent to the buffer since each line has different no of bytes requirement as in some columns roll no was 3 digit and marks in 1 digit.
- So if i read some amount of bytes for the first line if the second line required one more or less bytes got affected in the second run of loop so i couldn't extract the line and its values.Thus,incorrect values get stored into the specific fields of the struct thereby giving wrong output.

Syscalls ,arguments,outcomes-

- Read syscall-requires file descriptor
- Fgets-requires pointer to that file ,size of the input line and the pointer to store input line. Once the line is stored in the array,I used `string_token (strtok())`to break it into parts and store it in the respective fields of the struct(here student).
- Usually `open` -is used to get file descriptor since I used `fgets()`, I used `fopen` to get file pointer.Both take similar arguments name of the file and mode of the file to be opened as(`fopen-"r"."w"` ; `open- O_RDONLY,O_WRONLY` etc) .
- I used the `printf` function to print the values to the console .

Assumptions made-

- The input file must contain the first line as a label(student id,section,scores etc).
- The input file must have only these entries and in the same order as roll no,section,marks1,marks2,marks3,marks4.
- The input file is a space separated csv file named as "data.csv".