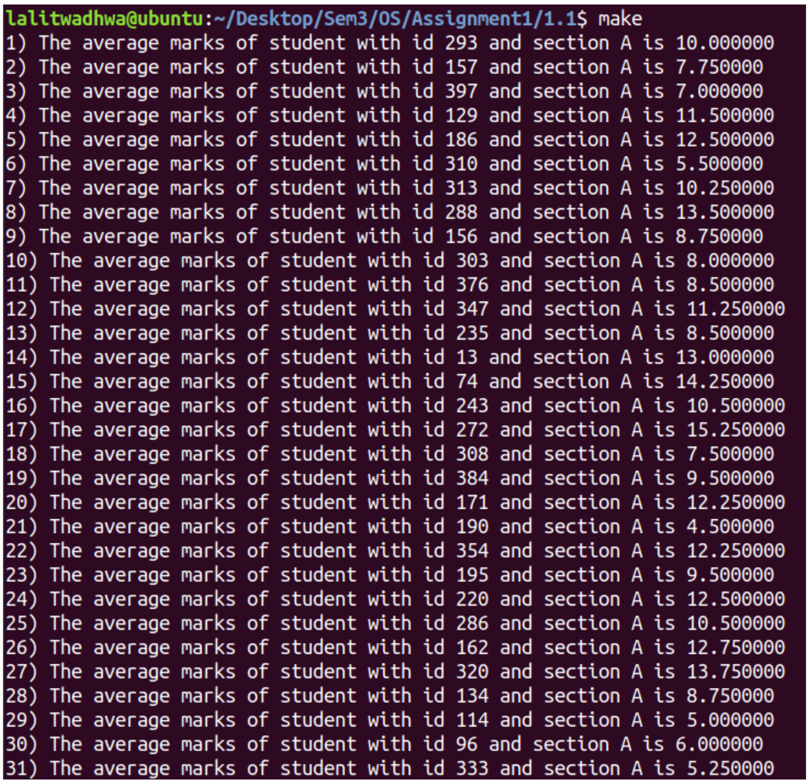
Exercise 1.1 Documentation

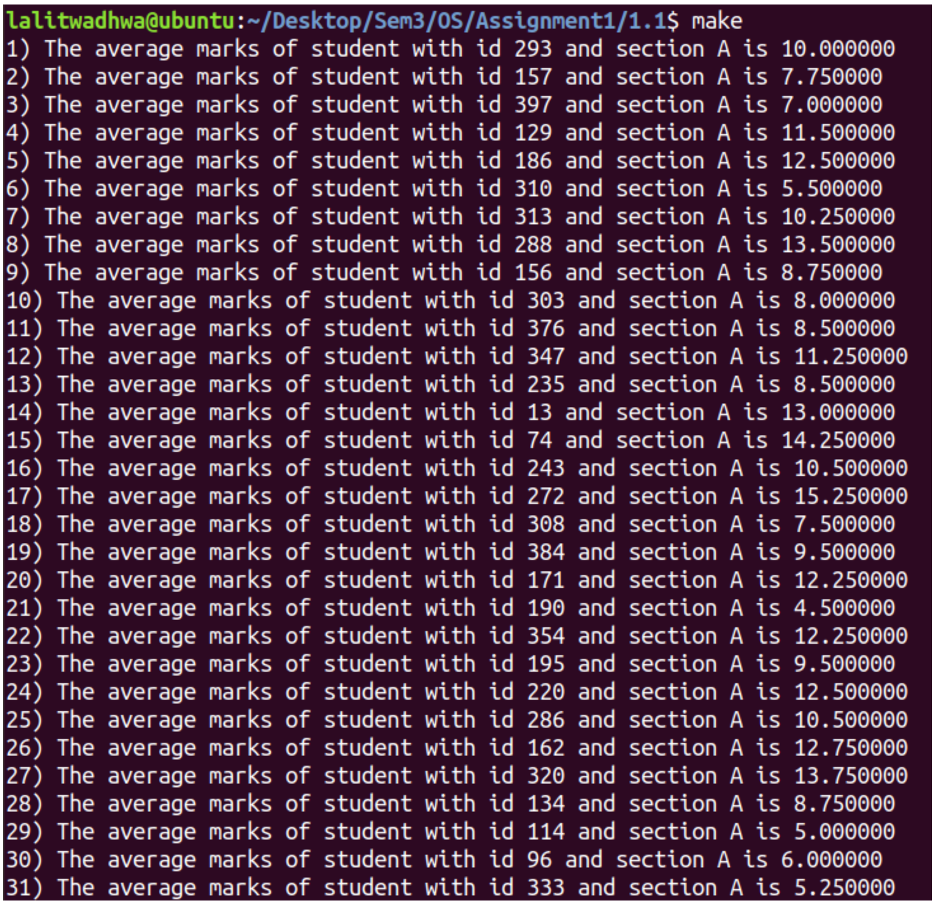
Process creation and termination using system calls

1.) How Program Works

- The program at the start declare the variables and the arrays to be used in the program.
- After that it calls the fork() command.
- In my machine, the parent is preferred to child, so when the command goes to the parent, there it encounters the waitpid command.
- At the same time, the child is also being executed
- In the child, it the first parses the given csv file using the open() and read() commands
- Then the data is divided into the arrays for ease of use.
- Since the child has to handle the section A students, it correspondingly accesses the data from the arrays and compute the average.
- Then using the write() system call, it prints the data on the screen.
- After this the child exits using the exit() command.
- After this, the parent's waiting period is over and it executes the above steps for the students of section B
- Output:



```
lalitwadhwa@ubuntu:~/Desktop/Sem3/OS/Assignment1/1.1$ make
1) The average marks of student with id 293 and section A is 10.000000
2) The average marks of student with id 157 and section A is 7.750000
3) The average marks of student with id 397 and section A is 7.000000
4) The average marks of student with id 129 and section A is 11.500000
5) The average marks of student with id 186 and section A is 12.500000
6) The average marks of student with id 310 and section A is 5.500000
7) The average marks of student with id 313 and section A is 10.250000
8) The average marks of student with id 288 and section A is 13.500000
9) The average marks of student with id 156 and section A is 8.750000
10) The average marks of student with id 303 and section A is 8.000000
11) The average marks of student with id 376 and section A is 8.500000
12) The average marks of student with id 347 and section A is 11.250000
13) The average marks of student with id 235 and section A is 8.500000
14) The average marks of student with id 13 and section A is 13.000000
15) The average marks of student with id 74 and section A is 14.250000
16) The average marks of student with id 243 and section A is 10.500000
17) The average marks of student with id 272 and section A is 15.250000
18) The average marks of student with id 308 and section A is 7.500000
19) The average marks of student with id 384 and section A is 9.500000
20) The average marks of student with id 171 and section A is 12.250000
21) The average marks of student with id 190 and section A is 4.500000
22) The average marks of student with id 354 and section A is 12.250000
23) The average marks of student with id 195 and section A is 9.500000
24) The average marks of student with id 220 and section A is 12.500000
25) The average marks of student with id 286 and section A is 10.500000
26) The average marks of student with id 162 and section A is 12.750000
27) The average marks of student with id 320 and section A is 13.750000
28) The average marks of student with id 134 and section A is 8.750000
29) The average marks of student with id 114 and section A is 5.000000
30) The average marks of student with id 96 and section A is 6.000000
31) The average marks of student with id 333 and section A is 5.250000
```



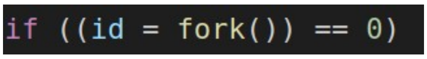
```
lalitwadhwa@ubuntu:~/Desktop/Sem3/OS/Assignment1/1.1$ make
1) The average marks of student with id 293 and section A is 10.000000
2) The average marks of student with id 157 and section A is 7.750000
3) The average marks of student with id 397 and section A is 7.000000
4) The average marks of student with id 129 and section A is 11.500000
5) The average marks of student with id 186 and section A is 12.500000
6) The average marks of student with id 310 and section A is 5.500000
7) The average marks of student with id 313 and section A is 10.250000
8) The average marks of student with id 288 and section A is 13.500000
9) The average marks of student with id 156 and section A is 8.750000
10) The average marks of student with id 303 and section A is 8.000000
11) The average marks of student with id 376 and section A is 8.500000
12) The average marks of student with id 347 and section A is 11.250000
13) The average marks of student with id 235 and section A is 8.500000
14) The average marks of student with id 13 and section A is 13.000000
15) The average marks of student with id 74 and section A is 14.250000
16) The average marks of student with id 243 and section A is 10.500000
17) The average marks of student with id 272 and section A is 15.250000
18) The average marks of student with id 308 and section A is 7.500000
19) The average marks of student with id 384 and section A is 9.500000
20) The average marks of student with id 171 and section A is 12.250000
21) The average marks of student with id 190 and section A is 4.500000
22) The average marks of student with id 354 and section A is 12.250000
23) The average marks of student with id 195 and section A is 9.500000
24) The average marks of student with id 220 and section A is 12.500000
25) The average marks of student with id 286 and section A is 10.500000
26) The average marks of student with id 162 and section A is 12.750000
27) The average marks of student with id 320 and section A is 13.750000
28) The average marks of student with id 134 and section A is 8.750000
29) The average marks of student with id 114 and section A is 5.000000
30) The average marks of student with id 96 and section A is 6.000000
31) The average marks of student with id 333 and section A is 5.250000
```

2.) System Calls

- The system calls used are:

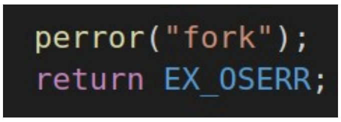
1. fork()

    - This command is used to create the child process.
    - No arguments are passed in it.
    - It returns:
        - 0 if returned to the newly created child process.
        - Negative Value: creation of a child process was unsuccessful.
        - Positive value: Returned to parent or caller. The value contains process ID of newly created child process.



```
if ((id = fork()) == 0)
```

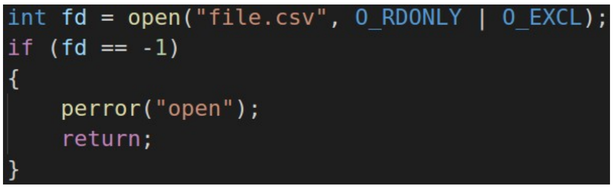
○



```
perror("fork");
return EX_OSERR;
```

2. open()

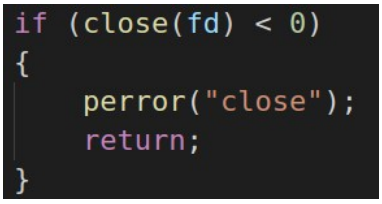
- Used to Open the file for reading, writing or both.
- It takes 2 arguments:
  - Path
  - Flags
- It returns:
  - file descriptor being used for the file.
  - -1 on failure



```c
int fd = open("file.csv", O_RDONLY | O_EXCL);
if (fd == -1)
{
    perror("open");
    return;
}
```

3. close()

- Used to close the file being pointed by the file descriptor.
- It takes 1 arguments:
  - file descriptor of the file to be closed
- It returns:
  - 0 on success
  - -1 on failure

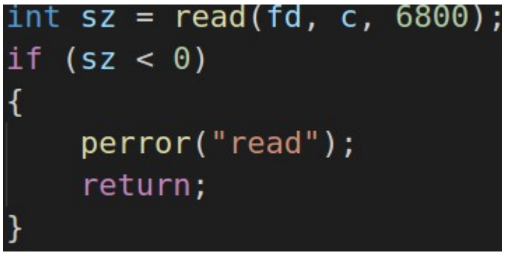


```c
if (close(fd) < 0)
{
    perror("close");
    return;
}
```

4. read()

- From the file indicated by the file descriptor fd, the read() function reads cnt bytes of input into the memory area indicated by buf.
- It takes 3 arguments:
  - fd: file descriptor
  - buf: buffer to read data from
  - cnt: length of buffer
- It returns:
  - return Number of bytes read on success
  - return 0 on reaching end of file

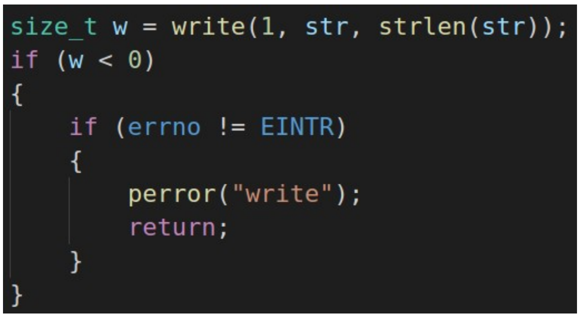
- return -1 on error
- return -1 on signal interrupt



```
int sz = read(fd, c, 6800);
if (sz < 0)
{
    perror("read");
    return;
}
```

5. write()

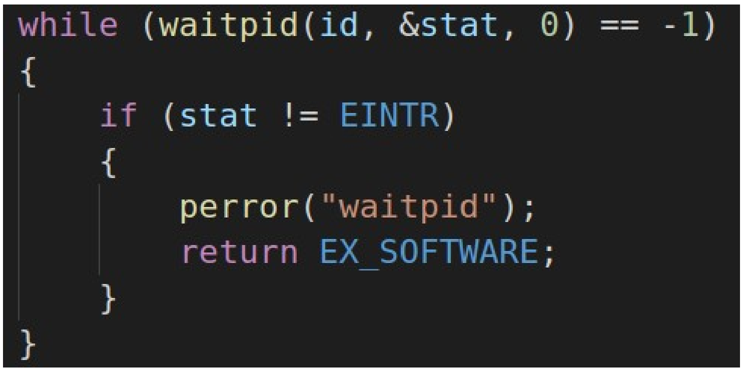
- Writes cnt bytes from buf to the file or socket associated with fd.
- It takes 3 arguments:
    - fd: file descripter
    - buf: buffer to write data to
    - cnt: length of buffer
- It returns:
    - return Number of bytes written on success
    - return 0 on reaching end of file
    - return -1 on error
    - return -1 on signal interrupt



```
size_t w = write(1, str, strlen(str));
if (w < 0)
{
    if (errno != EINTR)
    {
        perror("write");
        return;
    }
}
```

6. waitpid()

- if we want to reap any specific child process, waitpid is used
- it returns
    - pid of child, if child has exited
    - 0, if child hasn’t exited



```
while (waitpid(id, &stat, 0) == -1)
{
    if (stat != EINTR)
    {
        perror("waitpid");
        return EX_SOFTWARE;
    }
}
```