OS
Optional Assignment-4

Akshala Bhatnagar
2018012

- **The output of your program**

```
Car1 0
Car2 10
Car1 5
Car2 11
Car1 60                 //cheatmode is used
Car2 13
Car1 63
Car2 23
Car1 72
Car2 31
Car1 82
Car2 38
Car1 83
Car2 44
Car1 93
Car2 53
Car1 94
Car2 63
Car1 wins
```

- **How many pipes have you created? How are they being used?**
    - I have created 8 pipes.
    - fd_cheat_car1: This pipe is used between cheatmode and car1. This is used as a flag for whether cheat mode is being used for car1 or not. The cheatmode process writes 1 to the pipe if cheat mode is being used. Car1 reads from this pipe but makes it

non-blocking before reading. If the output of read is -1 then nothing is done, otherwise the position is changed.

- fd_cheat_car2: This pipe is used between cheatmode and car2. This is used as a flag for whether cheat mode is being used for car2 or not. The cheatmode process writes 1 to the pipe if cheat mode is being used. Car2 reads from this pipe but makes it non-blocking before reading. If the output of read is -1 then nothing is done, otherwise the position is changed.
- fd_cheat_car1_pos: This pipe is used between cheatmode and car1. This pipe has the car1 position after cheatmode is used. Cheatmode can write to this pipe and car1 reads from it.
- fd_cheat_car2_pos: This pipe is used between cheatmode and car1. This pipe has the car2 position after cheatmode is used. Cheatmode can write to this pipe and car2 reads from it.
- Fd_report_car1: This pipe is used between report and car1. This is used as a flag for when the race has terminated. The report writes 1 to the pipe if the race is over. Car1 reads from this pipe but makes it non-blocking before reading. If the output of read is -1 then nothing is done, otherwise the car1 process is killed.
- fd_report_car2: This pipe is used between report and car2. This is used as a flag for when the race has terminated. The report writes 1 to the pipe if the race is over. Car2 reads from this pipe but makes it non-blocking before reading. If the output of read is -1 then nothing is done, otherwise the car2 process is killed.
- fd_car1_pos: This pipe is used between report and car1. This pipe has the car1 position after the car1 process is performed. Car1 can write to it and report reads from it. Report needs to read from it as it needs to declare a winner when either of the cars cross 100 steps.
- fd_car2_pos: This pipe is used between report and car2. This pipe has the car2 position after the car2 process is performed. Car2 can write to it and report reads from it. Report needs to

read from it as it needs to declare a winner when either of the cars cross 100 steps.

- **How do you terminate all the processes when the race is over?**
  - Cheatmode: This process is killed by the kill system call
  - Car1: This process is terminated when the value read from fd_report_car1 pipe is 1. The pipe between report and car1 decides when to kill the process.
  - Car2: This process is terminated when the value read from fd_report_car2 pipe is 1. The pipe between report and car2 decides when to kill the process.
  - Report: This process is terminated when either of the cars reach 100 steps. The positions of the cars are accessed using fd_car1_pos and fd_car2_pos as explained above.

- **How many processes are you creating?**
  - I have created 4 processes, namely-
    - cheatmode
    - car1
    - car2
    - report