OS Lab 4

B19CSE071
Ridhima Kohli

How it works?

Generator file - Runs the process.c file 4 times in order to generate 4 such processes out of which 2 are CPU bound

Process.c sends the pid of processes along with arguments sleepingTime , priority , probability and number Of Iteration(Rounds) to Scheduler.c which schedules accordingly

Threads ,sockets and message queues are used for ready queue management ,IPC and argument passing.

Results have been renamed as RR.txt and PR.txt

How to run

First compile all the files as :

```
gcc scheduler.c -o s -lpthread

gcc process.c -o p

gcc generator.c -o g
```

Now open two terminals and run these commands:

```
./s RR ( for round robin)

./s PR (for priority)

./g
```

Sometimes the programs might hang in between , for that we need to restart PC. Reason is unknown

Also make sure to change address of CommandToRun in generator.c according to PC folders

What are RR and PR

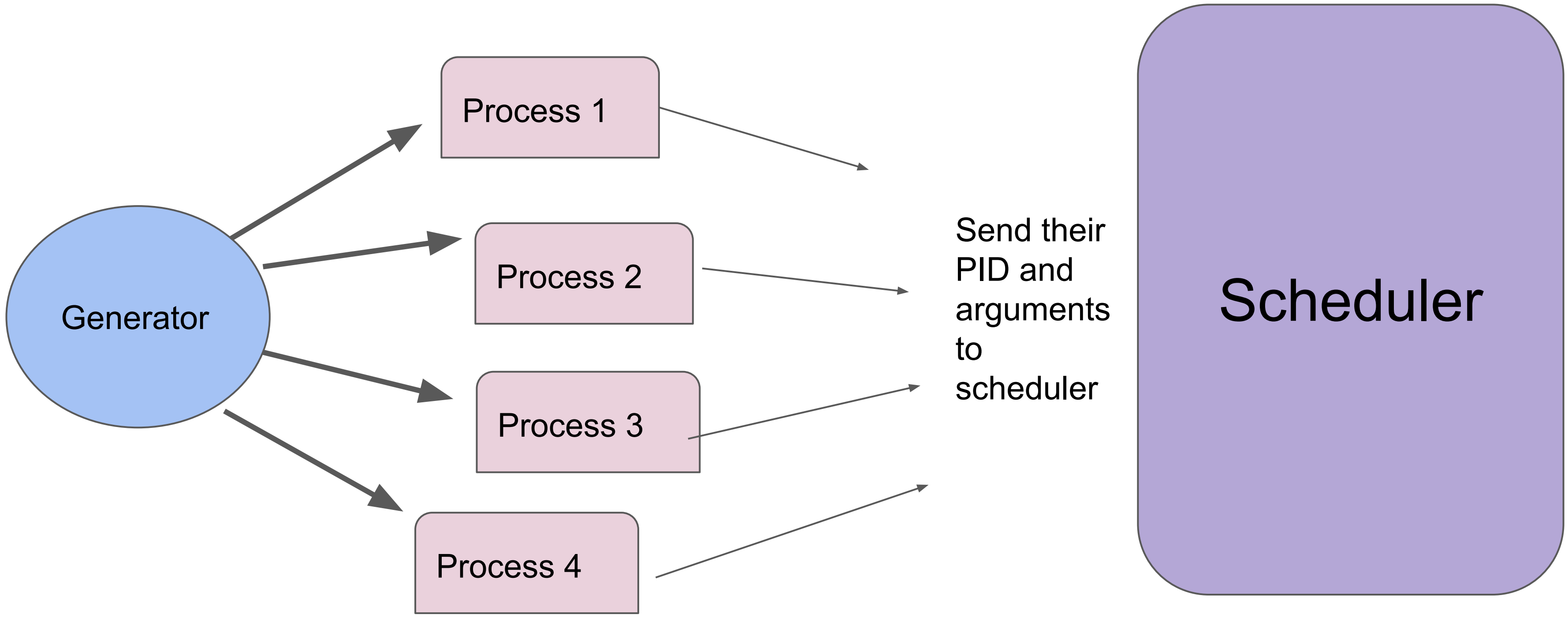
RR - Round Robin Process Scheduling

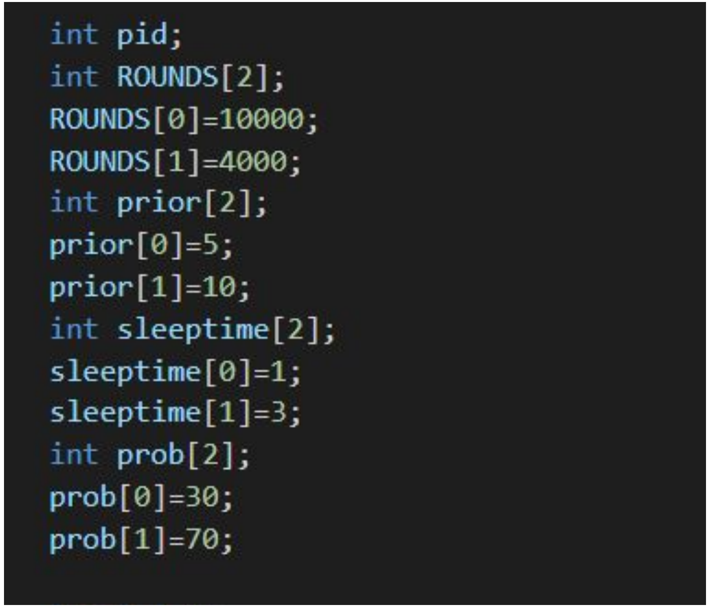
Allocates quantum time to every process for execution

PR - Priority Based Round Robin Process Scheduling

Gets process with order of maximum priority and executes them based on round robin

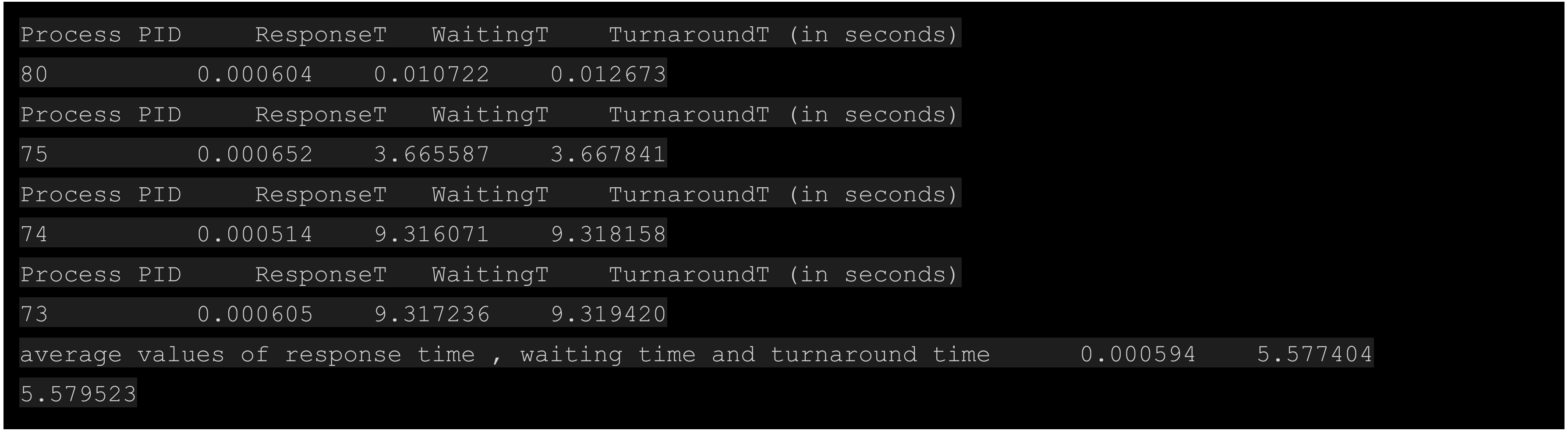
Diagram of working of the three files





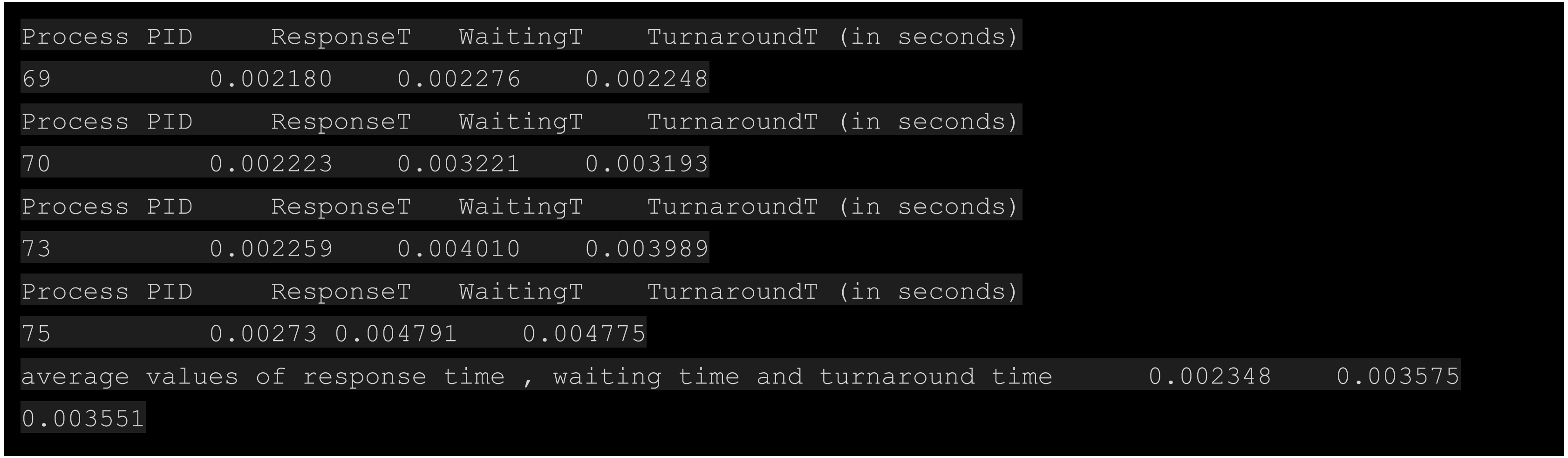
```
int pid;
int ROUNDS[2];
ROUNDS[0]=10000;
ROUNDS[1]=4000;
int prior[2];
prior[0]=5;
prior[1]=10;
int sleeptime[2];
sleeptime[0]=1;
sleeptime[1]=3;
int prob[2];
prob[0]=30;
prob[1]=70;
```

Output for Round Robin Scheduling



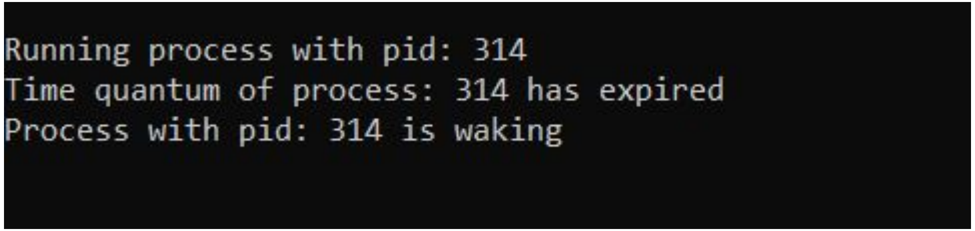
```
Process PID      ResponseT   WaitingT    TurnaroundT (in seconds)
80          0.000604    0.010722    0.012673
Process PID      ResponseT   WaitingT    TurnaroundT (in seconds)
75          0.000652    3.665587    3.667841
Process PID      ResponseT   WaitingT    TurnaroundT (in seconds)
74          0.000514    9.316071    9.318158
Process PID      ResponseT   WaitingT    TurnaroundT (in seconds)
73          0.000605    9.317236    9.319420
average values of response time , waiting time and turnaround time      0.000594    5.577404
5.579523
```

Output for Priority Round Robin Scheduling

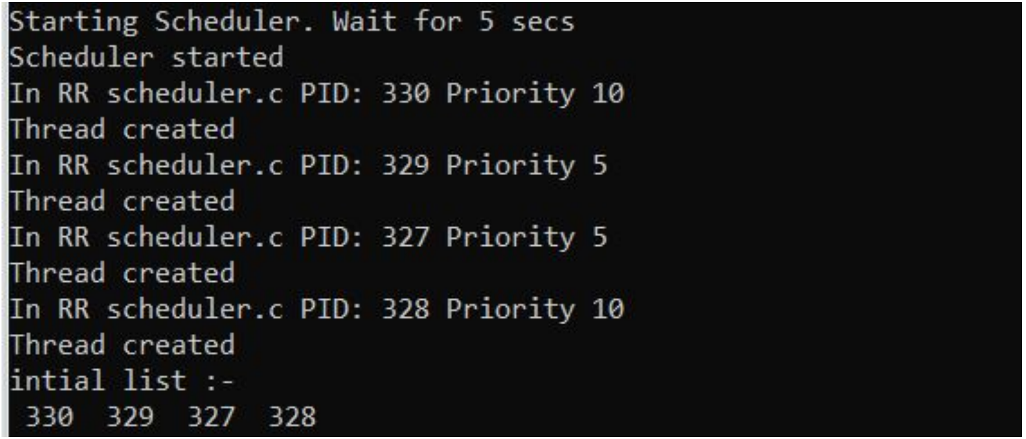


```
Process PID     ResponseT   WaitingT    TurnaroundT (in seconds)
69          0.002180    0.002276    0.002248
Process PID     ResponseT   WaitingT    TurnaroundT (in seconds)
70          0.002223    0.003221    0.003193
Process PID     ResponseT   WaitingT    TurnaroundT (in seconds)
73          0.002259    0.004010    0.003989
Process PID     ResponseT   WaitingT    TurnaroundT (in seconds)
75          0.00273 0.004791    0.004775
average values of response time , waiting time and turnaround time      0.002348    0.003575
0.003551
```

Scheduler



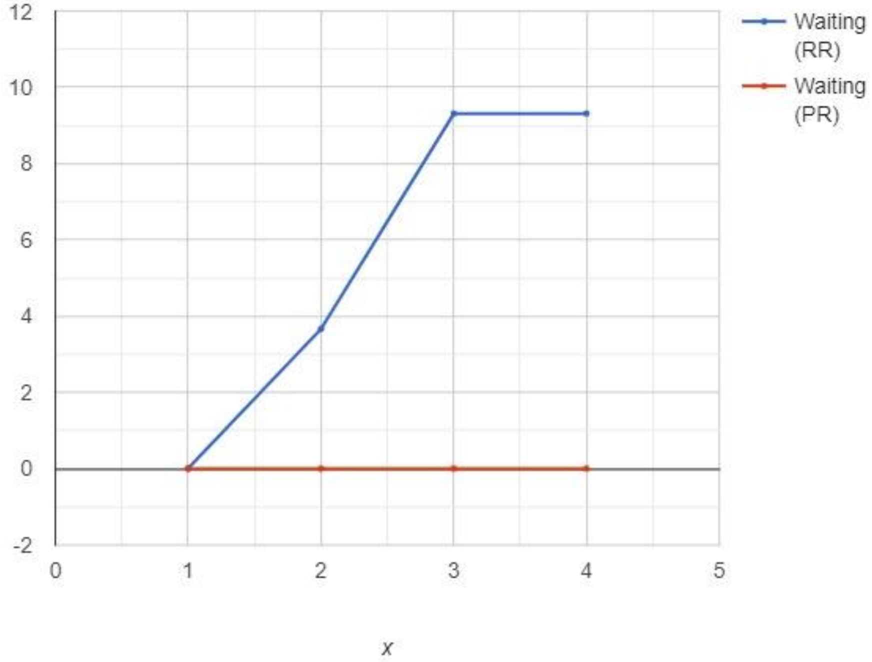
```
Running process with pid: 314
Time quantum of process: 314 has expired
Process with pid: 314 is waking
```



```
Starting Scheduler. Wait for 5 secs
Scheduler started
In RR scheduler.c PID: 330 Priority 10
Thread created
In RR scheduler.c PID: 329 Priority 5
Thread created
In RR scheduler.c PID: 327 Priority 5
Thread created
In RR scheduler.c PID: 328 Priority 10
Thread created
intial list :-
 330  329  327  328
```

Comparison

Parameter(avg)	Round Robin	Priority RR	difference
Waiting Time	5.577404	0.003575	5.573829
Turnaround Time	5.579523	0.003551	5.575972



Observations

The waiting time and turnaround in RR is much higher than PR

The response time in PR is higher

RR has more number of context switches

PR considers Priority