COURSEWORK-ASSIGNMENT

Name – Ashutosh Soni

Id – 2018ucp1505

Question: Write a report on

(1) new functions you have learned and a description of the function

(2) summary of what each program does

(3) Results and inferences from execution of the program

a) Process Intro (process-run.py)

Answer: This program allows us to see how the states of a process states are changing as it runs on a CPU. As we know that there are several states which process may take, they are:

- RUNNING: The process which is using the CPU
- READY: The process which is in ready Queue and waiting to Use CPU
- WAITING: The process is waiting for some I/O.
- DONE: The process had completed its execution.

The Program process- run.py can takes several arguments via command line before starting execution…

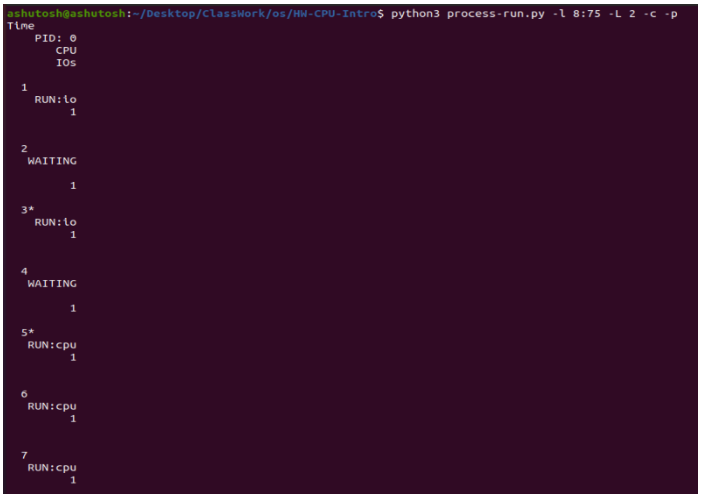
They are

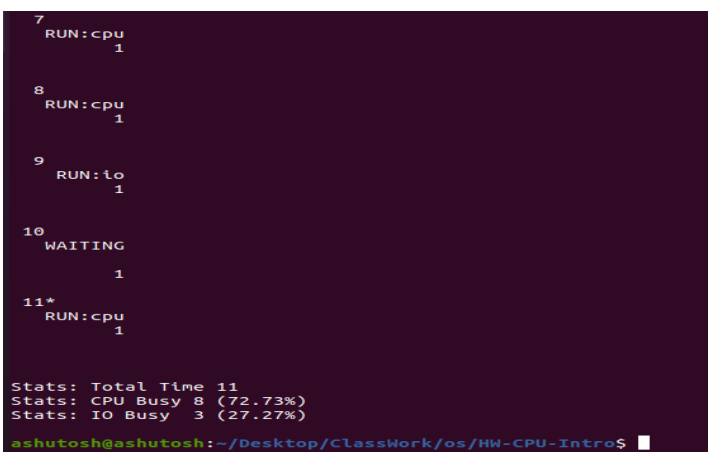
- **l: PROCESS_LIST**
  - a comma-separated list of processes to run, in the form X1:Y1, X2:Y2... where X is the number of instructions that process should run, and Y the chances (from 0 to 100) that an instruction will use the CPU or issue an IO
- **L: IO_LENGTH --iolength=IO_LENGTH**
  - How long can I/O take?
- **-S PROCESS_SWITCH_BEHAVIOR, --switch=PROCESS_SWITCH_BEHAVIOR**
  - when to switch between processes: SWITCH_ON_IO, SWITCH_ON_END
- **-I IO_DONE_BEHAVIOR, --iodone=IO_DONE_BEHAVIOR**
  - type of behaviour when IO ends: IO_RUN_LATER, IO_RUN_IMMEDIATE
- **-c**
  - compute answers for me
- **-p, --print stats**
  - print statistics at end; only useful with -c flag (otherwise, stats are not printed)

Let's run a process having 8 instructions ,75 out of 100 chances that it can use CPU or issue an I/O and IO length as 2 and computes its answer and stats for this we have to use command:

python3 process-run.py -l 8:75 -L 2 -c -p

The output I got is:



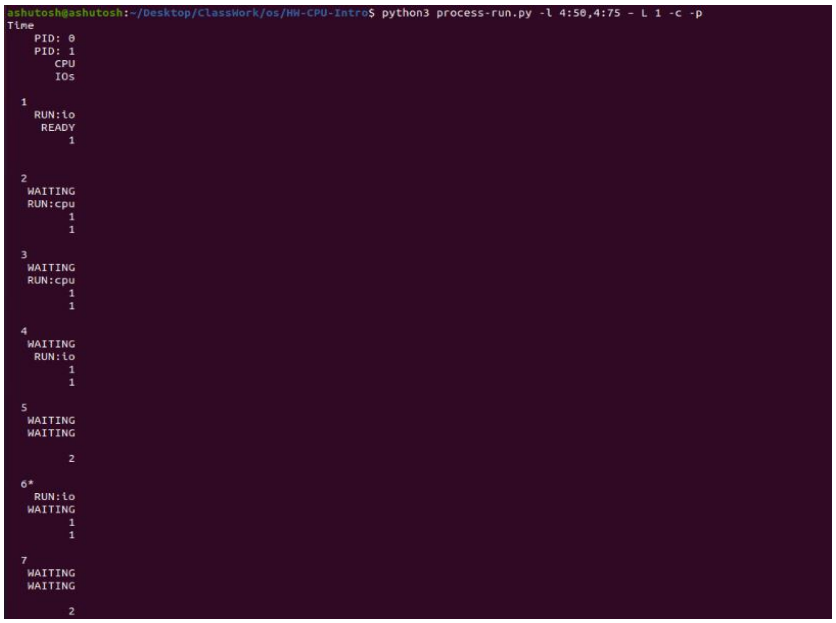


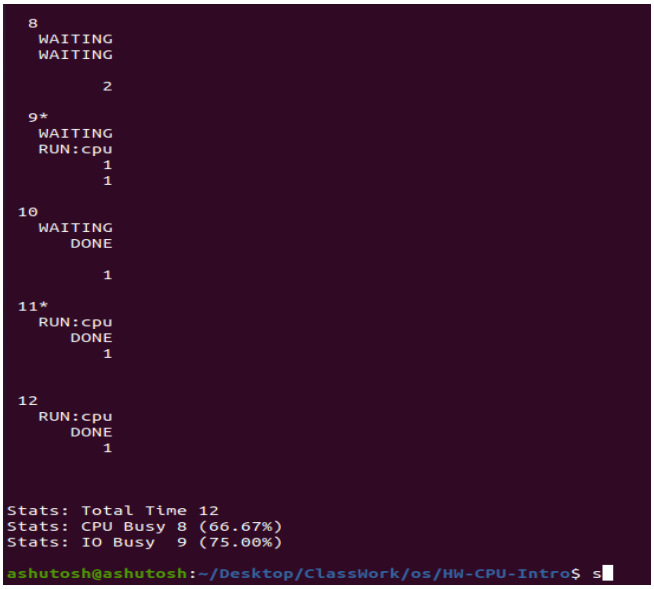
In the above the first line shows clock tick and then the process state and then number if CPU is used otherwise blank and then in next line it shows number if I/O is used otherwise black and have 1 line break.

Now Let's run 2 process simultaneously having 4 instructions each and one have 50 by 100 chances and other have 75 by 100 chances that it can use CPU or issue an I/O and have IO length of 1 for each process. Let's compute its answer and stats, for this our command be like…

python3 process-run.py -l 4:50,4:75 – L 1 -c -p

The output I got is:





b) Process API

Answer: In this there are system calls like fork (), wait (), exec () is programmed and we have to analyse accordingly the process that are running and some other experiments like passing variable and getting that are they passed successfully or not in this there are some programs written in c and we have executed them all accordingly to given guidelines this is written in C language.

c) Scheduling Basics (scheduler.py)

Answer: This program gives us platform to analyse different process under different scheduling metrics such as response time, turnaround time, and wait time per process as well as the average response time, average turnaround time and average wait time for all the given process.

We can send via arguments some arguments such the type of scheduling, number of process or job, random seed number, the length of the process and the maximum length of the process…

-p <type of scheduling> -j <number of jobs> -l <job-length of each job> -c

c is used at last for displaying the information of job...

Let's us suppose we have 4 job of length 10,8,14,5 ……

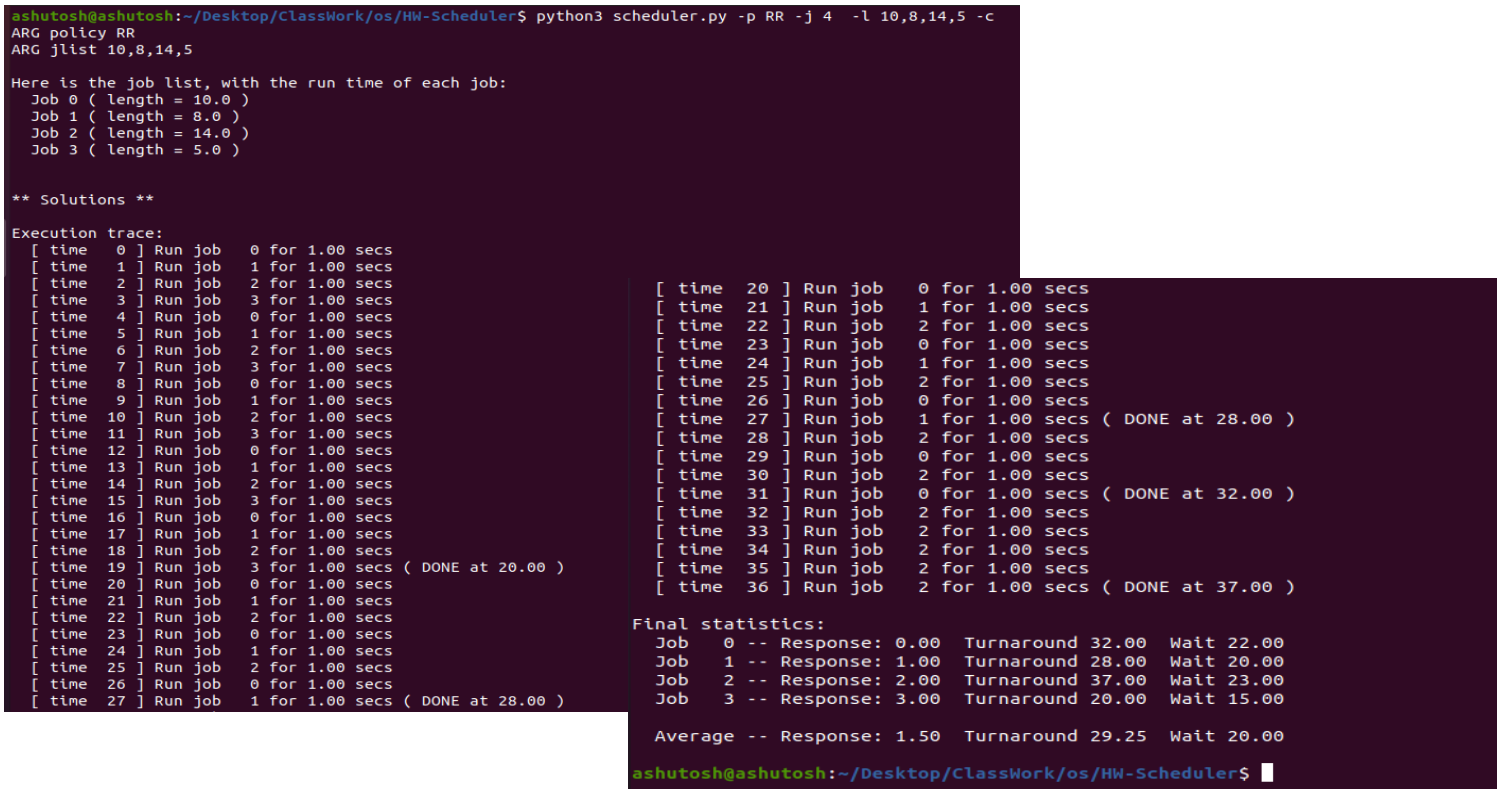
- Using First Come First Serve FCFS (FIFO) scheduling policy

```
ashutosh@ashutosh:~/Desktop/ClassWork/os/HW-Scheduler$ python3 scheduler.py -p FIFO -j 4  -l 10,8,14,5 -c
ARG policy FIFO
ARG jlist 10,8,14,5

Here is the job list, with the run time of each job:
  Job 0 ( length = 10.0 )
  Job 1 ( length = 8.0 )
  Job 2 ( length = 14.0 )
  Job 3 ( length = 5.0 )


** Solutions **

Execution trace:
  [ time   0 ] Run job 0 for 10.00 secs ( DONE at 10.00 )
  [ time  10 ] Run job 1 for 8.00 secs ( DONE at 18.00 )
  [ time  18 ] Run job 2 for 14.00 secs ( DONE at 32.00 )
  [ time  32 ] Run job 3 for 5.00 secs ( DONE at 37.00 )

Final statistics:
  Job   0 -- Response: 0.00  Turnaround 10.00  Wait 0.00
  Job   1 -- Response: 10.00  Turnaround 18.00  Wait 10.00
  Job   2 -- Response: 18.00  Turnaround 32.00  Wait 18.00
  Job   3 -- Response: 32.00  Turnaround 37.00  Wait 32.00

  Average -- Response: 15.00  Turnaround 24.25  Wait 15.00

ashutosh@ashutosh:~/Desktop/ClassWork/os/HW-Scheduler$
```

- Using Short Job First SJF scheduling policy

```
ashutosh@ashutosh:~/Desktop/ClassWork/os/HW-Scheduler$ python3 scheduler.py -p SJF -j 4  -l 10,8,14,5 -c
ARG policy SJF
ARG jlist 10,8,14,5

Here is the job list, with the run time of each job:
  Job 0 ( length = 10.0 )
  Job 1 ( length = 8.0 )
  Job 2 ( length = 14.0 )
  Job 3 ( length = 5.0 )


** Solutions **

Execution trace:
  [ time   0 ] Run job 3 for 5.00 secs ( DONE at 5.00 )
  [ time   5 ] Run job 1 for 8.00 secs ( DONE at 13.00 )
  [ time  13 ] Run job 0 for 10.00 secs ( DONE at 23.00 )
  [ time  23 ] Run job 2 for 14.00 secs ( DONE at 37.00 )

Final statistics:
  Job   3 -- Response: 0.00  Turnaround 5.00  Wait 0.00
  Job   1 -- Response: 5.00  Turnaround 13.00  Wait 5.00
  Job   0 -- Response: 13.00  Turnaround 23.00  Wait 13.00
  Job   2 -- Response: 23.00  Turnaround 37.00  Wait 23.00

  Average -- Response: 10.25  Turnaround 19.50  Wait 10.25

ashutosh@ashutosh:~/Desktop/ClassWork/os/HW-Scheduler$
```

- Using Round Robin RR scheduling policy



d) MLFQ Scheduling (mlfq.py)

Answer: This program is an implementation of Multi Level Feedback Queue scheduling. This program chooses the job according to their priority and then lead to job done. In this type of scheduling approach there are multiple level of queue s and used feedback to determine the priority of a given job. History is its guide: pay attention to how job behave over time and treat them accordingly.

There are some rules related with it:

- Rule 1: If Priority(A) > Priority(B), A runs (B doesn't).
- Rule 2: If Priority(A) = Priority(B), A & B run in round-robin fashion using the time slice (quantum length) of the given queue.
- Rule 3: When a job enters the system, it is placed at the highest priority (the topmost queue).
- Rule 4: Once a job uses up its time allotment at a given level (regardless of how many times it has given up the CPU), its priority is reduced (i.e., it moves down one queue).
- Rule 5: After some time

Some important terms which are used to give argument via command for this program are ….

- -s SEED, --seed = SEED the random seed
- n NUMQUEUES, --numQueues=NUMQUEUES number of queues in MLFQ
- q QUANTUM --quantum= QUANTUM length of time slice
- Q QUANTUMLIST, --quantumList=QUANTUMLIST length of time slice per queue level,
  Specified as x, y, z, … where x is the quantum length for the highest- priority
  Queue, y the next highest and so forth.
- -j NUMJOBS, --numJobs=NUMJOBS
  number of jobs in the system
- -m MAXLEN, --maxlen=MAXLEN
  max run-time of a job (if random)
- -M MAXIO, --maxio=MAXIO
  max I/O frequency of a job (if random)

- -B BOOST, --boost=BOOST
  how often to boost the priority of all?
  jobs back to high priority (0 means never)
- -i IOTIME, --iotime=IOTIME
  how long can I/O should last (fixed constant)
- -S, --stay reset and stay at same priority level
  when issuing I/O
- -l JLIST, --jlist=JLIST
  a comma-separated list of jobs to run,
  in the form x1, y1, z1:x2, y2, z2... where
  x is start time, y is run time, and z
  is how often the job issues and I/O request
- -c compute answers for me

Lets the quantum length of queue is 4, 5, 6 respectively and I/O time is 1 and c for describing how it runs. The command for that will be...

python3 mlfq.py -Q 4,5,6 -j 2 -i 1 -c

The output is ….

```
ashutosh@ashutosh:~/Desktop/ClassWork/os/HW-MLFQ$ python3 mlfq.py -Q 4,5,6 -j 2 -i 1 -c
Here is the list of inputs:
OPTIONS jobs 2
OPTIONS queues 3
OPTIONS allotments for queue  2 is   1
OPTIONS quantum length for queue  2 is   4
OPTIONS allotments for queue  1 is   1
OPTIONS quantum length for queue  1 is   5
OPTIONS allotments for queue  0 is   1
OPTIONS quantum length for queue  0 is   6
OPTIONS boost 0
OPTIONS ioTime 1
OPTIONS stayAfterIO False
OPTIONS iobump False


For each job, three defining characteristics are given:
  startTime : at what time does the job enter the system
  runTime   : the total CPU time needed by the job to finish
  ioFreq    : every ioFreq time units, the job issues an I/O
              (the I/O takes ioTime units to complete)

Job List:
  Job  0: startTime   0 - runTime  84 - ioFreq   7
  Job  1: startTime   0 - runTime  42 - ioFreq   3


Execution Trace:

[ time 0 ] JOB BEGINS by JOB 0
[ time 0 ] JOB BEGINS by JOB 1
[ time 0 ] Run JOB 0 at PRIORITY 2 [ TICKS 3 ALLOT 1 TIME 83 (of 84) ]
[ time 1 ] Run JOB 0 at PRIORITY 2 [ TICKS 2 ALLOT 1 TIME 82 (of 84) ]
[ time 2 ] Run JOB 0 at PRIORITY 2 [ TICKS 1 ALLOT 1 TIME 81 (of 84) ]
[ time 3 ] Run JOB 0 at PRIORITY 2 [ TICKS 0 ALLOT 1 TIME 80 (of 84) ]
[ time 4 ] Run JOB 1 at PRIORITY 2 [ TICKS 3 ALLOT 1 TIME 41 (of 42) ]
[ time 5 ] Run JOB 1 at PRIORITY 2 [ TICKS 2 ALLOT 1 TIME 40 (of 42) ]
[ time 6 ] Run JOB 1 at PRIORITY 2 [ TICKS 1 ALLOT 1 TIME 39 (of 42) ]
[ time 7 ] IO_START by JOB 1
IO DONE
[ time 7 ] Run JOB 0 at PRIORITY 1 [ TICKS 4 ALLOT 1 TIME 79 (of 84) ]
[ time 8 ] IO_DONE by JOB 1
[ time 8 ] Run JOB 1 at PRIORITY 2 [ TICKS 0 ALLOT 1 TIME 38 (of 42) ]
[ time 9 ] Run JOB 0 at PRIORITY 1 [ TICKS 3 ALLOT 1 TIME 78 (of 84) ]
[ time 10 ] Run JOB 0 at PRIORITY 1 [ TICKS 2 ALLOT 1 TIME 77 (of 84) ]
[ time 11 ] IO_START by JOB 0
IO DONE
```

```
[ time 11 ] IO_START by JOB 0
IO DONE
[ time 11 ] Run JOB 1 at PRIORITY 1 [ TICKS 4 ALLOT 1 TIME 37 (of 42) ]
[ time 12 ] IO_DONE by JOB 0
[ time 12 ] Run JOB 1 at PRIORITY 1 [ TICKS 3 ALLOT 1 TIME 36 (of 42) ]
[ time 13 ] IO_START by JOB 1
IO DONE
[ time 13 ] Run JOB 0 at PRIORITY 1 [ TICKS 1 ALLOT 1 TIME 76 (of 84) ]
[ time 14 ] IO_DONE by JOB 1
[ time 14 ] Run JOB 0 at PRIORITY 1 [ TICKS 0 ALLOT 1 TIME 75 (of 84) ]
[ time 15 ] Run JOB 1 at PRIORITY 1 [ TICKS 2 ALLOT 1 TIME 35 (of 42) ]
[ time 16 ] Run JOB 1 at PRIORITY 1 [ TICKS 1 ALLOT 1 TIME 34 (of 42) ]
[ time 17 ] Run JOB 1 at PRIORITY 1 [ TICKS 0 ALLOT 1 TIME 33 (of 42) ]
[ time 18 ] IO_START by JOB 1
IO DONE
[ time 18 ] Run JOB 0 at PRIORITY 0 [ TICKS 5 ALLOT 1 TIME 74 (of 84) ]
[ time 19 ] IO_DONE by JOB 1
[ time 19 ] Run JOB 0 at PRIORITY 0 [ TICKS 4 ALLOT 1 TIME 73 (of 84) ]
[ time 20 ] Run JOB 0 at PRIORITY 0 [ TICKS 3 ALLOT 1 TIME 72 (of 84) ]
[ time 21 ] Run JOB 0 at PRIORITY 0 [ TICKS 2 ALLOT 1 TIME 71 (of 84) ]
[ time 22 ] Run JOB 0 at PRIORITY 0 [ TICKS 1 ALLOT 1 TIME 70 (of 84) ]
[ time 23 ] IO_START by JOB 0
IO DONE
[ time 23 ] Run JOB 1 at PRIORITY 0 [ TICKS 5 ALLOT 1 TIME 32 (of 42) ]
[ time 24 ] IO_DONE by JOB 0
[ time 24 ] Run JOB 1 at PRIORITY 0 [ TICKS 4 ALLOT 1 TIME 31 (of 42) ]
[ time 25 ] Run JOB 1 at PRIORITY 0 [ TICKS 3 ALLOT 1 TIME 30 (of 42) ]
[ time 26 ] IO_START by JOB 1
IO DONE
[ time 26 ] Run JOB 0 at PRIORITY 0 [ TICKS 0 ALLOT 1 TIME 69 (of 84) ]
[ time 27 ] IO_DONE by JOB 1
[ time 27 ] Run JOB 0 at PRIORITY 0 [ TICKS 5 ALLOT 1 TIME 68 (of 84) ]
[ time 28 ] Run JOB 0 at PRIORITY 0 [ TICKS 4 ALLOT 1 TIME 67 (of 84) ]
[ time 29 ] Run JOB 0 at PRIORITY 0 [ TICKS 3 ALLOT 1 TIME 66 (of 84) ]
[ time 30 ] Run JOB 0 at PRIORITY 0 [ TICKS 2 ALLOT 1 TIME 65 (of 84) ]
[ time 31 ] Run JOB 0 at PRIORITY 0 [ TICKS 1 ALLOT 1 TIME 64 (of 84) ]
[ time 32 ] Run JOB 0 at PRIORITY 0 [ TICKS 0 ALLOT 1 TIME 63 (of 84) ]
[ time 33 ] IO_START by JOB 0
IO DONE
[ time 33 ] Run JOB 1 at PRIORITY 0 [ TICKS 2 ALLOT 1 TIME 29 (of 42) ]
[ time 34 ] IO_DONE by JOB 0
[ time 34 ] Run JOB 1 at PRIORITY 0 [ TICKS 1 ALLOT 1 TIME 28 (of 42) ]
[ time 35 ] Run JOB 1 at PRIORITY 0 [ TICKS 0 ALLOT 1 TIME 27 (of 42) ]
[ time 36 ] IO_START by JOB 1
IO DONE
```

```
[ time 36 ] IO_START by JOB 1
IO DONE
[ time 36 ] Run JOB 0 at PRIORITY 0 [ TICKS 5 ALLOT 1 TIME 62 (of 84) ]
[ time 37 ] IO_DONE by JOB 1
[ time 37 ] Run JOB 0 at PRIORITY 0 [ TICKS 4 ALLOT 1 TIME 61 (of 84) ]
[ time 38 ] Run JOB 0 at PRIORITY 0 [ TICKS 3 ALLOT 1 TIME 60 (of 84) ]
[ time 39 ] Run JOB 0 at PRIORITY 0 [ TICKS 2 ALLOT 1 TIME 59 (of 84) ]
[ time 40 ] Run JOB 0 at PRIORITY 0 [ TICKS 1 ALLOT 1 TIME 58 (of 84) ]
[ time 41 ] Run JOB 0 at PRIORITY 0 [ TICKS 0 ALLOT 1 TIME 57 (of 84) ]
[ time 42 ] Run JOB 1 at PRIORITY 0 [ TICKS 5 ALLOT 1 TIME 26 (of 42) ]
[ time 43 ] Run JOB 1 at PRIORITY 0 [ TICKS 4 ALLOT 1 TIME 25 (of 42) ]
[ time 44 ] Run JOB 1 at PRIORITY 0 [ TICKS 3 ALLOT 1 TIME 24 (of 42) ]
[ time 45 ] IO_START by JOB 1
IO DONE
[ time 45 ] Run JOB 0 at PRIORITY 0 [ TICKS 5 ALLOT 1 TIME 56 (of 84) ]
[ time 46 ] IO_START by JOB 0
IO DONE
[ time 46 ] IO_DONE by JOB 1
[ time 46 ] Run JOB 1 at PRIORITY 0 [ TICKS 2 ALLOT 1 TIME 23 (of 42) ]
[ time 47 ] IO_DONE by JOB 0
[ time 47 ] Run JOB 1 at PRIORITY 0 [ TICKS 1 ALLOT 1 TIME 22 (of 42) ]
[ time 48 ] Run JOB 1 at PRIORITY 0 [ TICKS 0 ALLOT 1 TIME 21 (of 42) ]
[ time 49 ] IO_START by JOB 1
IO DONE
[ time 49 ] Run JOB 0 at PRIORITY 0 [ TICKS 4 ALLOT 1 TIME 55 (of 84) ]
[ time 50 ] IO_DONE by JOB 1
[ time 50 ] Run JOB 0 at PRIORITY 0 [ TICKS 3 ALLOT 1 TIME 54 (of 84) ]
[ time 51 ] Run JOB 0 at PRIORITY 0 [ TICKS 2 ALLOT 1 TIME 53 (of 84) ]
[ time 52 ] Run JOB 0 at PRIORITY 0 [ TICKS 1 ALLOT 1 TIME 52 (of 84) ]
[ time 53 ] Run JOB 0 at PRIORITY 0 [ TICKS 0 ALLOT 1 TIME 51 (of 84) ]
[ time 54 ] Run JOB 1 at PRIORITY 0 [ TICKS 5 ALLOT 1 TIME 20 (of 42) ]
[ time 55 ] Run JOB 1 at PRIORITY 0 [ TICKS 4 ALLOT 1 TIME 19 (of 42) ]
[ time 56 ] Run JOB 1 at PRIORITY 0 [ TICKS 3 ALLOT 1 TIME 18 (of 42) ]
[ time 57 ] IO_START by JOB 1
IO DONE
[ time 57 ] Run JOB 0 at PRIORITY 0 [ TICKS 5 ALLOT 1 TIME 50 (of 84) ]
[ time 58 ] IO_DONE by JOB 1
[ time 58 ] Run JOB 0 at PRIORITY 0 [ TICKS 4 ALLOT 1 TIME 49 (of 84) ]
[ time 59 ] IO_START by JOB 0
IO DONE
```

```
[ time 59 ] Run JOB 1 at PRIORITY 0 [ TICKS 2 ALLOT 1 TIME 17 (of 42) ]
[ time 60 ] IO_DONE by JOB 0
[ time 60 ] Run JOB 1 at PRIORITY 0 [ TICKS 1 ALLOT 1 TIME 16 (of 42) ]
[ time 61 ] Run JOB 1 at PRIORITY 0 [ TICKS 0 ALLOT 1 TIME 15 (of 42) ]
[ time 62 ] IO_START by JOB 1
IO DONE
[ time 62 ] Run JOB 0 at PRIORITY 0 [ TICKS 3 ALLOT 1 TIME 48 (of 84) ]
[ time 63 ] IO_DONE by JOB 1
[ time 63 ] Run JOB 0 at PRIORITY 0 [ TICKS 2 ALLOT 1 TIME 47 (of 84) ]
[ time 64 ] Run JOB 0 at PRIORITY 0 [ TICKS 1 ALLOT 1 TIME 46 (of 84) ]
[ time 65 ] Run JOB 0 at PRIORITY 0 [ TICKS 0 ALLOT 1 TIME 45 (of 84) ]
[ time 66 ] Run JOB 1 at PRIORITY 0 [ TICKS 5 ALLOT 1 TIME 14 (of 42) ]
[ time 67 ] Run JOB 1 at PRIORITY 0 [ TICKS 4 ALLOT 1 TIME 13 (of 42) ]
[ time 68 ] Run JOB 1 at PRIORITY 0 [ TICKS 3 ALLOT 1 TIME 12 (of 42) ]
[ time 69 ] IO_START by JOB 1
IO DONE
[ time 69 ] Run JOB 0 at PRIORITY 0 [ TICKS 5 ALLOT 1 TIME 44 (of 84) ]
[ time 70 ] IO_DONE by JOB 1
[ time 70 ] Run JOB 0 at PRIORITY 0 [ TICKS 4 ALLOT 1 TIME 43 (of 84) ]
[ time 71 ] Run JOB 0 at PRIORITY 0 [ TICKS 3 ALLOT 1 TIME 42 (of 84) ]
[ time 72 ] IO_START by JOB 0
IO DONE
[ time 72 ] Run JOB 1 at PRIORITY 0 [ TICKS 2 ALLOT 1 TIME 11 (of 42) ]
[ time 73 ] IO_DONE by JOB 0
[ time 73 ] Run JOB 1 at PRIORITY 0 [ TICKS 1 ALLOT 1 TIME 10 (of 42) ]
[ time 74 ] Run JOB 1 at PRIORITY 0 [ TICKS 0 ALLOT 1 TIME 9 (of 42) ]
[ time 75 ] IO_START by JOB 1
IO DONE
[ time 75 ] Run JOB 0 at PRIORITY 0 [ TICKS 2 ALLOT 1 TIME 41 (of 84) ]
[ time 76 ] IO_DONE by JOB 1
[ time 76 ] Run JOB 0 at PRIORITY 0 [ TICKS 1 ALLOT 1 TIME 40 (of 84) ]
[ time 77 ] Run JOB 0 at PRIORITY 0 [ TICKS 0 ALLOT 1 TIME 39 (of 84) ]
[ time 78 ] Run JOB 1 at PRIORITY 0 [ TICKS 5 ALLOT 1 TIME 8 (of 42) ]
[ time 79 ] Run JOB 1 at PRIORITY 0 [ TICKS 4 ALLOT 1 TIME 7 (of 42) ]
[ time 80 ] Run JOB 1 at PRIORITY 0 [ TICKS 3 ALLOT 1 TIME 6 (of 42) ]
[ time 81 ] IO_START by JOB 1
IO DONE
[ time 81 ] Run JOB 0 at PRIORITY 0 [ TICKS 5 ALLOT 1 TIME 38 (of 84) ]
[ time 82 ] IO_DONE by JOB 1
[ time 82 ] Run JOB 0 at PRIORITY 0 [ TICKS 4 ALLOT 1 TIME 37 (of 84) ]
[ time 83 ] Run JOB 0 at PRIORITY 0 [ TICKS 3 ALLOT 1 TIME 36 (of 84) ]
[ time 84 ] Run JOB 0 at PRIORITY 0 [ TICKS 2 ALLOT 1 TIME 35 (of 84) ]
[ time 85 ] IO_START by JOB 0
IO DONE
```

```
[ time 85 ] IO_START by JOB 0
IO DONE
[ time 85 ] Run JOB 1 at PRIORITY 0 [ TICKS 2 ALLOT 1 TIME 5 (of 42) ]
[ time 86 ] IO_DONE by JOB 0
[ time 86 ] Run JOB 1 at PRIORITY 0 [ TICKS 1 ALLOT 1 TIME 4 (of 42) ]
[ time 87 ] Run JOB 1 at PRIORITY 0 [ TICKS 0 ALLOT 1 TIME 3 (of 42) ]
[ time 88 ] IO_START by JOB 1
IO DONE
[ time 88 ] Run JOB 0 at PRIORITY 0 [ TICKS 1 ALLOT 1 TIME 34 (of 84) ]
[ time 89 ] IO_DONE by JOB 1
[ time 89 ] Run JOB 0 at PRIORITY 0 [ TICKS 0 ALLOT 1 TIME 33 (of 84) ]
[ time 90 ] Run JOB 1 at PRIORITY 0 [ TICKS 5 ALLOT 1 TIME 2 (of 42) ]
[ time 91 ] Run JOB 1 at PRIORITY 0 [ TICKS 4 ALLOT 1 TIME 1 (of 42) ]
[ time 92 ] Run JOB 1 at PRIORITY 0 [ TICKS 3 ALLOT 1 TIME 0 (of 42) ]
[ time 93 ] FINISHED JOB 1
[ time 93 ] Run JOB 0 at PRIORITY 0 [ TICKS 5 ALLOT 1 TIME 32 (of 84) ]
[ time 94 ] Run JOB 0 at PRIORITY 0 [ TICKS 4 ALLOT 1 TIME 31 (of 84) ]
[ time 95 ] Run JOB 0 at PRIORITY 0 [ TICKS 3 ALLOT 1 TIME 30 (of 84) ]
[ time 96 ] Run JOB 0 at PRIORITY 0 [ TICKS 2 ALLOT 1 TIME 29 (of 84) ]
[ time 97 ] Run JOB 0 at PRIORITY 0 [ TICKS 1 ALLOT 1 TIME 28 (of 84) ]
[ time 98 ] IO_START by JOB 0
IO DONE
[ time 98 ] IDLE
[ time 99 ] IO_DONE by JOB 0
[ time 99 ] Run JOB 0 at PRIORITY 0 [ TICKS 0 ALLOT 1 TIME 27 (of 84) ]
[ time 100 ] Run JOB 0 at PRIORITY 0 [ TICKS 5 ALLOT 1 TIME 26 (of 84) ]
[ time 101 ] Run JOB 0 at PRIORITY 0 [ TICKS 4 ALLOT 1 TIME 25 (of 84) ]
[ time 102 ] Run JOB 0 at PRIORITY 0 [ TICKS 3 ALLOT 1 TIME 24 (of 84) ]
[ time 103 ] Run JOB 0 at PRIORITY 0 [ TICKS 2 ALLOT 1 TIME 23 (of 84) ]
[ time 104 ] Run JOB 0 at PRIORITY 0 [ TICKS 1 ALLOT 1 TIME 22 (of 84) ]
[ time 105 ] Run JOB 0 at PRIORITY 0 [ TICKS 0 ALLOT 1 TIME 21 (of 84) ]
[ time 106 ] IO_START by JOB 0
IO DONE
[ time 106 ] IDLE
[ time 107 ] IO_DONE by JOB 0
[ time 107 ] Run JOB 0 at PRIORITY 0 [ TICKS 5 ALLOT 1 TIME 20 (of 84) ]
[ time 108 ] Run JOB 0 at PRIORITY 0 [ TICKS 4 ALLOT 1 TIME 19 (of 84) ]
[ time 109 ] Run JOB 0 at PRIORITY 0 [ TICKS 3 ALLOT 1 TIME 18 (of 84) ]
[ time 110 ] Run JOB 0 at PRIORITY 0 [ TICKS 2 ALLOT 1 TIME 17 (of 84) ]
[ time 111 ] Run JOB 0 at PRIORITY 0 [ TICKS 1 ALLOT 1 TIME 16 (of 84) ]
[ time 112 ] Run JOB 0 at PRIORITY 0 [ TICKS 0 ALLOT 1 TIME 15 (of 84) ]
[ time 113 ] Run JOB 0 at PRIORITY 0 [ TICKS 5 ALLOT 1 TIME 14 (of 84) ]
[ time 114 ] IO_START by JOB 0
IO DONE
```

```
IO DONE
[ time 114 ] IDLE
[ time 115 ] IO_DONE by JOB 0
[ time 115 ] Run JOB 0 at PRIORITY 0 [ TICKS 4 ALLOT 1 TIME 13 (of 84) ]
[ time 116 ] Run JOB 0 at PRIORITY 0 [ TICKS 3 ALLOT 1 TIME 12 (of 84) ]
[ time 117 ] Run JOB 0 at PRIORITY 0 [ TICKS 2 ALLOT 1 TIME 11 (of 84) ]
[ time 118 ] Run JOB 0 at PRIORITY 0 [ TICKS 1 ALLOT 1 TIME 10 (of 84) ]
[ time 119 ] Run JOB 0 at PRIORITY 0 [ TICKS 0 ALLOT 1 TIME 9 (of 84) ]
[ time 120 ] Run JOB 0 at PRIORITY 0 [ TICKS 5 ALLOT 1 TIME 8 (of 84) ]
[ time 121 ] Run JOB 0 at PRIORITY 0 [ TICKS 4 ALLOT 1 TIME 7 (of 84) ]
[ time 122 ] IO_START by JOB 0
IO DONE
[ time 122 ] IDLE
[ time 123 ] IO_DONE by JOB 0
[ time 123 ] Run JOB 0 at PRIORITY 0 [ TICKS 3 ALLOT 1 TIME 6 (of 84) ]
[ time 124 ] Run JOB 0 at PRIORITY 0 [ TICKS 2 ALLOT 1 TIME 5 (of 84) ]
[ time 125 ] Run JOB 0 at PRIORITY 0 [ TICKS 1 ALLOT 1 TIME 4 (of 84) ]
[ time 126 ] Run JOB 0 at PRIORITY 0 [ TICKS 0 ALLOT 1 TIME 3 (of 84) ]
[ time 127 ] Run JOB 0 at PRIORITY 0 [ TICKS 5 ALLOT 1 TIME 2 (of 84) ]
[ time 128 ] Run JOB 0 at PRIORITY 0 [ TICKS 4 ALLOT 1 TIME 1 (of 84) ]
[ time 129 ] Run JOB 0 at PRIORITY 0 [ TICKS 3 ALLOT 1 TIME 0 (of 84) ]
[ time 130 ] FINISHED JOB 0

Final statistics:
  Job  0: startTime   0 - response   0 - turnaround 130
  Job  1: startTime   0 - response   4 - turnaround  93

  Avg  1: startTime n/a - response 2.00 - turnaround 111.50


ashutosh@ashutosh:~/Desktop/ClassWork/os/HW-MLFQ$
```

5) Lottery Scheduling (lottery.py)

This program is about lottery scheduling of the process. This program help us to analyse how the lottery scheduling takes place lead to job done. Some commands which we can use in order to analyse how the CPU and IO participates are:

- -h, --help
  show this help message and exit
- -s SEED, --seed=SEED
  the random seed
- -j JOBS, --jobs=JOBS
  number of jobs in the system
- -l JLIST, --jlist=JLIST
  instead of random jobs, provide a comma-separated list of run times and ticket values (e.g., 10:100,20:100 would have two jobs with run-times of 10 and 20, each with 100 tickets)
- -m MAXLEN, --maxlen=MAXLEN

max length of job

- -T MAXTICKET, --maxtick=MAXTICKET
  maximum ticket value, if randomly assigned
- -q QUANTUM, --quantum=QUANTUM
  length of time slice
- -c, --compute
  compute answers for me

Let's suppose we have 4 jobs and we have to analyse this ….

For that the command I use is

python3 lottery.py -j 4 -s 10 -c

```
ashutosh@ashutosh:~/Desktop/ClassWork/os/HW-Lottery$ python3 lottery.py -j 4 -s 10 -c
ARG jlist
ARG jobs 4
ARG maxlen 10
ARG maxticket 100
ARG quantum 1
ARG seed 10

Here is the job list, with the run time of each job:
  Job 0 ( length = 5, tickets = 42 )
  Job 1 ( length = 5, tickets = 20 )
  Job 2 ( length = 8, tickets = 82 )
  Job 3 ( length = 6, tickets = 16 )


** Solutions **

Random 520669 -> Winning ticket 29 (of 160) -> Run 0
  Jobs:
 (* job:0 timeleft:5 tix:42 )
 (  job:1 timeleft:5 tix:20 )
 (  job:2 timeleft:8 tix:82 )
 (  job:3 timeleft:6 tix:16 )

Random 327773 -> Winning ticket 93 (of 160) -> Run 2
  Jobs:
 (  job:0 timeleft:4 tix:42 )
 (  job:1 timeleft:5 tix:20 )
 (* job:2 timeleft:8 tix:82 )
 (  job:3 timeleft:6 tix:16 )

Random 249996 -> Winning ticket 76 (of 160) -> Run 2
  Jobs:
 (  job:0 timeleft:4 tix:42 )
 (  job:1 timeleft:5 tix:20 )
 (* job:2 timeleft:7 tix:82 )
 (  job:3 timeleft:6 tix:16 )

Random 952817 -> Winning ticket 17 (of 160) -> Run 0
  Jobs:
 (* job:0 timeleft:4 tix:42 )
 (  job:1 timeleft:5 tix:20 )
 (  job:2 timeleft:6 tix:82 )
 (  job:3 timeleft:6 tix:16 )
```

1.

2.

```
Random 996557 -> Winning ticket 77 (of 160) -> Run 2
  Jobs:
 (  job:0 timeleft:3 tix:42 )
 (  job:1 timeleft:5 tix:20 )
 (* job:2 timeleft:6 tix:82 )
 (  job:3 timeleft:6 tix:16 )

Random 44556 -> Winning ticket 76 (of 160) -> Run 2
  Jobs:
 (  job:0 timeleft:3 tix:42 )
 (  job:1 timeleft:5 tix:20 )
 (* job:2 timeleft:5 tix:82 )
 (  job:3 timeleft:6 tix:16 )

Random 860161 -> Winning ticket 1 (of 160) -> Run 0
  Jobs:
 (* job:0 timeleft:3 tix:42 )
 (  job:1 timeleft:5 tix:20 )
 (  job:2 timeleft:4 tix:82 )
 (  job:3 timeleft:6 tix:16 )

Random 603191 -> Winning ticket 151 (of 160) -> Run 3
  Jobs:
 (  job:0 timeleft:2 tix:42 )
 (  job:1 timeleft:5 tix:20 )
 (  job:2 timeleft:4 tix:82 )
 (* job:3 timeleft:6 tix:16 )

Random 381606 -> Winning ticket 6 (of 160) -> Run 0
  Jobs:
 (* job:0 timeleft:2 tix:42 )
 (  job:1 timeleft:5 tix:20 )
 (  job:2 timeleft:4 tix:82 )
 (  job:3 timeleft:5 tix:16 )

Random 283618 -> Winning ticket 98 (of 160) -> Run 2
  Jobs:
 (  job:0 timeleft:1 tix:42 )
 (  job:1 timeleft:5 tix:20 )
 (* job:2 timeleft:4 tix:82 )
 (  job:3 timeleft:5 tix:16 )
```

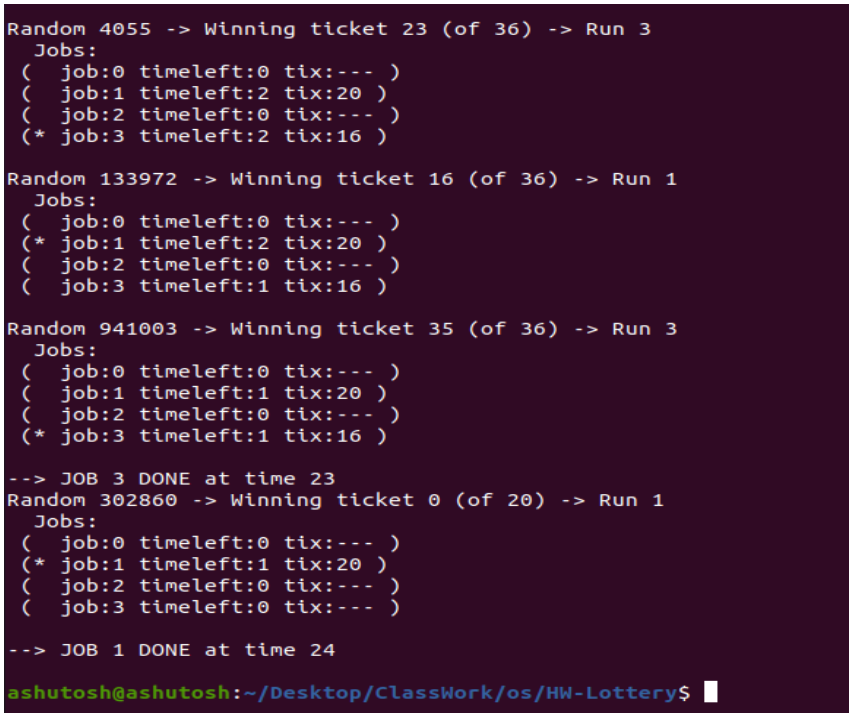
3.

```
Random 674965 -> Winning ticket 85 (of 160) -> Run 2
  Jobs:
 (  job:0 timeleft:1 tix:42 )
 (  job:1 timeleft:5 tix:20 )
 (* job:2 timeleft:3 tix:82 )
 (  job:3 timeleft:5 tix:16 )

Random 456831 -> Winning ticket 31 (of 160) -> Run 0
  Jobs:
 (* job:0 timeleft:1 tix:42 )
 (  job:1 timeleft:5 tix:20 )
 (  job:2 timeleft:2 tix:82 )
 (  job:3 timeleft:5 tix:16 )

--> JOB 0 DONE at time 12
Random 685862 -> Winning ticket 46 (of 118) -> Run 2
  Jobs:
 (  job:0 timeleft:0 tix:--- )
 (  job:1 timeleft:5 tix:20 )
 (* job:2 timeleft:2 tix:82 )
 (  job:3 timeleft:5 tix:16 )

Random 661846 -> Winning ticket 102 (of 118) -> Run 3
  Jobs:
 (  job:0 timeleft:0 tix:--- )
 (  job:1 timeleft:5 tix:20 )
 (  job:2 timeleft:1 tix:82 )
 (* job:3 timeleft:5 tix:16 )

Random 132978 -> Winning ticket 110 (of 118) -> Run 3
  Jobs:
 (  job:0 timeleft:0 tix:--- )
 (  job:1 timeleft:5 tix:20 )
 (  job:2 timeleft:1 tix:82 )
 (* job:3 timeleft:4 tix:16 )
```

4.

```
Random 982414 -> Winning ticket 64 (of 118) -> Run 2
  Jobs:
 (  job:0 timeleft:0 tix:--- )
 (  job:1 timeleft:4 tix:20 )
 (* job:2 timeleft:1 tix:82 )
 (  job:3 timeleft:3 tix:16 )

--> JOB 2 DONE at time 17
Random 969389 -> Winning ticket 17 (of 36) -> Run 1
  Jobs:
 (  job:0 timeleft:0 tix:--- )
 (* job:1 timeleft:4 tix:20 )
 (  job:2 timeleft:0 tix:--- )
 (  job:3 timeleft:3 tix:16 )

Random 613327 -> Winning ticket 31 (of 36) -> Run 3
  Jobs:
 (  job:0 timeleft:0 tix:--- )
 (  job:1 timeleft:3 tix:20 )
 (  job:2 timeleft:0 tix:--- )
 (* job:3 timeleft:3 tix:16 )

Random 44260 -> Winning ticket 16 (of 36) -> Run 1
  Jobs:
 (  job:0 timeleft:0 tix:--- )
 (* job:1 timeleft:3 tix:20 )
 (  job:2 timeleft:0 tix:--- )
 (  job:3 timeleft:2 tix:16 )

Random 4055 -> Winning ticket 23 (of 36) -> Run 3
  Jobs:
 (  job:0 timeleft:0 tix:--- )
 (  job:1 timeleft:2 tix:20 )
 (  job:2 timeleft:0 tix:--- )
 (* job:3 timeleft:2 tix:16 )

Random 133972 -> Winning ticket 16 (of 36) -> Run 1
  Jobs:
 (  job:0 timeleft:0 tix:--- )
 (* job:1 timeleft:2 tix:20 )
 (  job:2 timeleft:0 tix:--- )
 (  job:3 timeleft:1 tix:16 )
```

5.



There are so many functions that I learned Some of them are...

1) move_to_ready
2) move_to_wait
3) move_to_running
4) move_to_done

As the name suggests these function move the job to ready state ,wait state, running state and done state accordingly..