Laboratorio 3.
Estudiante: Rebeca Servellón Orellana.

Instrucciones
Parte 1

En esta parte del laboratorio, deben realizar los pasos descritos en el capítulo 7 del libro Three Easy Pieces, en la sección Homework(Simulation) - página 13. Encontrarán en esta actividad el archivo 'scheduler.py' y 'README'.

Nota: Deben tener el intérprete de Python3 instalado en la PC donde lo van a ejecutar.

Homework (Simulation)

This program, scheduler.py, allows you to see how different schedulers perform under scheduling metrics such as response time, turnaround time, and total wait time. See the README for details.

Questions
1. Compute the response time and turnaround time when running three jobs of length 200 with the SJF and FIFO schedulers.

FIFO
Response time: 200.00
Turnaround: 400.00

```
PS C:\Users\rebec\OneDrive\Escritorio\SO\Lab3> ./scheduler.py -p FIFO -l 200,200,200 -c
ARG policy FIFO
ARG jlist 200,200,200

Here is the job list, with the run time of each job:
  Job 0 ( length = 200.0 )
  Job 1 ( length = 200.0 )
  Job 2 ( length = 200.0 )


** Solutions **

Execution trace:
  [ time   0 ] Run job 0 for 200.00 secs ( DONE at 200.00 )
  [ time 200 ] Run job 1 for 200.00 secs ( DONE at 400.00 )
  [ time 400 ] Run job 2 for 200.00 secs ( DONE at 600.00 )

Final statistics:
  Job   0 -- Response: 0.00  Turnaround 200.00  Wait 0.00
  Job   1 -- Response: 200.00  Turnaround 400.00  Wait 200.00
  Job   2 -- Response: 400.00  Turnaround 600.00  Wait 400.00

  Average -- Response: 200.00  Turnaround 400.00  Wait 200.00
```

SJF
Response time: 200.00
Turnaround: 400.00

```
PS C:\Users\rebec\OneDrive\Escritorio\SO\Lab3> ./scheduler.py -p SJF -l 200,200,200 -c
ARG policy SJF
ARG jlist 200,200,200

Here is the job list, with the run time of each job:
  Job 0 ( length = 200.0 )
  Job 1 ( length = 200.0 )
  Job 2 ( length = 200.0 )


** Solutions **

Execution trace:
  [ time   0 ] Run job 0 for 200.00 secs ( DONE at 200.00 )
  [ time 200 ] Run job 1 for 200.00 secs ( DONE at 400.00 )
  [ time 400 ] Run job 2 for 200.00 secs ( DONE at 600.00 )

Final statistics:
  Job   0 -- Response: 0.00  Turnaround 200.00  Wait 0.00
  Job   1 -- Response: 200.00  Turnaround 400.00  Wait 200.00
  Job   2 -- Response: 400.00  Turnaround 600.00  Wait 400.00

  Average -- Response: 200.00  Turnaround 400.00  Wait 200.00
```

2. Now do the same but with jobs of different lengths: 100, 200, and 300.

FIFO
Response time: 133.33
Turnaround: 333.33

```
PS C:\Users\rebec\OneDrive\Escritorio\SO\Lab3> ./scheduler.py -p FIFO -l 100,200,300 -c
ARG policy FIFO
ARG jlist 100,200,300

Here is the job list, with the run time of each job:
  Job 0 ( length = 100.0 )
  Job 1 ( length = 200.0 )
  Job 2 ( length = 300.0 )


** Solutions **

Execution trace:
  [ time   0 ] Run job 0 for 100.00 secs ( DONE at 100.00 )
  [ time 100 ] Run job 1 for 200.00 secs ( DONE at 300.00 )
  [ time 300 ] Run job 2 for 300.00 secs ( DONE at 600.00 )

Final statistics:
  Job   0 -- Response: 0.00  Turnaround 100.00  Wait 0.00
  Job   1 -- Response: 100.00  Turnaround 300.00  Wait 100.00
  Job   2 -- Response: 300.00  Turnaround 600.00  Wait 300.00

  Average -- Response: 133.33  Turnaround 333.33  Wait 133.33
```

SJF
Response time: 133.33
Turnaround: 333.33

```
PS C:\Users\rebec\OneDrive\Escritorio\SO\Lab3> ./scheduler.py -p SJF -l 100,200,300 -c
ARG policy SJF
ARG jlist 100,200,300

Here is the job list, with the run time of each job:
  Job 0 ( length = 100.0 )
  Job 1 ( length = 200.0 )
  Job 2 ( length = 300.0 )


** Solutions **

Execution trace:
  [ time   0 ] Run job 0 for 100.00 secs ( DONE at 100.00 )
  [ time 100 ] Run job 1 for 200.00 secs ( DONE at 300.00 )
  [ time 300 ] Run job 2 for 300.00 secs ( DONE at 600.00 )

Final statistics:
  Job   0 -- Response: 0.00  Turnaround 100.00  Wait 0.00
  Job   1 -- Response: 100.00  Turnaround 300.00  Wait 100.00
  Job   2 -- Response: 300.00  Turnaround 600.00  Wait 300.00

  Average -- Response: 133.33  Turnaround 333.33  Wait 133.33
```

3. Now do the same, but also with the RR scheduler and a time-slice of 1.
RR
Response time: 1.00
Turnaround: 465.67

```
PS C:\Users\rebec\OneDrive\Escritorio\SO\Lab3> ./scheduler.py -p RR -l 100,200,300 -q 1 -c
ARG policy RR
ARG jlist 100,200,300

Here is the job list, with the run time of each job:
  Job 0 ( length = 100.0 )
  Job 1 ( length = 200.0 )
  Job 2 ( length = 300.0 )


** Solutions **

Execution trace:
  [ time   0 ] Run job   0 for 1.00 secs
  [ time   1 ] Run job   1 for 1.00 secs
  [ time   2 ] Run job   2 for 1.00 secs
  [ time   3 ] Run job   0 for 1.00 secs
  [ time   4 ] Run job   1 for 1.00 secs
  [ time   5 ] Run job   2 for 1.00 secs
  [ time   6 ] Run job   0 for 1.00 secs
  [ time   7 ] Run job   1 for 1.00 secs
  [ time   8 ] Run job   2 for 1.00 secs
  [ time   9 ] Run job   0 for 1.00 secs
  [ time  10 ] Run job   1 for 1.00 secs
```

```
  [ time 590 ] Run job   2 for 1.00 secs
  [ time 591 ] Run job   2 for 1.00 secs
  [ time 592 ] Run job   2 for 1.00 secs
  [ time 593 ] Run job   2 for 1.00 secs
  [ time 594 ] Run job   2 for 1.00 secs
  [ time 595 ] Run job   2 for 1.00 secs
  [ time 596 ] Run job   2 for 1.00 secs
  [ time 597 ] Run job   2 for 1.00 secs
  [ time 598 ] Run job   2 for 1.00 secs
  [ time 599 ] Run job   2 for 1.00 secs ( DONE at 600.00 )

Final statistics:
  Job   0 -- Response: 0.00  Turnaround 298.00  Wait 198.00
  Job   1 -- Response: 1.00  Turnaround 499.00  Wait 299.00
  Job   2 -- Response: 2.00  Turnaround 600.00  Wait 300.00

  Average -- Response: 1.00  Turnaround 465.67  Wait 265.67
```

4. For what types of workloads does SJF deliver the same turnaround times as FIFO?

Para los que el tamaño de los trabajos sean igual y en los que no se deban comparar y el orden de llegada sea el mismo

5. For what types of workloads and quantum lengths does SJF deliver the same response times as RR?

En los que el quantum es igual al tamaño del trabajo

SJF con 3 trabajos de tamaños 100,200,300
Job 0: Response de 0.00, turnaround de 100.00

Job 1: Response de 100.00, turnaround de 300.00
Job 2: Response de 300.00, turnaround de 600.00

```
PS C:\Users\rebec\OneDrive\Escritorio\SO\Lab3> ./scheduler.py -p SJF -l 100,200,300 -c
ARG policy SJF
ARG jlist 100,200,300

Here is the job list, with the run time of each job:
  Job 0 ( length = 100.0 )
  Job 1 ( length = 200.0 )
  Job 2 ( length = 300.0 )


** Solutions **

Execution trace:
  [ time   0 ] Run job 0 for 100.00 secs ( DONE at 100.00 )
  [ time 100 ] Run job 1 for 200.00 secs ( DONE at 300.00 )
  [ time 300 ] Run job 2 for 300.00 secs ( DONE at 600.00 )

Final statistics:
  Job   0 -- Response: 0.00  Turnaround 100.00  Wait 0.00
  Job   1 -- Response: 100.00  Turnaround 300.00  Wait 100.00
  Job   2 -- Response: 300.00  Turnaround 600.00  Wait 300.00
```

RR con quantum de 100
Response time: 0.00
Turnaround: 100.00

```
PS C:\Users\rebec\OneDrive\Escritorio\SO\Lab3> ./scheduler.py -p RR -l 100,200,300 -q 100 -c
ARG policy RR
ARG jlist 100,200,300

Here is the job list, with the run time of each job:
  Job 0 ( length = 100.0 )
  Job 1 ( length = 200.0 )
  Job 2 ( length = 300.0 )


** Solutions **

Execution trace:
  [ time   0 ] Run job   0 for 100.00 secs ( DONE at 100.00 )
  [ time 100 ] Run job   1 for 100.00 secs
  [ time 200 ] Run job   2 for 100.00 secs
  [ time 300 ] Run job   1 for 100.00 secs ( DONE at 400.00 )
  [ time 400 ] Run job   2 for 100.00 secs
  [ time 500 ] Run job   2 for 100.00 secs ( DONE at 600.00 )

Final statistics:
  Job   0 -- Response: 0.00  Turnaround 100.00  Wait 0.00
  Job   1 -- Response: 100.00  Turnaround 400.00  Wait 200.00
  Job   2 -- Response: 200.00  Turnaround 600.00  Wait 300.00

  Average -- Response: 100.00  Turnaround 366.67  Wait 166.67
```

RR con quantum de 200
Response time: 100.00
Turnaround: 300.00

```
PS C:\Users\rebec\OneDrive\Escritorio\SO\Lab3> ./scheduler.py -p RR -l 100,200,300 -q 200 -
ARG policy RR
ARG jlist 100,200,300

Here is the job list, with the run time of each job:
  Job 0 ( length = 100.0 )
  Job 1 ( length = 200.0 )
  Job 2 ( length = 300.0 )


** Solutions **

Execution trace:
  [ time   0 ] Run job   0 for 100.00 secs ( DONE at 100.00 )
  [ time 100 ] Run job   1 for 200.00 secs ( DONE at 300.00 )
  [ time 300 ] Run job   2 for 200.00 secs
  [ time 500 ] Run job   2 for 100.00 secs ( DONE at 600.00 )

Final statistics:
  Job   0 -- Response: 0.00  Turnaround 100.00  Wait 0.00
  Job   1 -- Response: 100.00  Turnaround 300.00  Wait 100.00
  Job   2 -- Response: 300.00  Turnaround 600.00  Wait 300.00
  Average -- Response: 133.33  Turnaround 333.33  Wait 133.33
```

RR con quantum de 300

Response time: 300.00
Turnaround: 600.00

```
PS C:\Users\rebec\OneDrive\Escritorio\SO\Lab3> ./scheduler.py -p RR -l 100,200,300 -q 300 -c
ARG policy RR
ARG jlist 100,200,300

Here is the job list, with the run time of each job:
  Job 0 ( length = 100.0 )
  Job 1 ( length = 200.0 )
  Job 2 ( length = 300.0 )


** Solutions **

Execution trace:
  [ time   0 ] Run job   0 for 100.00 secs ( DONE at 100.00 )
  [ time 100 ] Run job   1 for 200.00 secs ( DONE at 300.00 )
  [ time 300 ] Run job   2 for 300.00 secs ( DONE at 600.00 )

Final statistics:
  Job   0 -- Response: 0.00  Turnaround 100.00  Wait 0.00
  Job   1 -- Response: 100.00  Turnaround 300.00  Wait 100.00
  Job   2 -- Response: 300.00  Turnaround 600.00  Wait 300.00

  Average -- Response: 133.33  Turnaround 333.33  Wait 133.33
```

6. What happens to response time with SJF as job lengths increase? Can you use the simulator to demonstrate the trend?
Incrementa.

```
PS C:\Users\rebec\OneDrive\Escritorio\SO\Lab3> ./scheduler.py -p SJF -l 100,100,100 -c
ARG policy SJF
ARG jlist 100,100,100

Here is the job list, with the run time of each job:
  Job 0 ( length = 100.0 )
  Job 1 ( length = 100.0 )
  Job 2 ( length = 100.0 )


** Solutions **

Execution trace:
  [ time   0 ] Run job 0 for 100.00 secs ( DONE at 100.00 )
  [ time 100 ] Run job 1 for 100.00 secs ( DONE at 200.00 )
  [ time 200 ] Run job 2 for 100.00 secs ( DONE at 300.00 )

Final statistics:
  Job   0 -- Response: 0.00  Turnaround 100.00  Wait 0.00
  Job   1 -- Response: 100.00  Turnaround 200.00  Wait 100.00
  Job   2 -- Response: 200.00  Turnaround 300.00  Wait 200.00

  Average -- Response: 100.00  Turnaround 200.00  Wait 100.00
```

```
PS C:\Users\rebec\OneDrive\Escritorio\SO\Lab3> ./scheduler.py -p SJF -l 200,200,200 -c
ARG policy SJF
ARG jlist 200,200,200

Here is the job list, with the run time of each job:
  Job 0 ( length = 200.0 )
  Job 1 ( length = 200.0 )
  Job 2 ( length = 200.0 )


** Solutions **

Execution trace:
  [ time   0 ] Run job 0 for 200.00 secs ( DONE at 200.00 )
  [ time 200 ] Run job 1 for 200.00 secs ( DONE at 400.00 )
  [ time 400 ] Run job 2 for 200.00 secs ( DONE at 600.00 )

Final statistics:
  Job   0 -- Response: 0.00  Turnaround 200.00  Wait 0.00
  Job   1 -- Response: 200.00  Turnaround 400.00  Wait 200.00
  Job   2 -- Response: 400.00  Turnaround 600.00  Wait 400.00

  Average -- Response: 200.00  Turnaround 400.00  Wait 200.00
```

```
PS C:\Users\rebec\OneDrive\Escritorio\SO\Lab3> ./scheduler.py -p SJF -l 300,300,300 -c
ARG policy SJF
ARG jlist 300,300,300

Here is the job list, with the run time of each job:
  Job 0 ( length = 300.0 )
  Job 1 ( length = 300.0 )
  Job 2 ( length = 300.0 )


** Solutions **

Execution trace:
  [ time   0 ] Run job 0 for 300.00 secs ( DONE at 300.00 )
  [ time 300 ] Run job 1 for 300.00 secs ( DONE at 600.00 )
  [ time 600 ] Run job 2 for 300.00 secs ( DONE at 900.00 )

Final statistics:
  Job   0 -- Response: 0.00  Turnaround 300.00  Wait 0.00
  Job   1 -- Response: 300.00  Turnaround 600.00  Wait 300.00
  Job   2 -- Response: 600.00  Turnaround 900.00  Wait 600.00

  Average -- Response: 300.00  Turnaround 600.00  Wait 300.00
```

7. What happens to response time with RR as quantum lengths increase? Can you write an equation that gives the worst-case response time, given N jobs?

Incrementa.

E = sum(i*quantum for i in range(0, N)) / N