CSC 369 Worksheet 4 Solution

August 18, 2020

1. The following assumptions are made before calculation:

    - Each job runs for the same amount of time
    - All jobs arrive at the same time
    - Once started, each job runs to completion
    - All jobs only use the CPU (i.e they perform no I/O)
    - The run-time of each job is known

    First, I need to calculate the turnaround time when running three job of length 200 with the SJF and FIFO schedulers.

    I will do so in parts.

    - **Part 1:** Calculating turnaround time with FIFO schedulers

    $$\frac{200 + 400 + 600}{3} = 400 \tag{1}$$

    seconds.

    - **Part 2:** Calculating turnaround time with SJF schedulers

    $$\frac{200 + 400 + 600}{3} = 400 \tag{2}$$

    seconds.

    Second, I need to calculate the response time when running three job of length 200 with the SJF and FIFO schedulers.

    I will do so in parts.

- **Part 1:** Calculating response time with FIFO schedulers

$$\frac{0+200+400}{3} = 200 \tag{3}$$

seconds.

- **Part 2:** Calculating response time with SJF schedulers

$$\frac{0+200+400}{3} = 200 \tag{4}$$

seconds.

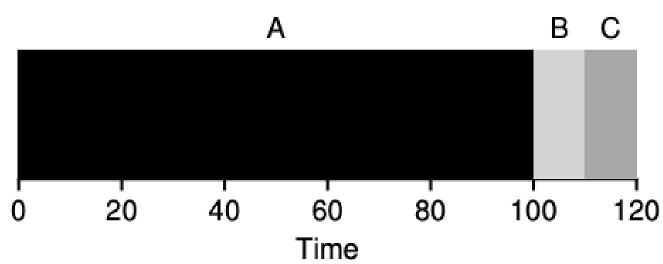
Notes

- **Scheduling:**
  - Is a process at which allows one process to use the CPU while another is on hold, to make full use of CPU
- **Turnaround Time:**
  - Is a performance metric
  - Is amount of time to execute a particular process [1]

$$T_{turnaround} = T_{completion} - T_{arrival} \tag{5}$$

  - $T_{completion} \rightarrow$ Time at which the job completes
  - $T_{arrival} \rightarrow$ Time at which the job arrived in the system
- **FIFO scheduling algorithm:**
  - Is the most basic scheduling algorithm

Example

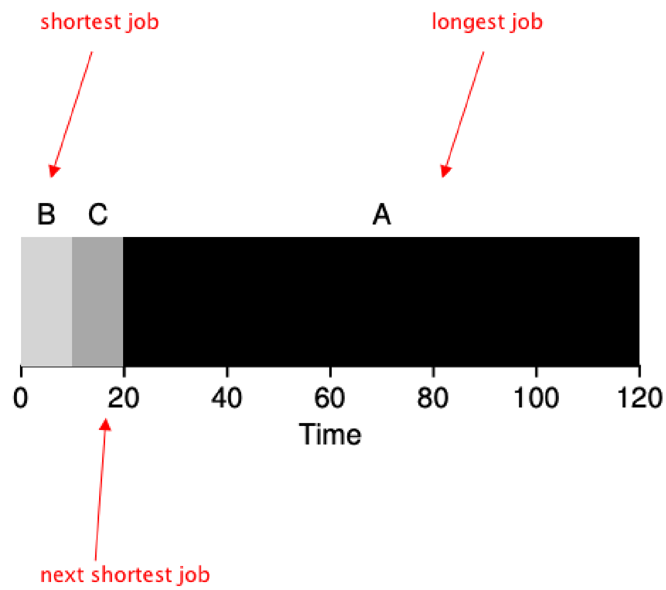


Here, the average turnaround time is:

$$\frac{100 + 110 + 120}{3} = 110 \tag{6}$$

- **SJF scheduling algorithm:**
  - Is a schduling policy where the shortest job is run first, then the next shortest and so on.

Example



Here, the average turnaround time is:

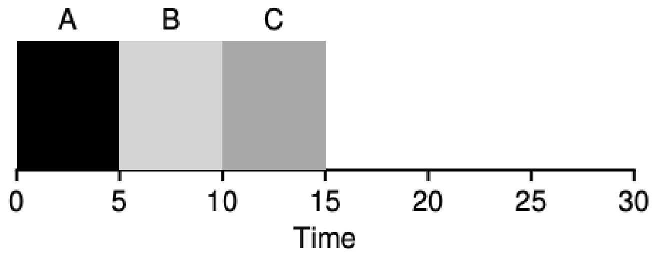
$$\frac{10 + 20 + 120}{3} = 50 \tag{7}$$

- **Response Time:**
  - Is amount of time from when a request was submitted until the first response is produced [1]

$$T_{response} = T_{firstrun} - T_{arrival} \tag{8}$$

  - $T_{firstrun} \rightarrow$ First time a job is scheduled
  - $T_{arrival} \rightarrow$ Time at which the job arrived in the system

Example



Here, the average response time is

$$\frac{0+5+10}{3} = 5 \tag{9}$$

References

1) Old Dominion University, CPU Scheduling link

2. First, I need to calculate turnaround time when running three jobs of different lengths $100, 200, 300$ with FIFO and SJF schedulers.

   I will do so in parts.

   - **Part 1:** Calculating turnaround time with FIFO schedulers

     $$\frac{100+300+600}{3} \approx 333.33 \tag{10}$$

     seconds.
   - **Part 2:** Calculating turnaround time with SJF schedulers

     $$\frac{100+300+600}{3} \approx 333.33 \tag{11}$$

     seconds.

   Second, I need to calculate response time when running three jobs of different lengths $100, 200, 300$ with FIFO and SJF schedulers.

   I will do so in parts.

- **Part 1:** Calculating turnaround time with FIFO schedulers

$$\frac{0+100+300}{3} \approx 133.33 \tag{12}$$

- **Part 2:** Calculating turnaround time with SJF schedulers

$$\frac{0+100+300}{3} \approx 133.33 \tag{13}$$

3. Let the time slice of round robin be 1.

First, I need to calculate turnaround time when running three jobs of different lengths $100, 200, 300$ with Round Robin schedulers.

The answer is

$$\frac{298+499+600}{3} \approx 465.67 \tag{1}$$

Given that the time of completetion are

$$\begin{align}
&\# \text{ of jobs} \times (\text{Len. job } 1 - 1) + 1 \tag{2}\\
&= 3 \times (100 - 1) + 1 \tag{3}\\
&= 298 \tag{4}
\end{align}$$

for job 1,

$$\begin{align}
&\# \text{ of jobs} \times (\text{Len. job } 1 - 1) + 2 + (2 \times (\text{Len. job } 2 - \text{Len. job } 1)) \tag{5}\\
&= 3 \cdot (100 - 1) + 2 + (2 \times (200 - 100)) \tag{6}\\
&= 299 + 200 \tag{7}\\
&= 499 \tag{8}
\end{align}$$

for job 2, and 600 for job 3.

Second, I need to calculate response time when running three jobs of different lengths $100, 200, 300$ with Round Robin schedulers.

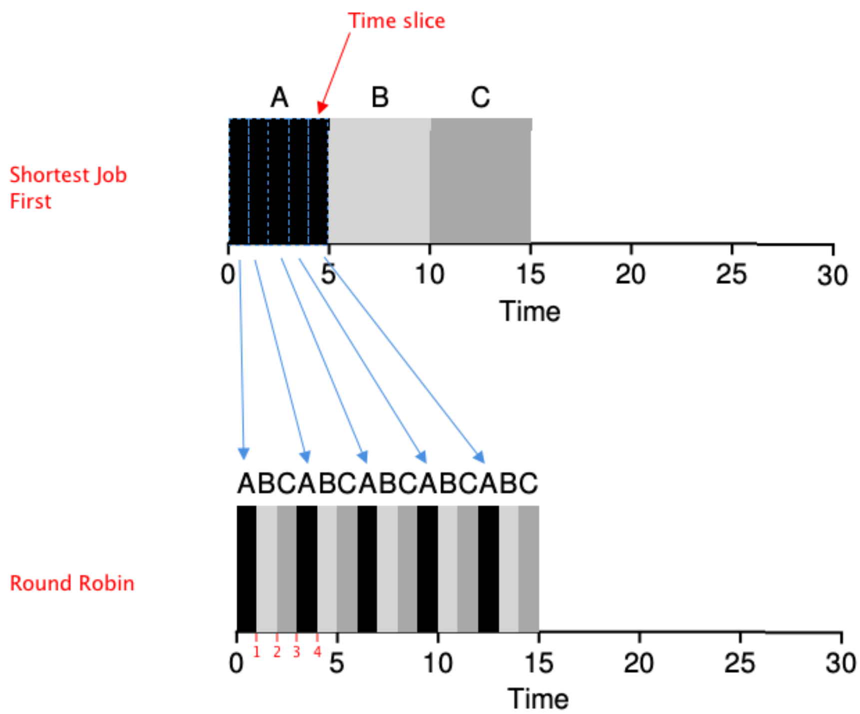
The answer is

$$\frac{0+1+2}{3} = 1 \tag{9}$$

seconds.

Notes

- **Round Robin:**
  - Solves the problem of having to wait for x number of seconds (e.g 10 seconds) before the next job.
  - Is sometimes-called **time-slicing**
  - Does so by running a job for **time slice** (sometimes called **scheduling quantum**)
  - Returns excellent running time, but not-so-good turnaround time



Notes

In above example, the Response time of Round Robin is

$$\frac{0+1+2}{3} = 1 \tag{10}$$

seconds.

4. I need to ansewr the types of workloads SJF deliver the same turnaround time as FIFO.

SJF delivers the same turnaround time as FIFO when workloads are either

1. In increasing order by job length (e.g. job 1 has length 100, job 2 has length 200, and job 3 has length 300)
2. Have the same job length (e.g All jobs have length of 200)

5. I need to answer answer what types of workloads and quantum lengths does SJF deliver the same response time as RR?

   SJF delivers the same response time as RR when the length of each job is the same as the quantum length of each job.

6. As job lengths increase, the average response time also increases.

   The following simulations are done to demonstrate the relationship between job lengths and average response time

   - ./scheduler.py -p SJF -j 3 -l 100,100,100 -c
   - ./scheduler.py -p SJF -j 3 -l 200,200,200 -c
   - ./scheduler.py -p SJF -j 3 -l 300,300,300 -c
   - ./scheduler.py -p SJF -j 3 -l 400,400,400 -c

7. First, I need to answer what happens to response time with RR as quantum length increases.

   As quantum length increases, the response time with RR also increases.

   Second, I need to write an equation that gives the worst-case response time, given N jobs.

   Let m_i where $i = 1, ..., n$ represent the length of i^{th} job.

   Since the response time is highest when all of their quantum length are equal to their corresponding job length, the worst-case response time is

$$\frac{\sum_{i=1}^{N-1} \sum_{j=1}^{i} m_j}{N} \tag{11}$$

   seconds.

8.