**Operating Systems**

Shyam Rajendren

Operating Systems Environments & Administration

1. Explain the difference between the **CPU** and **I/O** cycles that make up a program.

Answer: The Process Scheduler makes use of alternating between CPU and I/O cycles when completing a task. Usually, a relatively long CPU cycle is both preceded and succeeded by a relatively short I/O cycle. During the first I/O cycle, information is retrieved from the user. During the CPU cycle, the necessary computations are done on the inputs given by the users and a final solution(s) are derived. During the second I/O cycle, the results of the task are displayed to the user.

1. Can a process run during an **I/O** cycle?

Answer: No, it cannot, the CPU must wait for the I/O cycle to complete before running another process.

1. Explain how the **Process Control Block** (PCB) is used to manage a process within the system.

Answer: A Process Control Block (PCB) is a data structure that contains information about the current status and characteristics of a process. The PCB is created when the Job Scheduler accepts a job and is updated as the job progresses from the beginning to the end of its execution. The PCB uses these various contents to manage a process:

**Pointer** – A stack pointer which is saved when a process is switched from one state to another so as to retain the current position of the process.

**Process state** – Stores the state of the process.

**Process number** – A unique ID which stores the process identifier.

**Program counter** – Stores the counter which contains the address of the next instruction that is to be executed.

**Register** – Includes the accumulator, base registers, and general-purpose registers.

**Memory limits** – Contains the information about the memory management system used by operating system.

**Open files list** – The list of files that were opened for this process.

1. What are the three (3) states of the **Process Scheduler**?

Answer: The three states of the Process Scheduler are the Ready State, the Running State, and the Waiting State.

1. List all of the paths a process can take between the three (3) states of the **Process Scheduler**.

Answer: The job’s transition from HOLD to READY is initiated by the Job Scheduler, according to a policy that’s predefined by the operating system’s designers. The availability of main memory and any requested devices is checked. The transition from READY to RUNNING is handled by the Process Scheduler according to a predefined algorithm. The transition from RUNNING back to READY is handled according to a predefined time limit or some other criteria such as a priority interrupt. The transition from RUNNING to WAITING is initiated in response to an instruction in the job, such as a command to READ, WRITE, or another I/O request. The transition from WAITING to READY is initiated by a signal from the I/O device manager that occurs when the I/O request has been satisfied. In the case of a page fetch, the page fault handler will signal that the page is now in memory and the process can be placed back in the READY queue. The transition from RUNNING to FINISHED is initiated by the Process Scheduler, or the Job Scheduler, when the job is successfully completed or when the operating system indicates that an error has occurred and that the job must be terminated.

1. What are some advantages and disadvantages of using a single I/O interrupt queue?

Answer: Some advantages are that it is easy to implement and that the average wait time is minimized. Some disadvantages are that there is an unpredictable turn around time and some jobs can be indefinitely postponed.

1. What are some advantages and disadvantages of using multiple I/O interrupt queues?

Answer: Some advantages are that it has a flexible scheme and allows ageing or other queue movement to counteract indefinite postponement. Some disadvantages are that is relatively difficult to implement and that an overhead is incurred due to monitoring multiple queues.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Job** | A | B | C | D | E | F |
| **Arrival Time** | 0 | 2 | 4 | 6 | 8 | 10 |
| **CPU Cycles** | 8 | 1 | 5 | 2 | 4 | 3 |

1. Refer to the table above to answer the following questions.  
   1. What is the average turnaround time for the First-Come, First-Served (FCFS) process scheduling algorithm?

Answer:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Job** | **Arrival Time** | **CPU Cycles** | **Completion Time** | **Turnaround Time (Completion Time – Arrival Time)** |
| A | 0 | 8 | 8 | 8 |
| B | 2 | 1 | 9 | 7 |
| C | 4 | 5 | 14 | 10 |
| D | 6 | 2 | 16 | 10 |
| E | 8 | 4 | 20 | 12 |
| F | 10 | 3 | 23 | 13 |

Total Turnaround Time = 8+7+10+10+12+13 = 60

Average Turnaround time = Total Turnaround Time / Total number of jobs = 60/6 = **10**

* 1. What is the average turnaround time for the Shortest Job Next (SJN) process scheduling algorithm?  
     Answer:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Job** | **Arrival Time** | **CPU Cycles** | **Completion Time** | **Turnaround Time (Completion Time – Arrival Time)** |
| A | 0 | 8 | 8 | 8 |
| B | 2 | 1 | 9 | 7 |
| C | 4 | 5 | 23 | 19 |
| D | 6 | 2 | 11 | 5 |
| E | 8 | 4 | 18 | 10 |
| F | 10 | 3 | 14 | 4 |

Total Turnaround Time = 8+7+19+5+10+4 = 53

Average Turnaround time = Total Turnaround Time / Total number of jobs = 53/6 = **8.83**

* 1. What is the average turnaround time for the Shortest Remaining Time (SRT) process scheduling algorithm?

Answer:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Job** | **Arrival Time** | **CPU Cycles** | **Completion Time** | **Turnaround Time (Completion Time – Arrival Time)** |
| A | 0 | 8 | 11 | 11 |
| B | 2 | 1 | 3 | 1 |
| C | 4 | 5 | 23 | 19 |
| D | 6 | 2 | 8 | 2 |
| E | 8 | 4 | 18 | 10 |
| F | 10 | 3 | 14 | 4 |

Total Turnaround Time = 11+1+19+2+10+4 = 47

Average Turnaround time = Total Turnaround Time / Total number of jobs = 47/6 = **7.83**

* 1. What is the average turnaround time for the Round Robin process scheduling algorithm with a time quantum of 2?

Answer:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Job** | **Arrival Time** | **CPU Cycles** | **Completion Time** | **Turnaround Time (Completion Time – Arrival Time)** |
| A | 0 | 8 | 17 | 17 |
| B | 2 | 1 | 3 | 1 |
| C | 4 | 5 | 20 | 16 |
| D | 6 | 2 | 11 | 5 |
| E | 8 | 4 | 22 | 14 |
| F | 10 | 3 | 23 | 13 |

Total Turnaround Time = 17+1+16+5+14+13= 66

Average Turnaround time = Total Turnaround Time / Total number of jobs = 66/6 = **11**

* 1. Does changing the time quantum to 3 increase or decrease the average turnaround time for the Round Robin process scheduling algorithm?

Answer:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Job** | **Arrival Time** | **CPU Cycles** | **Completion Time** | **Turnaround Time (Completion Time – Arrival Time)** |
| A | 0 | 8 | 14 | 14 |
| B | 2 | 1 | 4 | 2 |
| C | 4 | 5 | 22 | 18 |
| D | 6 | 2 | 12 | 6 |
| E | 8 | 4 | 23 | 15 |
| F | 10 | 3 | 20 | 10 |

Total Turnaround Time = 14+2+18+6+15+10 = 65

Average Turnaround time = Total Turnaround Time / Total number of jobs = 65/6 = **10.83**  
**Increasing the time quantum to 3 marginally decreased the average turnaround time.**

1. In general, which process scheduling algorithm produces the smallest average turnaround time?

Answer: The Shortest Remaining Time (SRT) process scheduling algorithm produced the smallest average turnaround time.

1. Explain the advantages and disadvantages of a non-pre-emptive process scheduling algorithm.

Answer: Some advantages are that it is easy to implement, the average waiting time is minimized, and it ensures fast completion of important jobs. Some disadvantages are that is has an unpredictable turnaround time and leads to indefinite postponement of some jobs.

1. Explain the advantages and disadvantages of a pre-emptive process scheduling algorithm.

Answer: Some advantages are that it ensures fast completion of short jobs, provides fair CPU allocation, and that it attempts to complete jobs in a timely manner. Some disadvantages are that a large overhead can be incurred by context switching in case of Shortest Remaining Time (SRT), a good time quantum must be selected in case of Round Robin, and a large overhead is required to monitor dynamic deadlines in case of Earliest Deadline First (EDF).