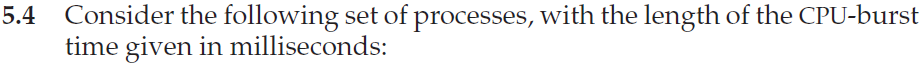
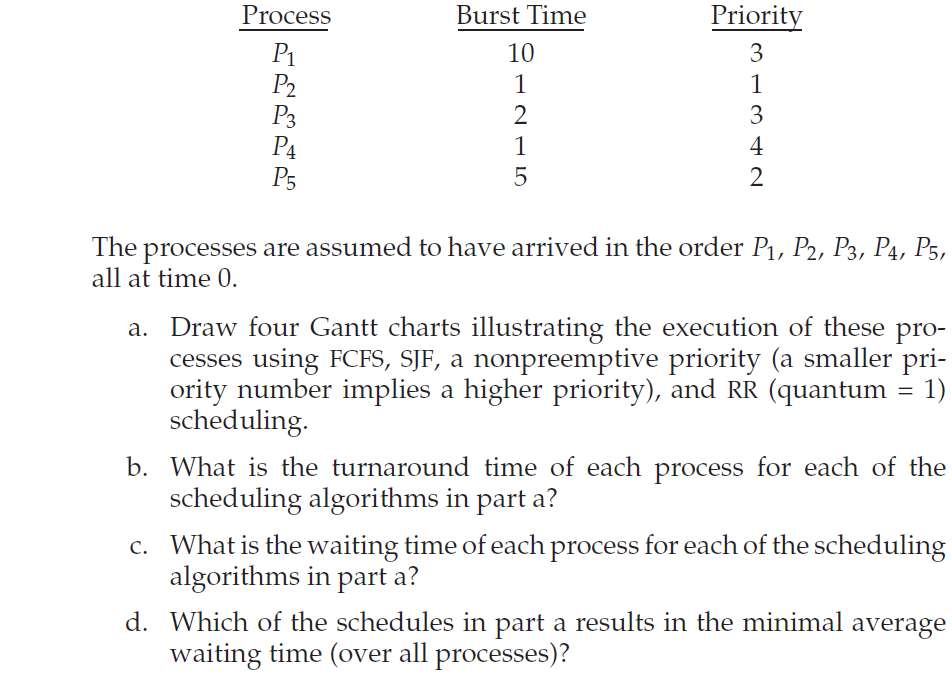


(스케줄러가 입/출력 중심 프로그램과 CPU 중심 프로그램을 구분하는 것은 왜 중요한가?)

입/출력 중심의 프로그램은 전형적으로 짧은 CPU 버스트를 많이 가지고, CPU 지향 프로그램은 다수의 긴 CPU 버스트를 가진다. 그래서 적절한 CPU 스케줄링 알고리즘을 선택하기 위해 이 둘을 구분하는 것은 매우 중요하다.





(다음 프로세스들의 집합을 생각해 보자. CPU 버스트 시간 단위는 밀리초이다.

프로세스, 버스트 시간, 우선순위

프로세스들은 시간 0에 P1, P2, P3, P4, P5 순서로 도착된다고 가정한다.

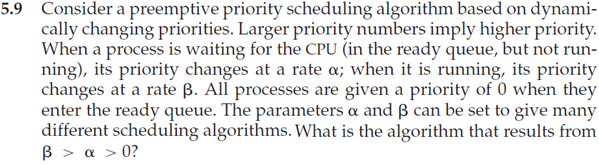
1. 선입 선처리, SJF, 비선점 우선순위 (작은 우선순위 값이 높은 우선순위를 의미) 그리고 라운드 로빈(할당량 = 1) 스케줄링을 이용해 프로세스들의 실행을 보이는 Gantt 차트를 그려라.
2. a에서 각 스케줄링 알고리즘의 각 프로세스에 대한 총처리 시간은 얼마인가?
3. a에서 각 스케줄링 알고리즘의 각 프로세스에 대한 대기 시간은 얼마인가?
4. a에서 어느 스케줄이 최소의 평균 대기 시간(모든 프로세스에 대해)을 갖는가?)

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | | 3 | | 4 | | 5 | 6 | 7 | 8 | 9 | 10 | | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 |
| 선입 선처리 | P1 | | | | | | | | | | | | | P2 | | P3 | | P4 | P5 | | | | |
| SJF | P2 | | P4 | | P3 | | | P5 | | | | | P1 | | | | | | | | | | |
| 비선점 우선순위 | P2 | | P5 | | | | | | | P1 | | | | | | | | | | | P3 | | P4 |
| 라운드 로빈 | P1 | | P2 | | P3 | | P4 | P5 | P1 | P3 | P5 | P1 | P5 | | P1 | P5 | P1 | P5 | P1 | | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 선입 선처리 | SJF | 비선점 우선순위 | 라운드 로빈 |
| P1 | 10 | 19 | 16 | 19 |
| P2 | 11 | 1 | 1 | 2 |
| P3 | 13 | 4 | 18 | 7 |
| P4 | 14 | 2 | 19 | 4 |
| P5 | 19 | 9 | 6 | 14 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 선입 선처리 | SJF | 비선점 우선순위 | 라운드 로빈 |
| P1 | 0 | 9 | 6 | 9 |
| P2 | 10 | 0 | 0 | 1 |
| P3 | 11 | 2 | 16 | 5 |
| P4 | 13 | 1 | 18 | 3 |
| P5 | 14 | 4 | 1 | 9 |

1. SJF가 최소의 평균 대기 시간을 갖는다.



(동적으로 변화하는 우선순위에 기반을 둔 다음의 선점형 우선순위 스케줄링 알고리즘에 대해 생각해 보자. 우선순위 값이 클수록 높은 우선순위를 의미한다. 프로세스가 CPU를 대기할 때 (준비 완료 큐에 있으나 실행되지는 않는) 그 프로세스들의 우선순위는 α의 비율로 변화한다. 프로세스가 실행되면 우선순위는 β의 비율로 변한다. 준비 완료 큐에 들어올 때 모든 프로세스들의 우선순위는 0으로 주어진다. 매개변수 α와 β의 값에 따라 여러 가지 다른 스케줄링 알고리즘이 만들어진다. Β > α > 0이면 무슨 알고리즘인가?)

First Come First Served (FCFS or FIFO) 알고리즘이다. 왜냐하면 대기 중인 프로세스가 실행중인 프로세스보다 우선순위가 항상 낮기 때문에, 먼저 들어온 것이 먼저 나가는 FIFO 알고리즘이다.