

4W__Questions

4W Questions

Abstract

Make note atomic.

Point of capturing note is that I want get into the habit of capturing and if I feel the friction of, this is not atomic enough, not properly then less likely to capture notes as a habit. So you've given yourself the permission to just capture notes however you want to do it.

This note should have Questions, Ideas, Supplementary tools. Question tags question, Ideas tags paradox or counter-intuitive and Supplementary tools tags quote, anecdote and scientific study.

Status could be build(develop ideas), check(confirm idea, for accuracy, source, etc.), to-read and to-watch(find time to consume and take notes) as Input area. As output could anything if you want to contribute or publish status could be magazine name or instagram etc.

Title should be easy to find via keyword(general word). Putting the keywords that you think you will search for later down the line.

Idea:

1. 프로세스 메모리 영역 중 전역변수, 지역변수, 매개변수가 저장되는 위치를 설명하시오.
2. RR에서 time slice(time quantum)가 길어진다면 어떤 스케줄링과 비슷해지는지 그 이유와 함께 설명하시오.
3. RR을 사용하기 좋은 경우에 대해 설명하시오.
4. RR을 사용할 때, CPU bound 프로세스가 많은 경우를 반환 시각과 관련하여 설명하고, 이때 사용할 time slice의 길이(길게/짧게)를 정한 후 그 이유를 설명하시오..^[1]
5. 스케줄링에서 응답시간은 "준비 큐에서 처음 CPU를 할당 받는 데까지 걸린 시간"을 의미한다. 다음 5 개 프로세스의 큐 도착 시간과 Burst time이 주어질 때 SJF와 SRT 스케줄링에서 평균 대기 시간과 평균 응답 시간을 구하시오..^[2]
(단, SJF와 SRT는 각각 CPU 사용이 종료될 때, 새로운 process가 도착할 때 스케줄링을 실시한다)

Process	Arrival Time	Burst Time
P_1	0.0	7
P_2	2.0	4
P_3	4.0	1
P_4	5.0	4

6. Turnaround time은 프로세스가 큐에서 반환되는 시각을 의미한다. 프로세스가 위 표 처럼 주어졌을 때, RR에서 평균 반환 시각과 응답 시각을 SJF의 경우와 비교하여 설명하시오..^[3]

Bearing:

Info

There is four area, West, East, North, South. These based on Question, Evidence and Conclusion.

West means What's similar to X. In this field supplementary tools and related concepts are involved.

East means What's opposite of X.

North means Where X comes from.

South means Where X leads to.

West: similar

East: opposite

North: theme / question

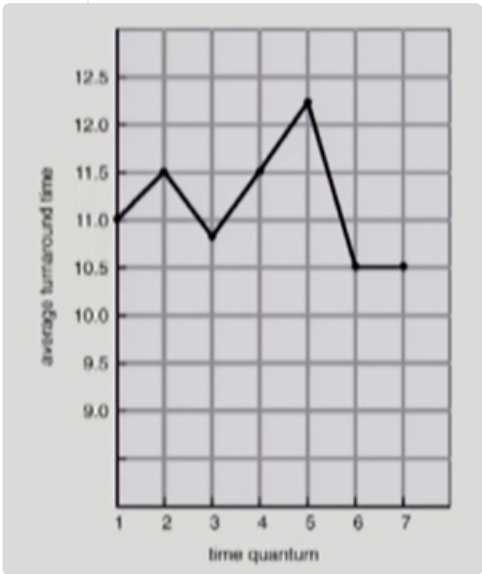
South: what does this lead to

Sources

1. Operating_System_5.CPU_Scheduling_1_20230731 > 4.5.4.2. Turnaround Time Varies With Time Quantum

4.5.4.2. Turnaround Time Varies With Time Quantum

Process	Burst time
P ₁	6
P ₂	3
P ₃	1
P ₄	7



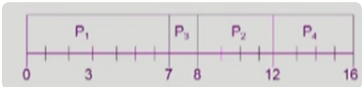
- 위 표는 time quantum을 늘려나갔을 때 average turnaround time을 확인한다.
- 표에서는 확인하기 어렵지만, RR에서 time quantum을 늘릴 수록 average turnaround time은 더 감소하는 경향이 있다.

2. Operating_System_5.CPU_Scheduling_1_20230731 > 4.5.2.1. Examples

4.5.2.1. Examples

Process	Arrival Time	Burst Time
P ₁	0.0	7
P ₂	2.0	4
P ₃	4.0	1
P ₄	5.0	4

- SJF (Non-preemptive)
 - CPU 사용이 종료된 시점에 스케줄링을 실시한다.



- Average waiting time = $(0 + 6 + 3 + 7)/4 = 4$
- 1. 제일 먼저 도착한 P₁은 처리가 완료될 때까지 CPU를 사용 후 반납한다.
- 2. P₁이 완료된 시점에 확인했을 때 P₂, P₃, P₄가 도착했으며, 이 중 가장 짧은 CPU burst time을 가진 P₃에 CPU를 할당한다.
- 3. 처리가 완료되면 보다 먼저 도착한 P₂에 CPU를 할당한다.
- 4. P₄에 CPU를 할당한다.

- SJF (Preemptive)
 - 새로운 process가 도착하면 스케줄링을 실시한다.



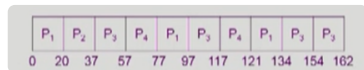
- Average waiting time waiting-time-in-preemptive = $(9 + 1 + 0 + 2)/4 = 3$
1. 가장 먼저 도착한 P_1 에 CPU를 할당한다.
 2. P_2 가 도착한 2초에 P_1 과 P_2 의 CPU burst time을 비교, 더 짧은 P_2 에 CPU를 할당한다.
 3. P_3 가 도착한 4초에 P_1 , P_2 , P_3 의 CPU burst time을 비교, 가장 짧은 P_3 에 CPU를 할당한다.
 4. P_3 종료 후 queue에 남은 P_1 , P_2 , P_4 의 CPU burst time을 비교해 가장 짧은 P_2 를 할당한다.
 5. P_1 , P_4 중 burst time이 짧은 P_4 를 할당한다.
 6. 남은 P_1 을 실행한다.

3. Operating_System_5.CPU_Scheduling_1_20230731 > 4.5.4.1. Example

4.5.4.1. Example

Process	Burst Time
P_1	53
P_2	17
P_3	68
P_4	24

- RR with Time Quantum = 20



- 일반적으로 SJF 보다 average turnaround time이 길지만 response time은 더 짧다. rr-turnaround-time