**운영체제(Operating System) - 과제 #4**

**202195039**

**임서윤**

1. 디스패처(Dispatcher)의 주요 목적은 무엇인가? 그리고 호출되는 시점에 대해 설명하시오

* 디스패처의 주요 목적은 운영 체제에서 CPU 시간을 효율적으로 관리하고 프로세스 또는 스레드에게 CPU를 할당하는 것입니다..

 **프로세스 생성 시:** 새로운 프로세스가 생성되면 디스패처는 해당 프로세스를 준비 큐에 추가하고 스케줄링합니다.

 **프로세스 종료 시:** 프로세스가 종료되면 디스패처는 CPU를 다른 프로세스에 할당합니다.

 **인터럽트 발생 시:** I/O 요청, 타이머 인터럽트 등이 발생하면 현재 실행 중인 프로세스를 중단하고 디스패처를 호출하여 다음에 실행할 프로세스를 결정합니다.

 **시스템 콜 발생 시:** 프로세스가 시스템 콜을 통해 운영 체제의 서비스를 요청하면 디스패처가 호출되어 해당 요청을 처리하고 CPU를 적절히 할당합니다.

1. Batch 처리 방식의 운영체제와 대화형 운영체제에서 중요시하는 성능 요구사항 을 비교하여 설명하시오.

* Bach 처리 방식의 운영체제 중요시하는 성능 요구사항은 처리량, 전환 시간, CPU 이용률이다.

처리량은 단위 시간당 처리하는 작업의 양을 최대화하는 것이 중요합니다. 즉, 주어진 시간 안에 얼마나 많은 작업을 처리할 수 있는지가 핵심적인 성능 지표입니다.

전환 시간은 작업을 제출한 시점부터 작업이 완료될 때까지 걸리는 시간을 최소화하는 것이 중요합니다.

CPU 이용률은 CPU를 최대한 활용하여 유휴 시간을 줄이는 것이 중요합니다.

* 대화형 운영체제가 중요시하는 성능 요구사항은 응답 시간, 예측 가능성, 사용자 친화성이다.

응답 시간은 사용자의 명령에 대한 시스템의 반응 속도를 최소화하는 것이 중요합니다. 즉, 사용자가 명령을 입력하고 시스템이 반응하는 데까지 걸리는 시간이 짧아야 합니다.

예측 가능성은 시스템의 응답 시간이 일정하고 예측 가능해야 합니다.

사용자 친화성은 사용자 인터페이스가 직관적이고 사용하기 쉬워야 합니다.

Bach 처리 방식은 대량의 데이터를 처리하는 데 적합하며, 대화형 운영체제는 사용자와의 상호 작용이 많은 응용 프로그램에 적합합니다.

(3) 순수 우선순위 기반 스케줄링 방식이 사용될 때 발생가능한 문제점에 대해 설 명하시오.

1. 순수 우선순위 기반 스케줄링 방식이 사용될 때 발생가능한 문제점에 대해 설 명하시오.

주요 문제점은 기아 현상(starvation)입니다. 우선순위가 낮은 프로세스가 계속해서 실행 기회를 얻지 못하고 무기한 대기할 수 있습니다.

1. HRRN(Highest Response Ratio Next) 스케줄링에 대해 간략하게 설명하시오.

HRRN(Highest Response Ratio Next) 비선점형 알고리즘 :

* 응답률이 가장 높은 프로세스를 다음 실행 대상으로 선택합니다.
* 응답률 = (대기 시간 + 예상 실행 시간) / 예상 실행 시간
* 이 방식은 대기 시간이 길고 실행 시간이 짧은 프로세스에 우선권을 부여합니다.

1. 다단계 피드백 스케줄링 정책은 처리기-중심(CPU-bound) 프로세스와 입출력- 중심(I/O-bound) 프로세스 중 어느 유형의 프로세스를 더 우대하는가? 그 이유 를 간략하게 설명하여라.

다단계 피드백 스케줄링은 입출력 – 중심(I/O- bound) 프로세스를 더 우대합니다.

* I/O-bound 프로세스는 CPU 사용 시간이 짧고 자주 I/O 작업을 수행합니다.
* 이러한 프로세스는 높은 우선순위 큐에 머물 가능성이 높습니다.
* CPU-bound 프로세스는 시간 할당량을 모두 사용하여 낮은 우선순위 큐로 이동하게 됩니다.

1. 단일처리기 환경에서 처리해야 하는 다음 작업들에 대해 질문에 답하시오.

|  |  |  |
| --- | --- | --- |
| 프로세스 이름 | CPU burst 시간(ms) | 도착 시간(ms) |
| P1 | 4 | 0 |
| P2 | 12 | 2 |
| P3 | 8 | 6 |
| P4 | 6 | 16 |
| P5 | 10 | 20 |
|  |  |  |

① 비선점 모드로 동작하는 Shortest Process Next(SPN)과 선점 모드로 동작하는 Shortest Remaining Time(SRT) 알고리즘을 사용하여 스케줄링하시오. 각 알고 리즘의 스케줄링 결과를 GANTT 챠트로 나타내시오.

텍스트, 친필, 평행, 문서이(가) 표시된 사진

자동 생성된 설명

② 각 알고리즘에 대해 평균 반환 시간을 계산하여 제시하시오.

SPN(Shortest Process Next)

* P1 : 4 – 0 = 4
* P2 : 16 – 2 = 14
* P3 : 30 – 6 = 24
* P4 : 22 – 16 = 6
* P5 : 40 – 20 = 20

평균 반환 시간 = (4 + 14 + 24 + 6 + 20) / 5 = 13.6MS

끝나는 점 – cpu 도착지점

SRT(Shortest Remaining Time)

* P1 : 4 – 0 = 4
* P2 : 30 – 4 = 26
* P3 : 13 – 6 = 7
* P4 : 22 – 16 = 6
* P5 : 40 – 30 = 10

평균 반환 시간 = (4 + 25 + 7 + 6 + 10) / 10.4ms

끝나는 점 – cpu 시작지점

1. A부터 E까지 5개의 일괄 작업이 어떤 컴퓨터 센터에 동시에 도작하였다. 이들 의 실행 예상 시간은 각각 15, 9, 3, 6, 12분이다. 또한 외부에서 지정해준 이 들 작업의 우선순위는 각각 6, 3, 7, 9, 4이다(값이 작을수록 우선순위는 높 다). 다음의 4가지 스케줄링 정책 각각에 대해 각 프로세스의 반환 시간과 모든 프로세스들에 대한 평균 반환 시간을 계산하여라. 프로세스간의 문맥 교환에 드는 시간을 무시한다. 각각 시간이 산출되는 과정을 제시하여라. 아래의 정책 중에 마지막 마지막 3개의 정책에서는 한 번에 한 작업이 끝날 때까지 계속 실행된다 고 가정하다. 또, 모든 작업들은 처리기-중심 프로세스라고 가정한다.
2. **시간 할당량이 1분인 라운드 로빈**
3. **우선순위 스케줄링**
4. **FCFS(실행 순서는 영문자 순으로 가정한다.)**
5. **Shortest Job First(SJF)**

**A**

|  |  |  |  |
| --- | --- | --- | --- |
| 프로세스 | 실행시간 | 완료시간 | 반환시간 |
| A | 15 | 75 | 75 |
| B | 9 | 45 | 45 |
| C | 3 | 15 | 15 |
| D | 6 | 30 | 30 |
| E | 12 | 60 | 60 |

(75 + 45 + 15 + 30 + 60) / 5 = 45ms

**B**

|  |  |  |  |
| --- | --- | --- | --- |
| 프로세스 | 실행시간 | 완료시간 | 반환시간 |
| B | 9 | 9 | 9 |
| E | 12 | 21 | 21 |
| A | 15 | 36 | 36 |
| C | 3 | 39 | 39 |
| D | 6 | 45 | 45 |

우선순위 : B > E > A > C > D

(9 + 21 + 36 + 39 + 45) / 5 = 30ms

**C**

|  |  |  |  |
| --- | --- | --- | --- |
| 프로세스 | 실행시간 | 완료시간 | 반환시간 |
| A | 15 | 15 | 15 |
| B | 9 | 24 | 24 |
| C | 3 | 27 | 27 |
| D | 6 | 33 | 33 |
| E | 12 | 45 | 45 |

실행 순서는 영문자 순서 : A > B > C > D > E

(15 + 24 + 27 + 33 + 45) / 5 = 28.8ms

**D**

|  |  |  |  |
| --- | --- | --- | --- |
| 프로세스 | 실행시간 | 완료시간 | 반환시간 |
| C | 3 | 3 | 3 |
| D | 6 | 9 | 9 |
| B | 9 | 18 | 18 |
| E | 12 | 30 | 30 |
| A | 15 | 45 | 45 |

실행시간 짧은 순서 : C > D > B > E > A

(3 + 9 + 18 + 30 + 45) / 5 = 21ms

1. 멀티프로세서 쓰레드 스케줄링에 사용되는 네 가지 기법에 대해 간략하게 설명 하여라.

* 부하공유(Load Sharing) : 프로세스들을 특정 처리기에 고정 할당하지 않고 여러 처리기에 분산시키는 방식
* 갱 스케줄링(Gang Scheduling) : 서로 관련된 스레드들을 그룹으로 묶어 여러 처리기에서 동시에 실행하는 방식
* 전용 프로세서 할당(Dedicated Processor Assignment) : 각 스레드를 특정 처리기에 고정 할당하고 실행이 완료될 때까지 그 처리기에서 실행하는 방식
* 동적 스케줄링(Dynamic Scheduling) : 실행 중에 작업 부하에 따라 유연하게 스레드를 처리기에 할당하는 방식

1. 경성 실시간 태스크와 연성 실시간 태스크의 차이점은 무엇인가?

* 경성 실시간 태스크 : 반드시 기한 내에 완료되어야 하며, 기한을 넘기면 심각한 결과를 초래할 수 있습니다.
* 연성 실시간 태스크 : 기한을 넘겨도 시스템에 치명적인 영향을 주지 않으며, 일정 수준의 지연이 허용됩니다.

1. 실시간 운영체제 환경에 발생하는 우선순위 역전에 대해 설명하고, 이에 대해 실시간 운영체제가 지원하는 해결책에 대해 설명하여라.

우선순위 역전 : 높은 우선순위 프로세스가 낮은 우선순위 프로세스에 의해 간접적으로 블록되는 현상입니다.

* 우선순위 상속(Priority Inheritance)
* 우선순위 천장 프로토콜(Priority Ceiling Protocol)
* 우선순위 역전 방지 프로토콜(Priority Inversion Pervention Protocol)

1. 다음의 실시간 CPU 스케줄링 알고리즘을 비교하여 설명하시오.

① EDF(Earliest Deadline First) 알고리즘

* 가장 가까운 데드라인을 가진 태스크에 우선순위 부여
* 동적 우선순위 할당
* 최적의 스케줄링 가능

② RM(Rate Monotonic) 알고리즘

* 주기가 짧은 태스크에서 높은 우선순위 부여
* 정적 우선순위 할당
* 구현이 간단하지만 EDF보다 효율성이 떨어질 수 있음

# 과제 제출

- 2024년 12월 11일(수) 수업시간까지 오프라인 제출