1 스레싱

1 스레싱

1 스레싱의 개념

- 페이지 교환이 계속 일어나는 현상
- 어떤 프로세스에 프레임이 충분하지 않다면 할당된 프레임을 최소 프레임 수까지 줄일 수 있다 하더라도 실제 사용하는 프레임 수만큼 갖지 못하면 빈번하게 페이지 부재가 발생 가능
- 기타 자원 부족으로 필요한 연산을 수행할 수 없는 상태가 되면 운영체제는 다른 프로세스에서 자원을 회수하는 등 방법을 이용하여 자원 확보
- 페이징 동작(스왑 인, 스왑 아웃),
  즉 디스크와 메모리 간에 빈번한 페이지 교환 발생

1 스레싱

1 스레싱의 개념

- 모든 페이지를 실제로 사용하고 있어도 페이지를 교환해야 한다면 페이지 부재 연속 발생
- 프로세스는 계속 페이지를 교환하려고 많은 시간을 낭비하면서 유용한 작업을 느리게 하므로 시스템 성능 낮추거나 축소 야기
- 어떤 프로세스가 프로세스 수행에 보내는 시간보다 페이지 교환에 보내는 시간이 더 길면 '스레싱을 하고 있다'고 표현

1 스레싱

2 스레싱의 발생 원인

- 너무 많은 다중 프로그래밍, 즉 새로운 프로세스가 늘어나면 프로세스에 고정된 프레임 수가 현재 활동에 충분하지 않으므로 서로 프레임을 빼앗기고 뺏는 과정(지속적인 페이지 부재)을 유지하려고 함
- 일부 시스템에서 프로세서 사용률이 낮으면 다중 프로그래밍을 증가해야 한다고 해석하여 악화 야기
- 그러나 전역 페이지 대치에서는 새로운 프로세스가 수행 중인 프로세스의 페이지를 빼앗아서 수행 시작시 더 많은 페이지 부재 발생
- 그러므로 각 프로세스는 자신에게 필요한 만큼 프레임을 배당 받지 못하게 됨

1 스레싱

2 스레싱의 발생 원인

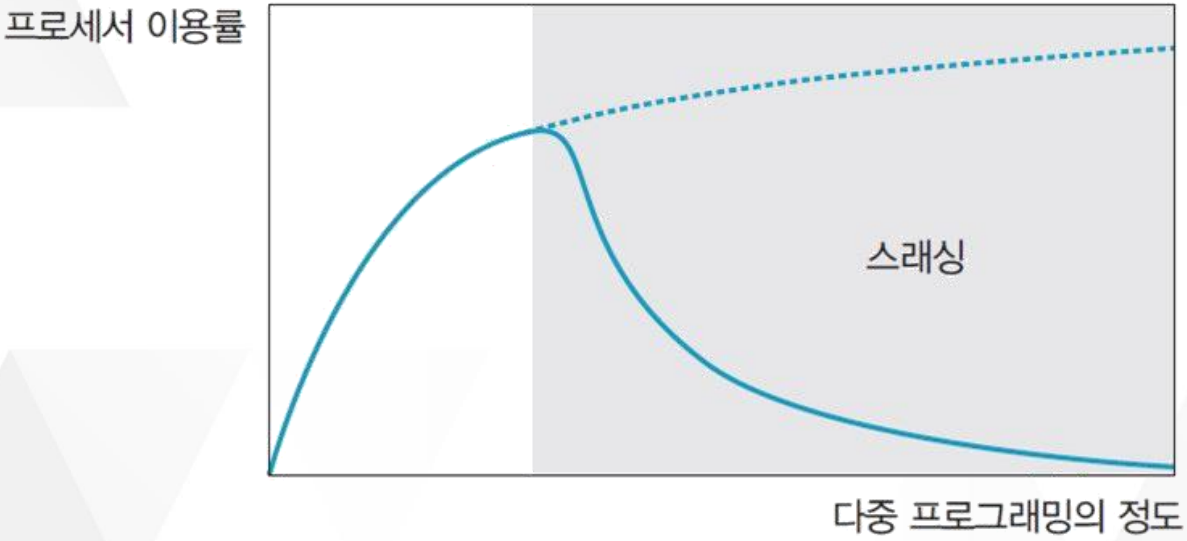
- 프로세서가 요구하는 최소한의 수보다 페이지 프레임 수가 적으면 적을수록 페이지 부재 비율 증가
- 페이지 부재가 발생하면 페이징 처리장치를 사용하려고 프로세스가 대기하기 때 문에 준비 큐는 비게 됨
- 페이지 부재가 많이 발생할수록 프로세스가 페이징 처리장치를 기 다리는 시간은 길어지므로 프로세스의 효율성 낮아짐

1 스레싱

2 스레싱의 발생 원인

- 결국 프로세서의 이용률은 더 낮아지고 프로세서 스케줄러는 이용률을 올리려고 다중 프로그래밍의 정도를 점점 더 높이지만, 프로세스의 실행은 점점 느려짐
- 이런 페이지 부재로 프로세서의 이용률이 감소하면 결국 스레싱 발생

3 프로세서의 이용률에 따른 스레싱 발생률



※출처: 그림으로 배우는 구조와 원리 운영체제, 구현회, 한빛아카데미, 2016

4 스레싱의 예방

- 스레싱 방지 방법
  - 작업 집합(워킹 셋) 모델
    : 프로세스가 실제로 얼마큼 프레임을 많이 사용하는지 검사하여 지역 모델(작업 집합 모델)을 정의
  - 지역 모델
    : 프로세스를 실행할 때 프로그램은 보통 지역 몇 개로 중첩해서 구성하므로 한 지역에서 다른 지역으로 이동하는 과정을 의미함

2 지역성

2 지역성

1 지역성의 개념

- 실행 중인 프로세스에서 나타나는 특성
- 동일한 값이나 관련 저장 위치를 자주 액세스하는 현상
- 프로세스는 어느 실행 단계 동안 메모리의 정보를 균일하게 액세스하는 것이 아니라 선호하는 특정 페이지만 집중적으로 참조하는 현상으로 국부성이라고도 함

2 지역성

1 지역성의 개념

- 잠시 동안 적은 양의 데이터만 집중적으로 참조하다가 데이터의 또 다른 작은 규모의 데이터 덩어리로 이동 경향
- 프로그램들의 순환이나 서브 프로그램, 스택, 변수들의 계산과 합계, 배열 순례, 순 차적 코드의 실행 등으로 발생
- 프로그래머들이 관련 있는 변수들을 서로 근처에 배치 시켜서 발생

2 지역성의 종류

- 시간 지역성
  - 특정 자원들을 상대적으로 짧은 시간 안에 재사용하는 의미로 한 순간 한 지점에서 특정 메모리 위치를 참조할 때 동일한 위치에서 가까운 미래에 다시 참조 할 가능성 높음, 즉 최근에 액세스한 항목은 오래지 않아 다시 액세스할 가능성 있음
  - 동일한 메모리 위치에 인접한 참조 사이에는 시간적 근접성으로 미래 위치가 현재 위치와 동일하다면 시간 지역성은 공간 지역성의 매우 특별한 경우가 되며, 순환(루프)와 서브 프로그램, 스택, 계산이나 합계에 사용하는 변수는 시간 지역성에 적용하는 예

2 지역성의 종류

- 공간 지역성
  - 프로세스가 메모리의 어떤 위치를 참조하면 그 근처를 이후에도 계속 참조 할 가능성 높음
  - 상대적으로 가까운 위치에서 데이터 요소를 사용한다는 것 의미
  - 메모리의 특정 위치를 특정 시간에 참조할 때 가까운 미래에 가까운 메모리 위치 참조 가능성 높음

2 지역성의 종류

- 공간 지역성
  - 데이터 정렬과 1차원 배열의 요소 탐색과 같이 선형 액세스할 때를 순차적 지역성이라고 하며 이는 공간 지역성의 특수한 경우로 배열 검색(순회), 순차적 코드의 실행, 근처의 관련 변수 선언 등이 공간 지역성을 적용한 예

3 작업 집합 모델(Working Set Model)의 개념

- 프로세스가 메모리에서 페이지 부재를 가장 최소 비율로 유지하도록 가장 최근의 계산으로 필요한 메모리를 구하는 방법으로 제안된 모델
- 프로세스가 많이 참조하는 페이지 집합을 메모리 공간에 계속 상주시켜 빈번한 페이지 대치 현상 줄임
- 프로세스들을 메모리에 저장해야 프로세스를 효율적으로 실행 가능하며 그렇지 않으면 빈번한 페이지 대치 작업으로 스레싱 발생

3 작업 집합 모델(Working Set Model)의 개념

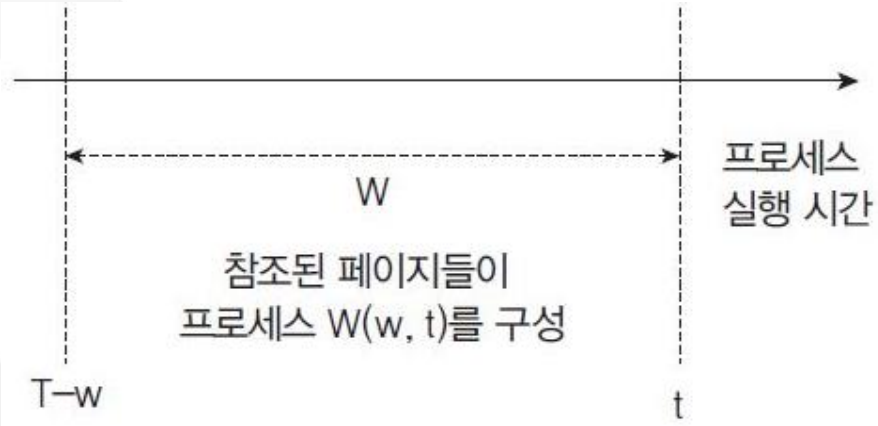
- 프로세스의 작업 집합 모델을 구성하려면 작업 집합의 크기를 알아야 하는데, 작업 집합 크기는 작업 집합 창을 이용하여 구함
- 합창은 매개변수를 사용하여 정의하며, '현재 시간(t)에서 최근의 일정 시간 단위( )'를 정하여 결정
- 작업 집합
  : 가장 최근의 페이지 참조에 있는 페이지 집합

3 작업 집합 모델(Working Set Model)의 개념

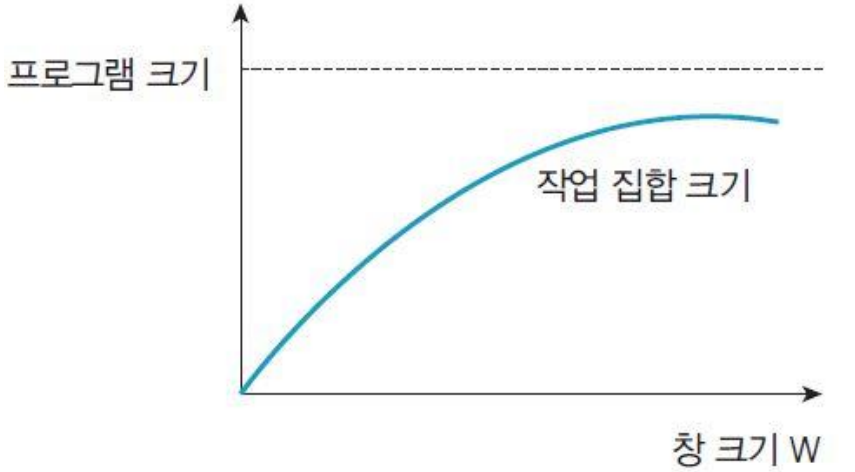
- 작업 집합 모델 관리 방법은 프로세스의 작업 집합 모델 내의 페이지들, 즉 최근에 참조된 페이지들을 메인 메모리에 유지시켜 프로세스를 빠르게 실행함, 그리고 새로운 프로세스들은 메인 메모리에 자신들의 작업 집합을 적재할 수 있는 공간이 있을 때만 시작

3 작업 집합 모델(Working Set Model)의 개념

창 크기와 작업 집합



[창 크기와 작업 집합]



[프로세스의 작업 집합 정의]

※출처: 그림으로 배우는 구조와 원리 운영체제, 구현회, 한빛아카데미, 2016

4 작업 집합 모델을 사용하는 방법

- 운영체제가 각 프로세스의 작업 집합을 감시하고 각 프로세스에 작업 집합 크기에 맞는 충분한 프레임 할당
- 여분의 페이지 프레임이 있을 때는 준비 상태에 있는 다른 프로세스를 불러들인 후 프레임을 할당하여 다중 프로그래밍의 정도를 증가
- 다중 프로그래밍의 정도를 계속 증가시켜서 모든 프로세스가 갖는 작업 집합 크기의 합이 전체 유효 프레임 수보다 커지면 잠시 중지시킬 프로세스를 선정하여 페이지 회수

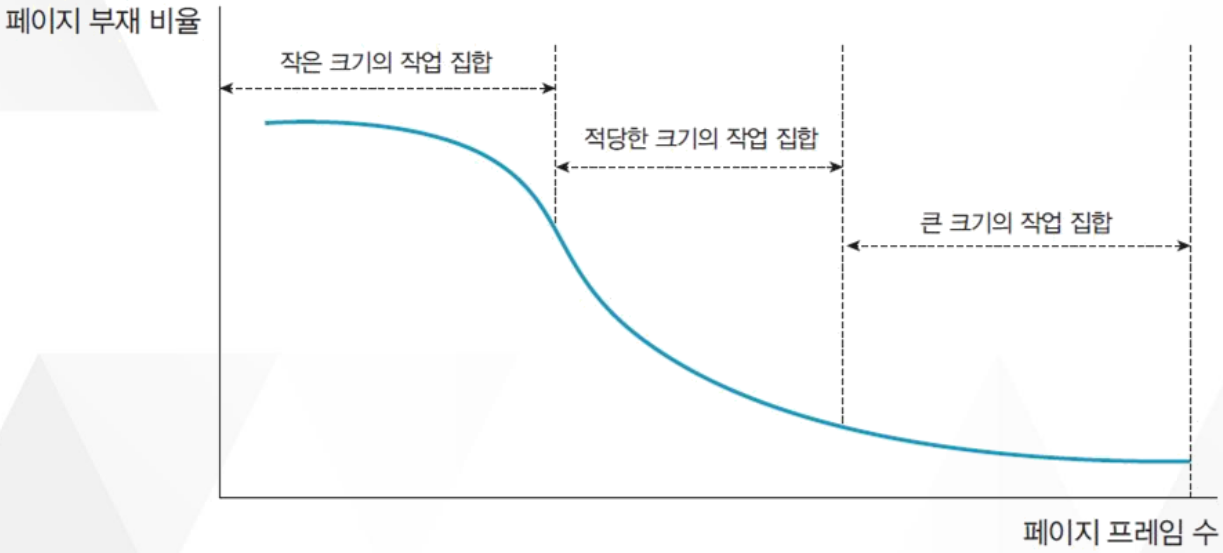
4 작업 집합 모델을 사용하는 방법

- 작업 집합 방법은 가능한 다중 프로그래밍의 정도를 높이면서 스레싱을 방지하는 효과를 제공하여 프로세서의 효율성 최적화하려고 함 (이때 발생하는 문제)
  - 우선 작업 집합이 갖는 과거의 참조가 미래의 참조를 항상 보장하지는 않음
  - 작업 집합 크기와 구성 페이지들은 시간이 경과하면 변함
  - 각 프로세스에서 작업 집합을 모두 측정한다는 것은 현실적으로 불가능함

4 작업 집합 모델을 사용하는 방법

- 작업 집합 방법은 가능한 다중 프로그래밍의 정도를 높이면서 스레싱을 방지하는 효과를 제공하여 프로세서의 효율성 최적화하려고 함 (이때 발생하는 문제)
  - 프로세스를 실행하면서 작업 집합을 삭제·추가하기도 하므로 변화가 심함
  - 각 프로세스가 참조한 페이지 시간과 시간 순서로 된 페이지 큐를 유지해야 하므로 작업 집합으로 메모리를 관리하기가 복잡
  - 작업 집합 창의 크기를 나타내는 매개변수인의 최적 값이 알려져 있지 않고, 처리하는 프로세스의 성격에 따라 의 최적 값은 매우 다양

5 작업 집합 크기에 따른 페이지 프레임 수와 부재 비율 관계



※출처: 그림으로 배우는 구조와 원리 운영체제, 구현회, 한빛아카데미, 2016

3 페이지 부재 빈도

3 페이지 부재 빈도

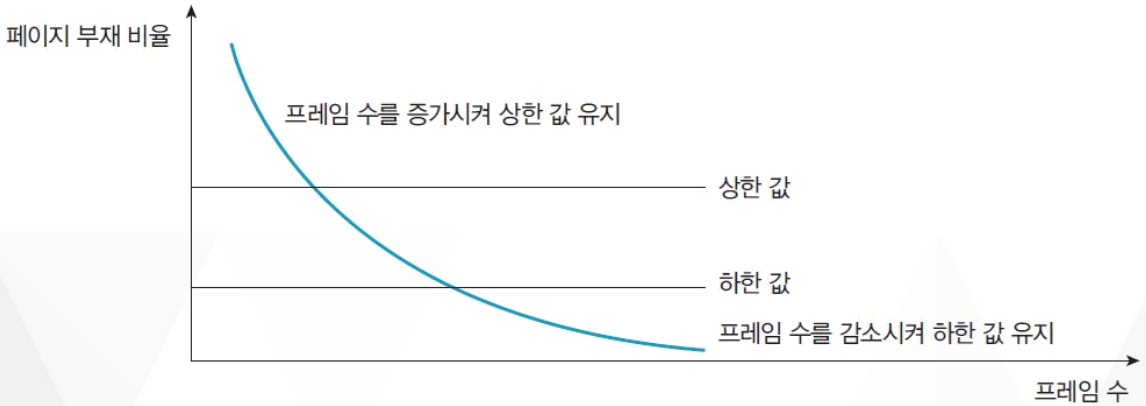
1 페이지 부재 빈도의 개념

- 스레싱을 예방하는 직접적인 액세스 방법
- 페이지 환경에서 프로세스의 실행을 측정하는 기준
- 작업 집합 모델은 페이지가 메모리에 액세스할 때 조절하는 반면에, 페이지 부재 비율은 페이지 부재가 발생할 때 조절하여 작업 집합 모델보다 오버헤드 적음
- 스레싱은 페이지 부재에서 발생하므로 페이지 부재 비율 조절 필요
- 페이지 부재 비율이 높다는 것은 프로세스에 더 많은 프레임이 필요하다는 의미이고, 페이지 부재 비율이 낮다는 것은 프로세스에 프레임이 너무 많다는 의미

3 페이지 부재 빈도

1 페이지 부재 빈도의 개념

[페이지 부재 비율]



※출처: 그림으로 배우는 구조와 원리 운영체제, 구현회, 한빛아카데미, 2016

3 페이지 부재 빈도

1 페이지 부재 빈도의 개념

- 페이지 부재 비율의 상한 값과 하한 값 예측 방법
  - 빈 프레임이 있을 때
    : 페이지 부재 비율이 상한 값을 초과하면 현재 프로세스에 프레임을 할당, 페이지 부재 비율이 하한 값 이하로 떨어지면 현재 프로세스의 프레임을 제거
  - 빈 프레임이 없을 때
    : 페이지 부재 비율이 상한 값을 초과하면 희생(제거)시킬 프로세스 선택하고, 그 프로세스의 실행을 일시 중단, 현재 프로세스의 페이지 부재 비율이 하한 값 이하로 떨어지면 현재 프로세스의 프레임을 제거

3 페이지 부재 빈도

2 프리 페이징 개념

- 처음에 발생하는 많은 페이지 부재를 방지하는 방법
- 예상되는 모든 페이지를 사전에 한꺼번에 메모리 내로 가져옴
- 프로세스가 사용할 페이지를 잘못 결정하거나 페이지 프레임, 디스크 입출력 등 많은 자원을 사용하면 오히려 요구 페이징 시스템보다 프리 페이징의 성능이 더 떨어질 수 있음

3 페이지 부재 빈도

2 프리 페이징 개념

- 프리 페이징에 할당된 메인 메모리 크기와 한 번에 미리 가져올 수 있는 페이지 수, 어떤 페이지를 미리 가져올지 결정할 수 있는 경험적 (공간적·시간적 지역성에 따라 예상하는) 알고리즘이 중요한 요소
- 입출력 인터럽트를 위해 연속된 페이지를 한 번에 메모리로 가져오기 때문에 입출력을 여러 번 수행하는 요구 페이징보다 성능이 좋음
- 비용이 그에 상당하는 페이지 부재를 해결하는데 드는 비용 보다는 적은가 하는 문제가 발생

3 페이지 부재 빈도

3 페이지 크기

- 하드웨어 디자인에서 매우 중요한 요소
- 운영체제를 설계하는 사람에게는 페이지 크기에서 선택권 거의 없지만, 최적 페이지 크기와 관련된 결정 해야 하며, 보통 페이지 크기는 2의 지수승이고, 256(28)에서 4,096(212)바이트나 워드

3 페이지 부재 빈도

3 페이지 크기

- 페이지 크기는 메인 메모리의 크기와 프로그램 크기 자체에 영향, 객체 지향 접근 방법은 많은 수의 개체에 참조가 분산되기 때문에 작은 프로그램과 데이터 모듈 요구되며 다중 스레드 응용 프로 그램은 때때로 명령어 스트림에서 급격한 변화의 원인

- 페이지 크기 결정 시 페이지 테이블의 크기 고려, 가상 메모리 공간이 주어 졌을 때 페이지 크기를 감소시키면 페이지 수가 증가하여 페이지 테이블의 크기도 증가

3 페이지 부재 빈도

3 페이지 크기

- 메모리는 크기가 작은 페이지가 이용 용이
- 내부 단편화를 최소화하려면 크기가 작은 페이지 필요
- 입출력 시간을 최소화시키려면 페이지 크기를 더 크게 하는 것이 유리
- 페이지 크기는 컴퓨터 종류별로 매우 다양
  - 프로세서의 속도와 메모리 용량, 디스크 속도가 증가하면서 오늘날 페이지 크기는 더 커짐

3 페이지 부재 빈도

3 페이지 크기

- 페이지 크기를 증가시키는 것이
  페이지 부재의 빈도를 줄이는 데 더 유리
- 페이지 크기가 증가하면 내부 단편화 증가

[페이지 크기별 특징]

작은 페이지	큰 페이지
• 페이지 테이블의 크기 증가 • 내부 단편화 감소 • 디스크 입출력 증가 • 지역성 증가와 페이지 부재 비율 증가	• 페이지 테이블의 크기 감소 • 내부 단편화 증가 • 디스크 입출력 감소 • 지역성 악화와 페이지 부재 비율 감소

※출처: 그림으로 배우는 구조와 원리 운영체제, 구현회, 한빛아카데미, 2016