1. **가상 메모리 개관**

개관

가상 메모리로 실제 메모리(주기억장치)의 용량보다 더 많은 양의 기억공간이 필요한 프로그램을 실행 학습 목표

1. **가상 메모리의 개념과 주소변환 과정을 이해**
2. **페이징 기법과 세그먼테이션 기법에서 주소변환 과정을 이해**
3. **요구 페이지 호출기법과 예상 페이지 호출기법을 이해 주요 용어**

가상 주소 사상

동적 주소변환 페이징 기법 세그먼테이션 기법

* 1. **가상 메모리의 개념 [131]**

**가상메모리 (virtual memory)**

컴퓨터 시스템의 메모리 크기보다 더 큰 기억 공간이 필요한 프로세스를 실행할 수 있게 하는 방법 연속 메모리 할당방식으로는 메모리 크기보다 더 큰 기억 공간이 필요한 프로세스를 실행 시킬 수 X 핵심

실행 중인 프로세스에 의해 참조되는 주소(가상주소)를 메모리에서 사용하는 주소(실주소)와 분리하는 것 전체 프로그램 및 데이터 중 현재 필요한 일부만 메모리에 적재함으로써 프로세스의 수행이 가능

가상 주소와 실 주소

**가상 주소(virtual address)**

실행 프로세스가 참조하는 주소

**실주소(real address) / 물리적 주소(physical address)**

실제 메모리에 사용하는 주소

**가상주소 공간(virtual address space) V**

**실행 프로세스가 참조하는 가상주소의 범위 실주소 공간(real address space) R**

**특정 컴퓨터 시스템에서 사용 가능한 실주소의 범위 사상(mapping)**

1) 프로세스는 오직 가상주소만을 참조하지만, 2) 실제 실행되는 곳은 메모리이므로 3) 프로세스가 실행되면 가상주소는

실주소로 변환 => 변환함수로 표시

**운영체제가 가상주소를 실주소로 변환시키는 일을 수행 동적 주소변환 (Dynamic Address Translation: DAT)**

프로세스가 실행되는 동안 가상주소를 실주소로 바꾸는 절차 인위적 연속성

프로세스의 가상주소 공간에서 연속적인 주소가 실주소 공간에서도 연속적일 필요는 없다.

배경

사용자의 가상주소 공간이 실주소 공간보다 커짐 여러 사용자가 메모리를 공유

이에 따른 효율적인 다중 프로그래밍 시스템 구축을 위해 필요

* 1. **블록 단위 주소변환 [133]**

**동적 주소변환(DAT) 방법**

**가상 메모리에서의 위치가 현재 메모리의 어디에 위치하는지를 나타내는 "주소변환 사상표(mapping table)"을 유지해야 한 다.**

**주소변환이 가상주소 내의 각 항목별로, 즉 바이트(byte)나 워드(word) 단위로 이루어진다면?**

프로세스 요구 메모리 공간보다 사상표가 더 큰 메모리 공간이 필요 블록 사상 시스템

블록 사상 시스템

항목 대신 "블록" 단위로 주소변환이 이뤄지며 각 블록이 메모리의 어디에 위치하는지만 관리

구성

**블록 변위(displacement)**

가상주소 v는 참조될 항복이 있는 블록의 번호 b와 블록의 시작 부분으로부터의 변위 d로 구성된 순서쌍 (b,d)에 의해 지정

블록의 크기는 적절히 설정해야 한다. 블록의 크기가 커질수록

사상정보 저장에 필요한 메모리 양 ↓

페이지 교체 전송시간 ↑

메모리 공유 프로세스 수 ↓

블록의 크기가 작아질수록

블록 전송시간(페이지 교체 전송시간) ↓

메모리 공유 프로세스 수 ↓

사상 정보 저장에 필요한 메모리 양 ↑

**페이지(page)**

블록의 크기가 동일할 때 그 블록을 일컫는 말 페이징 기법

**세그먼트(segment)**

**크기가 서로 다를 수 있는 블록 세그먼테이션(segmentation) 기법**

**9.2 블록 단위 주소변환 (2) [134]**

페이징 기법

**가상 메모리를 고정된크기의 블록, 가상주소 v = (p, d)**

**페이지 번호 : p**

**즉 페이지(page) 단위로 나누어 관리**

페이지 p의 시작 부분으로부터의 변위 : d

예시

**가상주소 v = (3, 8)**

**가상 메모리 상 a 위치는 3번 페이지 안, 3번 페이지 시작 위치로부터 8byte 만큼 떨어져 있다. 만약 페이지 크기가 1024byte라면?**

# v = 3 x 1024 + 8 = 3080

**변위에 10bit이 사용되므로 가상주소 v는?**

# 11 0000001000 (2)

**페이지 프레임(page frame)**

프로세스가 실행하기 위해 특정 페이지를 참조하려면 해당 페이지는 메모리상에 위치해야만 한다.

(실) 메모리 영역도 가상 메모리와 동일하게 고정된 크기의 블록으로 나누어 두고, 각 블록을 page frame이라고 한다 가상 메모리상의 페이지를 담을 수 있도록 실제 메모리에 틀(frame)을 만들어 둔 것

비어 있는 프레임 아무 곳에나 적재 가능 페이징 사상표

페이지가 메모리에 적재된 후에도 바로 찾을 수 있도록 프로세스가 사용하는 가상주소를 실주소로 동적 변환 할 수 있어야 함 구성

페이지 존재 비트

해당 페이지가 현재 (실)메모리에 존재하는지 여부를 나타내는 비트값 보조기억장치 주소

존재하지 않을 때 보조기억장치의 어디에 있는지에 대한 정보

(실주소의) 페이지 프레임 번호

가상주소의 페이지 번호에 대한 실주소의 페이지 프레임 번호가 저장

**[그림 9-8 참조] a의 위치를 찾기 위해 프로세스가 가진 가상주소 (3,8)을 통해 동적 주소변환을 한다면? (단, 페이지 프레임 크기가 1024byte)이다**

# 5 X 1024 + 8 byte

직접사상

페이지 사상표를 직접 이용하여 동적 주소변환을 하는 것 연관사상

페이지 변환 정보를 연관 메모리에 저장한 연관사상표를 이용하여 동적 주소변환을 하는 방식 연관 메모리

저장된 값을 이용하여 데이터를 액세스하는 고속 메모리 장치 직접 사상과 다르게

**1) 인덱스 필요 X 2) 메모리 블록의 어떤 정보도 포함X 연관/직접 사상**

1. **연관사상표에는 가장 최근에 참조된 페이지 항목만 보관 2) 나머지는 페이지 사상표에 수록하여, 3) 연관사상표에 없을 때 만 직접 사상을 이용하도록 구현**

**연관/직접 사상의 동적 주소변환 과정 [138]**

**실행 프로그램은 가상주소 v=(p,d)를 참조**

* 1. **우선 연관 사상표에서 가상 페이지 p를 찾는다.**
  2. **연관 사상표에 페이지 p에 대응되는 항목이 없다면 메모리의 직접 페이지 사상표를 탐색 페이지 사상표의 시작주소 b가 p와 더해져 메모리의 직접 페이지 사상표에 있는 p에 해당하는 페이지 프레임 p'를 지시. p'은 변위 d와 합하여 가 상주소 v = (p, d)에 대응하는 실주소 r을 생성**
  3. **하드웨어는 최근에 가장 적게 사용된 항목 대신 이 항목을 연관사상표에 포함하도록 갱신**

특징

페이징 기법에서 프로세스 사이의 메모리 보호는 ( ) 단위로 이뤄진다. 페이지 단위

외부 단편화가 발생하지 않는다.

메모리도 (가상 메모리와) 동일 크기의 페이지 프레임으로 나뉘어 사용되므로 페이지 내에 내부 단편화는 발생할 수 있다.

**마지막 부분 동적 주소변환 (3) [139]**

세그먼테이션 기법

**논리적 크기에 부합하는 다양한 크기의 블록, 즉 세그먼트(segment) 단위로 가상 메모리를 나누어 관리하는 기법 가상주소 v는 순서쌍 (s, d)**

세그먼트 번호 s

세그먼트 s의 시작 위치로부터 변위 d

**예 - 가상주소 v = (3, 8)에서 가상 메모리상에서 a의 위치 세그먼트를 메모리에 적재하는 법**

자신을 수용할 수 있을 만큼 충분히 큰 메모리의 사용 가능한 영역에 두기

**동적 분할 프로그래밍에서 흔히 사용되는 최초 적합(first-fit), 최적 적합(best-fit) 등의 방법과 동일 세그먼테이션 기법에서 동적 주소 변환**

세그먼트 사상표 구성

해당 세그먼트가 현재 메모리에 존재하는지 여부를 나타내는 비트값 보조기억장치 주소

세그먼트 길이

세그먼트 오버플로우 확인용 세그먼트 시작 주소

가상주소의 세그먼트 번호에 대한 실주소에서의 시작 위치 저장 페이징/세그먼테이션 혼용기법

1. **각 세그먼트를 다시 페이지 단위로 분할하고 2) 메모리도 페이지 프레임으로 분할하여 3) 하나의 페이지만 페이지 프레임에 적재하는 방식**

# v = (s, p, d)

세그먼트 번호 s

세그먼트의 페이지 번호 p

항목이 위치하는 페이지 내의 변위 d

예제

**실행 프로세스는 가상주소 v = (s, p, d)를 참조**

* 1. **가장 최근에 참조된 페이지는 연관사상표에 있으므로 연관 메모리에서 (s, p) 위치를 찾기 위한 탐색이 수행된다.**

**(s, p)가 발견되면 세그먼트 s의 페이지 p에 대응하는 메모리에 위치하는 페이지 프레임 p'이 변위 d와 합하여 가상주 소 v = (s, p, d)에 대응되는 실주소 r을 생성함으로써 주소변환이 종료된다.**

* 1. **요구가 연관사상표에 없는 경우 직접사상 변환은 다음과 같다.**

메모리의 세그먼트 사상표의 시작주소 b가 세그먼트 번호 s에 더해져서 세그먼트 s에 대한 세그먼트 사상표의 항목주

소 b + s를 생성한다. 세그먼트 사상표의 항목은 세그먼트 s에 대한 페이지 사상표의 시작주소 s'를 가리킨다. 페이지

번호 p가 s'과 더해져서 세그먼트 s의 페이지 p에 대한 페이지 사상표의 항목주소 p + s'를 생성한다. 이제 페이지 사 상표의 p'이 가상 페이지 p에 대응하는 페이지 프레임 번호라는 사실을 알려 준다. 페이지 프레임 번호 p'은 변위 d와 합쳐져서 가상주소 v = (s, p, d)에 사상하는 실주소 r을 생성한다.

* 1. **메모리 호출기법 [143]**

메모리 호출기법

어느 시점에 페이지 또는 세그먼트를 메모리에 적재할 것인가를 결정하는 기법

* + 1. **요구 페이지 호출기법 (demand page fetch strategy)**

한 프로세스의 페이지 요구가 있을 때 비로소 요구 페이지를 메모리에 적재하는 방법 실행 중인 프로세스에 의해 명백히 참조될 때에만 보조기억장치에서 메모리로 옮겨진다. 특징

오버헤드를 최소화

메모리에 옮겨진 페이지는 모두 프로세스에 의해 실제로 참조된 것임을 확신

**but 페이지 부재 자주 발생**

새로운 프로세스가 시작되는 시점에 순수한 요구 페이지 호출기법을 적용하는 경우라면 메모리에는 첫 번째 페이지만 적재 된다.

이후 프로세스의 실행이 진행됨에 따라 연속으로 페이지 부재(page fault)가 발생하며, 이에 따라 해당 페이지가 메모리 로 옮겨지므로 항상 프로세스 시작 시점에 성능이 저하되기 때문이다.

* + 1. **예상 페이지 호출기법 (prepaging fetch strategy)**

현 시점에서 액세스되고 있지만 않지만 곧 사용될 것으로 예상되는 페이지를 미리 메모리에 옮겨 놓는 방법이다. 특징

예상이 옳았다면 실제로 필요한 시점이 되었을 때 프로세스의 실행이 단절되지 않아 실행시간이 감소된다. 예상이 잘못된 경우에는 예상 적재에 따른 시간과 메모리 공간이 낭비된다.

요약

1. **가상 메모리는 메모리 크기보다 더 큰 기억공간을 사용하는 프로세스를 실행할 수 있다.**
2. **프로세스에서 사용되는 가상주소는 동적 주소변환을 통해 메모리의 실주소로 변환된다.**
3. **연속적인 가상주소가 실주소 공간에서도 연속적일 필요는 없다.**
4. **페이징 기법은 페이지라는 고정된 크기의 블록 단위로 기억장치를 관리하는 기법이다.**
5. **세그먼테이션 기법은 모듈화에 따른 논리적 의미에 부합하는 다양한 크기의 세그먼트 단위로 기억장치를 관리하는 기법이다.**
6. **요구 페이지 호출기법은 페이지가 필요한 시점에 메모리에 적재하는 방법이다.**
7. **예상 페이지 호출기법은 앞으로 사용될 것으로 예상되는 페이지를 미리 메모리에 적재하는 방법이다.**