4과목-프로그래밍 언어 활용

(Part 3. 응용 SW 기초 기술 활용-2)

프로그래밍 언어 활용 총 파트

프로그래밍 언어 활용 4과목은 총 3Part로 이루어져 있다.

1장 서버 프로그램 구현(0.69%)

2장 프로그래밍 언어 활용(44.83%)

3장 응용 SW 기초 기술 활용(54.48%)

프로그래밍 언어 활용

응용 SW 기초 기술 활용 Part는 17개의 섹션으로 구성되어 있다.

001 운영체제의 개념
002 Windows
003 UNIX / LINUX/ MacOS
004 기억장치 관리의 개요
005 주기억장치 할당 기법 등급
006 가상기억장치 구현 기법/페이지 교체 알고리즘
007 가상기억장치 기타 관리 사항
008 프로세스의 개요
009 스케줄링
010 주요 스케줄링 알고리즘
011 환경 변수
012 운영체제 기본 명령어
013 인터넷
014 OSI 참조 모델
015 네트워크 관련 장비
016 프로토콜의 개념
017 TCP/IP

4.응용 SW 기초기술 활용-SEC_06(가상기억장치 구현 기법 / 페이지 교체 알고리즘)

1) 가상기억장치의 개요

; 가상기억장치는 보조기억장치(SSD(Solid State Disk), HDD(Hard Disk Driver))의 일부를 주기억장치처럼 사용하는 것으로, 용량이 작은 주기억장치를 마치 큰 용량을 가진 것처럼 사용하는 기법이다.

- 프로그램을 여러 개의 작은 **블록** 단위로 나누어서 가상기억장치에 보관해 놓고, 프로그램 실행 시 요구되는 블록만 주기억장치에 불연속적으로 할당하여 처리한다.
- 주기억장치의 용량보다 큰 프로그램을 실행하기 위해 사용한다.
- 주기억장치의 이용률과 다중 프로그래밍의 효율을 높일 수 있다.
- 가상기억장치에 저장된 프로그램을 실행하려면 가상기억장치의 주소를 주기억장치의 주소로 바꾸는 주소 변환 작업(매핑(Mapping)이 필요하다.
- 블록 단위로 나누어 사용하므로 연속 할당 방식에서 발생할 수 있는 단편화를 해결할 수 있다.
- 가상기억장치의 일반적인 구현 방법에는 블록의 종류에 따라 페이징 기법과 세그먼테이션 기법으로 나눌 수 있다.

블록 : 블록은 보조기억장치와 주기억장치 간에 전송되는 데이터의 최소 단위이다.
주소 변환 : 가상기억장치에 있는 프로그램이 주기억장치에 적재되어 실행될 때 논리적인 가상주소를 물리적인 실기억 주소로 변환하는 것으로 주소 매핑(Mapping)이라고도 한다. 이 때 연속적인 가상주소가 반드시 연속적인 실기억 주소로 변환되지 않아도 되는데, 이를 **인위적 연속성(Artificial Contiguity)**이라고 한다.

4.응용 SW 기초기술 활용-SEC_06(가상기억장치 구현 기법 / 페이지 교체 알고리즘)

2) 페이징(Paging) 기법

; 페이징 기법은 가상기억장치에 보관되어 있는 프로그램과 주기억장치의 영역을 동일한 크기로 나눈 후 나눠진 프로그램(페이지)을 동일하게 나눠진 주기억장치의 영역(페이지 프레임)에 적재시켜 실행하는 기법이다.

- 프로그램을 일정한 크기로 나눈 단위를 페이지(Page)라고 하고, 페이지 크기로 일정하게 나누어진 주기억장치의 단위를 페이지 프레임(Page Frame)이라고 한다.
- 외부 단편화는 발생하지 않으나 내부 단편화는 발생할 수 있다.
- 주소 변환을 위해서 페이지의 위치 정보를 가지고 있는 페이지 맵 테이블(Page Map Table)이 필요하다.
- 페이지 맵 테이블 사용으로 비용이 증가되고, 처리 속도가 감소된다.

페이지의 크기 : 일반적으로 페이지의 크기는 1 ~ 4KB 이다.
내부 단편화는 발생할 수 있다는 것은 페이지 크기가 4KB이고, 사용할 프로그램이 17KB라면 프로그램은 페이지 단위로 4KB씩 나누어지게 된다. 이때 마지막 페이지의 실제 용량은 1KB(17KB-16KB)가 되고 이것이 주기억장치에 적재되면 3KB의 내부 단편화가 발생된다.

4.응용 SW 기초기술 활용-SEC_06(가상기억장치 구현 기법 / 페이지 교체 알고리즘)

3) 세그먼테이션(Segmentation) 기법

; 세그먼테이션 기법은 가상기억장치에 보관되어 있는 프로그램을 다양한 크기의 논리적인 단위로 나눈 후 주기억장치에 적재시켜 실행시키는 기법이다.

- 프로그램을 배열이나 함수 등과 같은 논리적인 크기로 나눈 단위를 세그먼트(Segment)라고 하며, 각 세그먼트는 고유한 이름과 크기를 갖는다.
- 기억장치의 사용자 관점을 보존하는 기억장치 관리 기법이다.
- 세그먼테이션 기법을 이용하는 궁극적인 이유는 기억공간을 절약하기 위해서이다.
- 주소 변환을 위해서 세그먼트가 존재하는 위치 정보를 가지고 있는 세그먼트 맵 테이블(Segment Map Table)이 필요하다.
- 세그먼트가 주기억장치에 적재될 때 다른 세그먼트에게 할당된 영역을 침범할 수 없으며, 이를 위해 기억장치 보호키(Storage Protection Key)가 필요하다.
- 내부 단편화는 발생하지 않으나 외부 단편화는 발생할 수 있다.

4.응용 SW 기초기술 활용-SEC_06(가상기억장치 구현 기법 / 페이지 교체 알고리즘)

3) 세그먼테이션(Segmentation) 기법

세그먼테이션 기법의 일반적인 주소 변환

● 주소 형식에 따른 주소와 세그먼트 맵 테이블의 구성

- 가상주소는 세그먼트 번호를 나타내는 s와 세그먼트 내에 실제 내용이 위치하고 있는 곳까지의 거리를 나타내는 변위값 d로 구성된다.

가상주소 형식

세그먼트 번호(s)	변위값(d)

- 실기억 주소는 완전 주소 형태를 사용하며 이는 세그먼트의 기준번지와 변위값을 더함으로써 얻을 수 있다.

실기억 주소 형식

실기억주소(세그먼트 기준번지 + 변위값)

- 세그먼트 맵 테이블(Segment Map Table)은 세그먼트 번호 s와 세그먼트의 크기 L(한계번지), 주기억장치 상의 기준번지(시작 주소) b로 구성된다.

4.응용 SW 기초기술 활용-SEC_06(가상기억장치 구현 기법 / 페이지 교체 알고리즘)

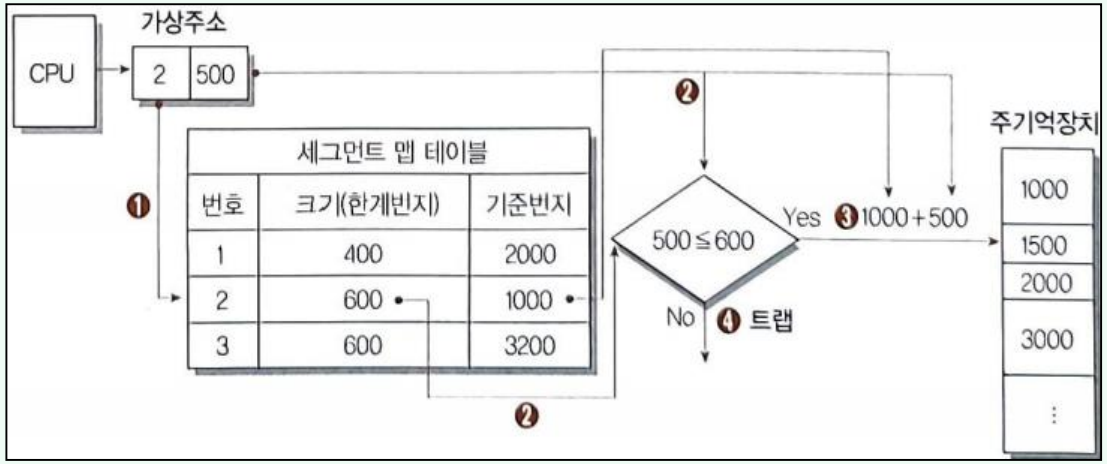
3) 세그먼테이션(Segmentation) 기법

세그먼테이션 기법의 일반적인 주소 변환

● 주소 변환 순서

세그먼트 맵 테이블

세그먼트 번호(s)	세그먼트 크기(L)	기준번지(b)



① 가상주소의 세그먼트 번호로 세그먼트 맵 테이블에서 해당 세그먼트의 기준번지와 세그먼트 크기를 구한다. 세그먼트 번호는 세그먼트 맵 테이블에 대한 색인으로 사용된다.

② 가상주소의 변위값과 세그먼트의 크기를 비교한다.

③ 변위값이 작거나 같으면 기준번지와 변위값을 더하여 실기억주소를 만들어 주기억장치를 액세스 한다.

④ 변위값이 크면 다른 영역을 침범하게 되므로 실행 권한을 운영체제에게 넘기고 트랩을 발생시킨다 (변위값이 크다는 것은 현재 찾는 세그먼트의 위치가 해당 세그먼트의 크기(한계번지)를 초과하였다는 의미이다).

trap : 세그먼트의 크기보다 크다면 메모리 오류를 출력하고 해당 프로세스를 강제 종료한다. 이 때 발생하는 오류를 trap이라고 한다.

변위값 : 가상주소에 있는 주소값을 변위값으로 보면 된다.

4.응용 SW 기초기술 활용-SEC_06(가상기억장치 구현 기법 / 페이지 교체 알고리즘)

4) 페이지 교체 알고리즘

; 페이지 교체 알고리즘은 페이지 부재(Page Fault)가 발생했을 때 가상기억장치의 필요한 페이지를 주기억장치에 적재해야 하는데, 이때 주기억장치의 모든 페이지 프레임이 사용 중이면 어떤 페이지 프레임을 선택하여 교체할 것인지를 결정하는 기법이다.

● 페이지 교체 알고리즘에는 OPT, FIFO, LRU, LFU, NUR, SCR 등이 있다.

① OPT(OPTimal replacement, 최적 교체)

▶ OPT는 앞으로 가장 오랫동안 사용하지 않을 페이지를 교체하는 기법이다.

▶ 벨레이디(Belady)가 제안한 것으로, 페이지 부재 횟수가 가장 적게 발생하는 가장 효율적인 알고리즘이다.

페이지 부재(Page Fault) : 페이지 부재는 CPU가 액세스한 가상 페이지가 주기억장치에 없는 경우를 말한다. 페이지 부재가 발생하면 해당 페이지를 디스크에서 주기억장치로 가져와야 한다.

4.응용 SW 기초기술 활용-SEC_06(가상기억장치 구현 기법 / 페이지 교체 알고리즘)

4) 페이지 교체 알고리즘

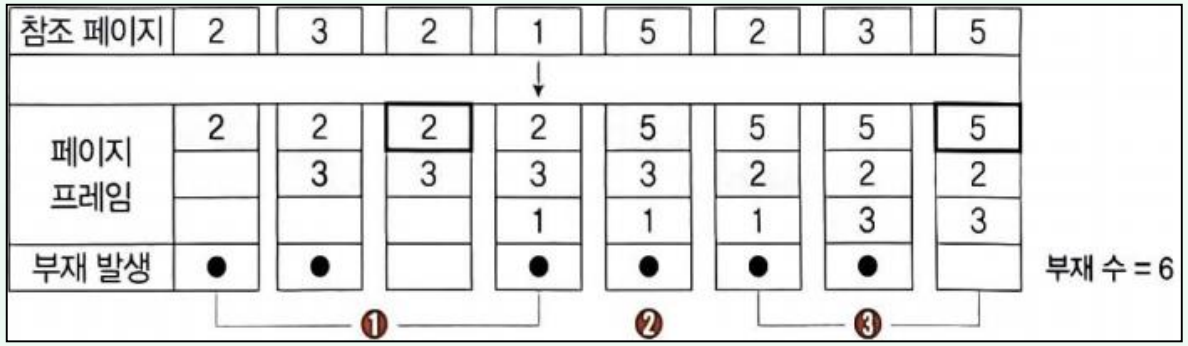
● 페이지 교체 알고리즘에는 OPT, FIFO, LRU, LFU, NUR, SCR 등이 있다.

② FIFO(First In First Out)

▶ FIFO는 각 페이지가 주기억장치에 적재될 때마다 그때의 시간을 기억시켜 가장 먼저 들어와서 가장 오래 있었던 페이지를 교체하는 기법이다.

▶ 이해하기 쉽고, 프로그래밍 및 설계가 간단하다.

예제) 다음의 참조 페이지를 세 개의 페이지 프레임을 가진 기억장치에서 FIFO 알고리즘을 사용하여 교체했을 때 페이지 부재의 수는? (단, 초기 페이지 프레임은 모두 비어 있는 상태이다.)



① 참조 페이지를 각 페이지 프레임에 차례로 적재시키되 이미 적재된 페이지는 해당 위치의 페이지 프레임을 사용한다.
② 사용할 페이지 프레임이 없을 경우 가장 먼저 들어와서 오래 있었던 페이지 2를 제거한 후 5를 적재한다.
③ 그 다음에 적재된 페이지 3을 제거한 후2를 적재하며, 같은 방법으로 나머지 참조 페이지를 수행한다.

4.응용 SW 기초기술 활용-SEC_06(가상기억장치 구현 기법 / 페이지 교체 알고리즘)

4) 페이지 교체 알고리즘

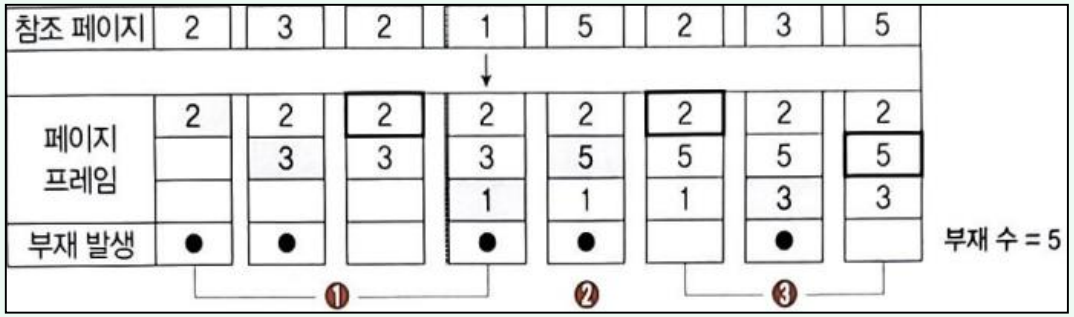
● 페이지 교체 알고리즘에는 OPT, FIFO, LRU, LFU, NUR, SCR 등이 있다.

③ LRU(Least Recently Used)

▶ LRU는 최근에 가장 오랫동안 사용하지 않은 페이지를 교체하는 기법이다.

▶ 각 페이지마다 계수기(Counter)나 스택(Stack)을 두어 현 시점에서 가장 오랫동안 사용하지 않은 즉, 가장 오래 전에 사용된 페이지를 교체한다.

예제) 다음의 참조 페이지를 세 개의 페이지 프레임을 가진 기억장치에서 LRU 알고리즘을 사용하여 교체했을 때 페이지 부재의 수는? (단, 초기 페이지 프레임은 모두 비어 있는 상태이다.)



① 참조 페이지를 각 페이지 프레임에 차례로 적재시키되 이미 적재된 페이지는 해당 위치의 페이지 프레임을 사용한다.
② 사용할 페이지 프레임이 없을 경우 현재 시점에서 가장 오랫동안 사용되지 않은 페이지 3을 제거한 후 5를 적재한다.
③ 같은 방법으로 나머지 참조 페이지를 수행한다.

계수기(Counter) : 계수기는 각 페이지 별로 존재하는 논리적 시계(Logical Clock)로, 해당 페이지가 사용될 때마다 0으로 클리어 시킨 후 시간을 증가시켜서 시간이 가장 오래된 페이지를 교체한다.

4.응용 SW 기초기술 활용-SEC_06(가상기억장치 구현 기법 / 페이지 교체 알고리즘)

4) 페이지 교체 알고리즘

● 페이지 교체 알고리즘에는 OPT, FIFO, LRU, LFU, NUR, SCR 등이 있다.

④ LFU(Least Frequently Used)

▶ LFU는 사용 빈도가 가장 적은 페이지를 교체하는 기법이다.

▶ 활발하게 사용되는 페이지는 사용 횟수가 많아 교체되지 않고 사용된다.

⑤ NUR(Not Used Recently)

▶ NUR은 LRU와 비슷한 알고리즘으로, 최근에 사용하지 않은 페이지를 교체하는 기법이다.

▶ 최근에 사용되지 않은 페이지는 향후에도 사용되지 않을 가능성이 높다는 것을 전제로, LRU에서 나타나는 시간적인 오버헤드를 줄일 수 있다.

▶ 최근의 사용 여부를 확인하기 위해서 각 페이지마다 두 개의 비트, 즉 참조 비트(Reference Bit)와 변형 비트(Modified Bit, Dirty Bit)가 사용된다.

▶ 다음과 같이 참조 비트와 변형 비트의 값에 따라 교체될 페이지의 순서가 결정된다.

참조 비트	0	0	1	1
변형 비트	0	1	0	1
교체 순서	1	2	3	4

참조 비트(Reference Bit) : 참조 비트는 페이지가 호출되지 않았을 때는 0, 호출되었을 때는 1로 지정됨
변형 비트(Modified Bit, Dirty Bit) : 변형 비트는 페이지 내용이 변경되지 않았을 때는 0, 변경되었을 때는 1로 지정됨

4.응용 SW 기초기술 활용-SEC_06(가상기억장치 구현 기법 / 페이지 교체 알고리즘)

4) 페이지 교체 알고리즘

● 페이지 교체 알고리즘에는 OPT, FIFO, LRU, LFU, NUR, SCR 등이 있다.

⑥ SCR(Second Chance Replacement, 2차 기회 교체)

▶ SCR은 가장 오랫동안 주기억장치에 있던 페이지 중 자주 사용되는 페이지의 교체를 방지하기 위한 것으로, FIFO 기법의 단점을 보완하는 기법이다.

응용 SW 기초기술 활용 - SEC_06(가상기억장치 구현 기법/페이지 교체 알고리즘) 기출 및 출제 예상 문제

기출 및 출제 예상 문제(가상기억장치 구현 기법/페이지 교체 알고리즘)

1. 다음 중 페이지 교체(Page Replacement) 알고리즘이 아닌 것은?

① FIFO(First-In-First-Out)

② LUF(Least Used First)

③ OPTimal replacement

④ LRU(Least Recently Used)

LUF(Least Used First)라는 페이지 교체 알고리즘은 존재하지 않는다.

페이지 교체 알고리즘

페이지 교체 알고리즘은 페이지 부재(Page Fault)가 발생했을 때 가상 기억장치의 필요한 페이지를 주기억장치에 적재해야 하는데, 이 때 주기억장치의 모든 페이지 프레임 사용 중이라면 어떤 페이지 프레임을 선택하여 페이지 교체를 할 것인지를 결정하는 기법이다.

▶ 페이지 교체 알고리즘에는 **OPT, FIFO, LRU, LFU, NUR, SCR** 등이 있다.

2. 페이징 기법과 세그먼테이션 기법에 대한 설명으로 옳지 않은 것은?

① 페이징 기법에서는 주소 변환을 위한 페이지 맵 테이블이 필요

3. 다음 설명의 ㉠과 ㉡에 들어갈 내용으로 옳은 것은?

가상기억장치의 일반적인 구현 방법에는 프로그램을 고정된 크기의 일정한 블록으로 나누는 (㉠) 기법과 가변적인 크기의 블록으로 나누는 (㉡)기법이 있다.

① ㉠ : Paging, ㉡ : Segmentation

② ㉠ : Segmentation, ㉡ : Allocation

③ ㉠ : Segmentation, ㉡ : Compaction

④ ㉠ : Paging, ㉡ : Linking

고정된 크기라면 페이징이고, 가변(논리)적 크기라면 세그먼테이션 기법이다.

4. 세그먼테이션 기법에 대한 설명으로 옳지 않은 것은?

① 각 세그먼트는 고유한 이름과 크기를 갖는다.

② 세그먼트 맵 테이블이 필요하다.

③ 프로그램을 일정한 크기로 나눈 단위를 세그먼트라고 한다.

④ 기억장치 보호키가 필요하다.

프로그램을 일정한 크기로 나눈 단위는 페이지이며, 논리(가변)적인 크기로 나눈 단위는 세그먼트이다.

응용 SW 기초기술 활용 - SEC_06(가상기억장치 구현 기법/페이지 교체 알고리즘) 기출 및 출제 예상 문제

기출 및 출제 예상 문제(가상기억장치 구현 기법/페이지 교체 알고리즘)

5. 다음과 같은 세그먼트 테이블을 가지는 시스템에서 논리 주소 (2, 176)에 대한 물리 주소는?

세그먼트 번호	시작주소	길이(바이트)
0	670	248
1	1752	422
2	222	198
3	996	604

① 398 ② 400

③ 1928 ④ 1930

실기억 주소는 세그먼트의 기준번지 + 변위값이다.

논리 주소(2, 176)에서 2는 세그먼트 번호이고, 176이 변위값이다.

세그먼트 번호 2의 시작 주소인 222에 변위값 176을 더하면 물리적인 실기억장치의 주소가 된다.

세그먼테이션 기법의 일반적인 주소 변환 순서

① 가상주소의 세그먼트 번호로 세그먼트 맵 테이블에서 해당 세그먼트의 기준 번지(시작 주소)와 세크먼트의 크기를 구한다.

7. 가상기억장치에서 주기억장치로 페이지를 옮겨 넣을 때 주소를 조정해 주어야 하는데 이를 무엇이라 하는가?

① 매핑(Mapping)

② 스케줄링(Scheduling)

③ 매칭(Matching)

④ 로딩(Loading)

가상주소로부터 물리 주소를 찾는 것을 사상(Mapping)이라고 한다.

가상기억장치에 대한 내용

가상기억장치는 보조기억장치(SSD, HDD)의 일부를 주기억장처럼 사용하는 것으로, 용량이 작은 주기억장치를 마치 큰 용량을 가진 것처럼 사용하는 기법이다.

▶ 프로그램을 여러 개의 작은 블록 단위로 나누어서 가상기억장치에 보관해 놓고, 프로그램 실행 시에 요구되는 블록만 주기억장치에 불연속적으로 할당하여 처리한다.

▶ 주기억장치의 용량보다 큰 프로그램을 실행하기 위해서 사용한다.

▶ 주기억장치의 이용률과 다중 프로그래밍(Multi-Tasking)의 효율을 높일 수 있다.

▶ 가상기억장치에 저장된 프로그램을 실행하려면 가상기억장치의

응용 SW 기초기술 활용 - SEC_06(가상기억장치 구현 기법/페이지 교체 알고리즘) 기출 및 출제 예상 문제

기출 및 출제 예상 문제(가상기억장치 구현 기법/페이지 교체 알고리즘)

9. 요구 페이징 기법 중 가장 오랫동안 사용되지 않았던 페이지를 먼저 교체하는 기법에 해당되는 것은?

① FIFO ② LFU

③ LRU ④ NUR

페이지 교체 알고리즘

① OPT(OPTimal replacement, 최적 교체)

▶ OPT는 앞으로 가장 오랫동안 사용하지 않을 페이지를 교체하는 기법이다.

▶ 벨레이디(Belady)가 제안한 것으로, 페이지 부재 횟수가 가장 적게 발생하는 가장 효율적인 알고리즘이다.

② FIFO(First In First Out)

▶ FIFO는 각 페이지가 주기억장치에 적재될 때마다 그때의 시간을 기억시켜 가정 먼저 들어와서 가장 오래 있었던 페이지를 교체하는 기법이다.

▶ 이해하기 쉽고, 프로그래밍 및 설계가 간단하다.

③ LRU(Least Recently Used)

11. 4개의 페이지를 수용할 수 있는 주기억장치가 있으며, 초기에는 모두 비어 있다고 가정한다. 다음의 순서로 페이지 참조가 발생할 때, LRU 페이지 교체 알고리즘을 사용할 경우 몇 번의 페이지 결함이 발생하는가?

페이지 참조 순서 : 1, 2, 3, 1, 2, 4, 1, 2, 5

① 5회 ② 6회

③ 7회 ④ 8회

참조 페이지	**1**	**2**	**3**	**1**	**2**	**4**	**1**	**2**	**5**
페이지 프레임	1	1	1	**1**	1	1	**1**	1	1
		2	2	2	**2**	2	2	**2**	2
			3	3	3	3	3	3	**5**
						4	4	4	4
부재 발생	O	O	O	X	X	O	X	X	O

LRU 기법을 사용하여 페이지를 참조하는 것은 위의 표와 같다.

LRU 기법은 최근에 가장 오랫동안 사용되지 않은 페이지를 교체하는 기법이기에 마지막 페이지 5를 참조할 때는 재 참조가 일어나지 않았던 페이지 3을 제거하고 페이지 5를 적재시킨다.

12. 페이지 교체 기법 중 매 페이지마다 두 개의 하드웨어 비트, 즉 참조 비트와 변형 비트가 필요한 기법은?

응용 SW 기초기술 활용 - SEC_06(가상기억장치 구현 기법/페이지 교체 알고리즘) 기출 및 출제 예상 문제

기출 및 출제 예상 문제(가상기억장치 구현 기법/페이지 교체 알고리즘)

13. NUR 기법은 호출 비트와 변형 비트를 가진다. 다음 중 가장 나중에 교체될 페이지는?

① 호출 비트 : 0, 변형 비트 : 0

② 호출 비트 : 0, 변형 비트 : 1

③ 호출 비트 : 1, 변형 비트 : 0

④ 호출 비트 : 1, 변형 비트 : 1

참조(호출) 비트와 변형 비트에 따른 교체 순서를 기억하도록 하자.

NUR 기법은 최근에 사용되지 않은 페이지를 교체하는 것이므로 참조되고 변경된 것을 찾으면 된다.

호출(참조) 비트는 변형 비트에 우선순위가 높다.

위의 나온 내용은 1 -> 2 -> 3 -> 4 의 순으로 페이지 교체가 된다.

14. 가상기억장치 구현 기법에 대한 설명으로 옳지 않은 것은?

① 가상기억장치 기법은 말 그대로 가상적인 것으로 현재 실무에서는 실현되는 방법이 아니다.

② 가상기억장치를 구현하는 일반적 방법에는 Paging과 Segmentation 기법이 있다.

15. 페이징 기법과 세그먼테이션 기법에 대한 설명으로 옳지 않은 것은?

① 페이징 기법에서는 주소 변환을 위한 페이지 맵 테이블이 필요하다.

② 페이지 크기로 일정하게 나누어진 주기억장치의 단위를 페이지 프레임이라고 한다.

③ 페이징 기법에서는 하나의 작업을 다양한 크기의 논리적인 단위로 나눈 후 주기억장치에 적재시켜 실행한다.

④ 세그먼테이션 기법을 이용하는 궁극적인 이유는 기억 공간을 절약하기 위해서이다.

페이징 기법에서는 하나의 작업을 동일한 크기로 나눈 후 주기억장치에 적재시켜 사용하는 기법이다. ③의 내용은 세그먼테이션 기법에 대한 설명이다.

16. 3개의 페이지 프레임을 갖는 시스템에서 페이지 참조 순서가 1, 2, 1, 0, 4, 1, 3 일 경우 FIFO 알고리즘에 의한 페이지 교체의 경우 프레임의 최종 상태는?

① 1, 2, 0　　② 2, 4, 3

③ 1, 4, 2　　④ 4, 1, 3

참조 페이지	1	2	1	0	4	1	3

응용 SW 기초기술 활용 - SEC_06(가상기억장치 구현 기법/페이지 교체 알고리즘) 기출 및 출제 예상 문제

기출 및 출제 예상 문제(가상기억장치 구현 기법/페이지 교체 알고리즘)

17. 페이지 교체 기법 중 LRU와 비슷한 알고리즘이며, 최근에 사용하지 않은 페이지를 교체하는 기법으로 시간 오버헤드를 줄이기 위해 각 페이지마다 참조 비트와 변형 비트를 두는 교체 기법은?

① FIFO ② LFU

③ NUR ④ OPT

NUR(Not Used Recently) : LRU와 비슷한 알고리즘으로, 최근에 사용하지 않은 페이지를 교체하는 기법이며, 참조 비트와 변형 비트의 값에 따라 교체될 페이지의 순서가 결정된다.

18. 캐시의 라인 교체 정책 가운데, 최근에 가장 적게 사용된 라인부터 교체하는 정책은?

① LRU ② FIFO

③ LFU ④ LIFO

캐시 메모리 : CPU와 주기억장치의 중간에 위치하며, CPU의 처리 속도를 높이기 위한 메모리이며 용량은 작지만 상당히 고가인 메모리이다.

LRU 기법 : 최근에 가장 오랫동안 사용하지 않은 페이지를 교체하는

19. 4개의 페이지를 수용할 수 있는 주기억장치가 있으며, 초기에는 모두 비어 있다고 가정한다. 다음의 순서로 페이지 참조가 발생할 때, LRU 페이지 교체 알고리즘을 사용할 경우 몇 번의 페이지 결함이 발생하는가?

페이지 참조 순서 : 1, 2, 3, 1, 2, 4, 1, 2, 5

① 5회 ② 6회

③ 7회 ④ 8회

4개의 페이지를 수용할 수 있는 주기억장치이므로 아래 표와 같이 4개의 페이지 프레임으로 표현할 수 있다.

참조 페이지	1	2	3	1	2	4	1	2	5
페이지 프레임	1	1	1	**1**	1	1	**1**	1	1
		2	2	2	**2**	2	2	**2**	2
			3	3	3	3	3	3	**5**
						4	4	4	4
부재 발생	O	O	O	X	X	O	X	X	O

4.응용 SW 기초기술 활용-SEC_07(가상기억장치 기타 관리 사항)

1) 페이지 크기

; 페이징 기법을 사용하면 프로그램을 페이지 단위로 나누게 되는데, 페이지의 크기에 따라 시스템에 미치는 영향이 다르다. 페이지 크기에 따른 특징은 다음과 같다.

페이지 크기가 작을 경우

- 페이지 단편화가 감소되고, 한 개의 페이지를 주기억장치로 이동하는 시간이 줄어든다.
- 불필요한 내용이 주기억장치에 적재될 확률이 적으므로 효율적인 워킹 셋을 유지할 수 있다.
- Locality에 더 일치할 수 있기 때문에 기억장치 효율이 높아진다.
- 페이지 정보를 갖는 페이지 맵 테이블의 크기가 커지고, 매핑 속도가 늦어진다.
- 디스크 접근 횟수가 많아져서 전체적인 입·출력 시간은 늘어난다.

페이지 크기가 클 경우

- 페이지 정보를 갖는 페이지 맵 테이블의 크기가 작아지고, 매핑 속도가 빨라진다.
- 디스크 접근 횟수가 줄어들어 전체적인 입·출력의 효율성이 증가된다.
- 페이지 단편화가 증가되고, 한 개의 페이지를 주기억장치로 이동하는 시간이 늘어난다.
- 프로세스(프로그램) 수행에 불필요한 내용까지도 주기억장치에 적재될 수 있다.

Locality(국부성, 지역성, 구역성, 국소성)는 프로세스가 실행되는 동안 주기억장치를 참조할 때 일부 페이지만 집중적으로 참조하는 성질이 있다는 이론이다.
워킹 셋은 프로세스가 일정 시간 동안 자주 참조하는 페이지들의 집합이다.

4.응용 SW 기초기술 활용-SEC_07(가상기억장치 기타 관리 사항)

2) Locality

; Locality(국부성, 지역성, 구역성, 국소성)는 프로세스가 실행되는 동안 주기억장치를 참조할 때 일부 페이지만 집중적으로 참조하는 성질이 있다는 이론이다.

- 스래싱을 방지하기 위한 워킹 셋 이론의 기반이 되었다.
- 프로세스가 집중적으로 사용하는 페이지를 알아내는 방법 중 하나로, 가상기억장치 관리의 이론적인 근거가 된다.
- 데닝(Denning) 교수에 의해 구역성의 개념이 증명되었으며 캐시 메모리 시스템의 이론적 근거이다.
- Locality의 종류에는 시간 구역성(Temporal Locality)과 공간 구역성(Spatial Locality)이 있다.

스래싱은 프로세스의 처리 시간보다 페이지 교체에 소요되는 시간이 더 많아지는 현상이다.

4.응용 SW 기초기술 활용-SEC_07(가상기억장치 기타 관리 사항)

2) Locality

시간 구역성(Temporal Locality)

- 시간 구역성은 프로세스가 실행되면서 하나의 페이지를 일정 시간 동안 집중적으로 액세스하는 현상이다.
- 한 번 참조한 페이지는 가까운 시간 내에 계속 참조할 가능성이 높음을 의미한다.
- 시간 구역성이 이루어지는 기억 장소 : Loop(반복, 순환), 스택(Stack), 부 프로그램(Sub Routine), Counting(1씩 증감), 집계(Totaling)에 사용되는 변수(기억장소)

공간 구역성(Spatial Locality)

- 공간 구역성은 프로세스 실행 시 일정 위치의 페이지를 집중적으로 액세스하는 현상이다.
- 어느 하나의 페이지를 참조하면 그 근처의 페이지를 계속 참조할 가능성이 높음을 의미한다.
- 공간 구역성이 이루어지는 기억장소 : 배열 순회(Array Traversal, 배열 순례), 순차적 코드의 실행, 프로그래머들이 관련된 변수(데이터를 저장할 기억장소)들을 서로 근처에 선언하여 할당되는 기억장소, 같은 영역에 있는 변수를 참조할 때 사용

4.응용 SW 기초기술 활용-SEC_07(가상기억장치 기타 관리 사항)

3) 워킹 셋(Working Set)

; 워킹 셋은 프로세스가 일정 시간 동안 자주 참조하는 페이지들의 집합이다.

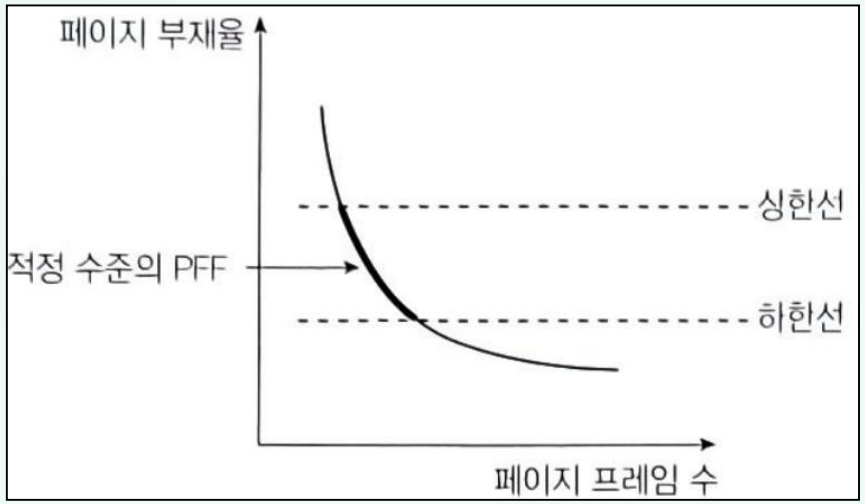
- 데닝(Denning)이 제안한 프로그램의 움직임에 대한 모델로, 프로그램의 Locality 특징을 이용한다.
- 자주 참조되는 워킹 셋을 주기억장치에 상주시킴으로써 페이지 부재 및 페이지 교체 현상이 줄어들어 프로세스의 기억장치 사용이 안정된다.
- 시간이 지남에 따라 자주 참조하는 페이지들의 집합이 변화하기 때문에 워킹 셋은 시간에 따라 변경된다.

4.응용 SW 기초기술 활용-SEC_07(가상기억장치 기타 관리 사항)

4) 페이지 부재 빈도 방식

; 페이지 부재(Page Fault)는 프로세스 실행 시 참조할 페이지가 주기억장치에 없는 현상이며, 페이지 부재 빈도(PFF; Page Fault Frequency)는 페이지 부재가 일어나는 횟수를 의미한다.

- 페이지 부재 빈도 방식은 페이지 부재율(Page Fault Rate)에 따라 주기억장치에 있는 페이지 프레임의 수를 늘리거나 줄여 페이지 부재율을 적정 수준으로 유지하는 방식이다.
- 운영체제는 프로세스 실행 초기에 임의의 페이지 프레임을 할당하고, 페이지 부재율을 지속적으로 감시하고 있다가 부재율이 상한선을 넘어가면 좀더 많은 페이지 프레임을 할당하고, 부재율이 하한선을 넘어가면 페이지 프레임을 회수하는 방식을 사용한다.



4.응용 SW 기초기술 활용-SEC_07(가상기억장치 기타 관리 사항)

5) 프리페이징(Prepaging)

; 프리페이징은 처음의 과도한 페이지 부재를 방지하기 위해 필요할 것 같은 모든 페이지를 한꺼번에 페이지 프레임에 적재하는 기법이다.

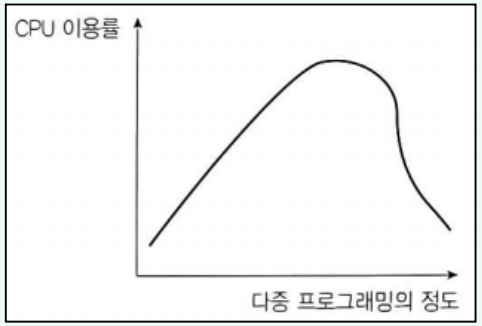
● 기억장치에 들어온 페이지들 중에서 사용되지 않는 페이지가 많을 수도 있다.

6) 스래싱(Thrashing)

; 스래싱은 프로세스의 처리 시간보다 페이지 교체에 소요되는 시간이 더 많아지는 현상이다.

● 다중 프로그래밍 시스템이나 가상기억장치를 사용하는 시스템에서 하나의 프로세스 수행 과정 중 자주 페이지 부재가 발생함으로써 나타나는 현상으로, 전체 시스템의 성능이 저하된다.

● 다중 프로그래밍의 정도가 높아짐에 따라 CPU의 이용률은 어느 특정 시점까지는 높아지지만, 다중 프로그래밍의 정도가 더욱 커지면 스래싱이 나타나고, CPU의 이용률은 급격히 감소하게 된다.



● **스래싱 현상 방지 방법**

- 다중 프로그래밍의 정도를 적정 수준으로 유지한다.
- 페이지 부재 빈도(Page Fault Frequency)를 조절하여 사용한다.
- 워킹 셋을 유지한다.
- 부족한 자원을 증설하고, 일부 프로세스를 중단시킨다.
- CPU 성능에 대한 자료의 지속적 관리 및 분석으로 임계치를 예상하여 운영한다.

응용 SW 기초기술 활용 - SEC_07(가상기억장치 기타 관리 사항) 기출 및 출제 예상 문제

기출 및 출제 예상 문제(가상기억장치 기타 관리 사항)

1. 페이징 기법에서 페이지 크기가 작아질수록 발생하는 현상이 아닌 것은?

① 기억장소 이용 효율이 증가한다.

② 입·출력 시간이 늘어난다.

③ 내부 단편화가 감소한다.

④ 페이지 맵 테이블의 크기가 감소한다.

페이지 기법을 사용하면 프로그램을 페이지 단위로 나누게 되는데, 페이지의 크기에 따라서 시스템에 미치는 영향이 다르다.

페이지 크기가 작을 경우

▶ 페이지 단편화가 감소되고, 한 개의 페이지를 주기억장치로 이동하는 시간은 줄어든다.

▶ 불필요한 내용이 주기억장치에 적재될 확률이 적으므로 효율적인 워킹 셋을 유지할 수 있다.

▶ 페이지 정보를 갖는 페이지 맵 테이블의 크기가 커지고, 매핑의 속도가 늦어진다.

▶ 디스크 접근 횟수가 많아져서 전체적인 입, 출력 시간은 늘어난다.

페이지 크기가 클 경우

3. 시간적 구역성(Temporal Locality)의 예가 아닌 것은?

① 루프 ② 서브루틴

③ 프로그램의 순차적 수행 ④ 스택

배열 순회, 순차적 코드, 프로그래머들이 관련된 변수들을 서로 근처 근처에 선언하여 할당되는 기억장소 등은 공간 구역성에 해당한다.

4. 프로세스들이 국부적인 부분만을 집중적으로 참조하는 구역성에는 시간 구역성과 공간 구역성이 있는데, 다음 중 공간 구역의 경우는?

① 순환(Looping)

② 배열 순례(Array Traversal)

③ 스택(Stack)

④ 집계(Totaling)에 사용되는 변수

응용 SW 기초기술 활용 - SEC_07(가상기억장치 기타 관리 사항) 기출 및 출제 예상 문제

기출 및 출제 예상 문제(가상기억장치 기타 관리 사항)

5. 운영체제의 가상기억장치 관리에서 프로세스가 일정 시간 동안 자주 참조하는 페이지들의 집합을 의미하는 것은?

① Locality ② Deadlock

③ Thrashing ④ Working Set

워킹 셋은 프로세스가 일정 시간 동안 자주 참조하는 페이지들의 집합이다.

▶ 데닝이 제안한 프로그램의 움직임에 대한 모델로, 프로그램의 Locality 특징을 이용한다.

▶ 자주 참조되는 워킹 셋을 주기억장치에 상주시킴으로써 페이지 부재 및 페이지 교체 현상이 줄어들어 프로세스의 기억장치 사용이 안정된다.

▶ 시간의 경과에 따라 자주 참조하는 페이지들의 집합이 변화하기 때문에 워킹 셋은 시간의 경과에 따라서 변경된다.

6. 페이지 오류율(Page Fault Ratio)과 스래싱(Thrashing)에 대한 설명으로 옳은 것은?

① 페이지 오류율이 크면 스래싱이 많이 발생한 것이다.

② 페이지 오류율과 스래싱은 전혀 관계가 없다.

7. Working set W(t, w)는 t-w 시간부터 까지 참조된 page들의 집합을 말한다. 그 시간에 참조된 페이지가 {2, 3, 5, 5, 6, 3, 7}라면 Working set은?

① {3, 5} ② {2, 6, 7}

③ {2, 3, 5, 6, 7} ④ {2, 7}

워킹 셋은 프로세스가 일정 시간 동안 참조하는 페이지들의 집합이므로 참조된 페이지에서 중복된 페이지를 제거하면 된다.

8. 다음 설명에 해당하는 내용은 무엇인가?

프로세스 처리 도중, 참조할 페이지가 주기억장치에 없어 프로세스 처리시간보다, 페이지 교체에 소요되는 시간이 더 많아지는 현상

① 스레드(Thread)

② 스래싱(Thrashing)

③ 페이지 부재(Page Fault)

④ 워킹 셋(Working Set)

프로세스 처리 시간 보다 페이지 교체 시간이 더 많아지는 현상을 스래싱 이라고 한다.

응용 SW 기초기술 활용 - SEC_07(가상기억장치 기타 관리 사항) 기출 및 출제 예상 문제

기출 및 출제 예상 문제(가상기억장치 기타 관리 사항)

9. 워킹 셋(Working Set)에 대한 설명으로 옳지 않은 것은?

① 프로세스가 실행하는 과정에서 시간이 지남에 따라 자주 참조하는 페이지들의 집합이 변화하기 때문에 워킹 셋은 시간에 따라 바뀌게 된다.

② 프로그램의 구역성(Locality) 특징을 이용한다.

③ 워킹 셋에 속한 페이지를 참조하면 프로세스의 기억장치 사용 은 안정상태가 된다.

④ 페이지 이동에 소요되는 시간과 프로세스 수행에 소요되는 시간의 차이를 의미한다.

10. 페이징 기법과 관련된 설명으로 옳지 않은 것은?

① 어떤 프로세스가 프로그램 실행에 사용하는 시간보다 페이지 적재/대체에 소비하는 시간이 더 큰 경우에 스래싱이 발생한다.

② 페이지 크기가 작을 경우 페이지 테이블의 공간이 많이 요구 된다.

③ 작업 셋(Working Set) 방식은 스래싱을 방지하는 방법 중의 하나이다.

④ 다중 프로그래밍의 정도가 높을수록 스래싱의 발생 빈도는 낮아진다.

11. 시스템을 설계할 때 최적의 페이지 크기에 관한 결정이 이루어져야만 한다. 페이지 크기에 관한 설명으로 옳지 않은 것은?

① 페이지 크기가 크면 페이지 테이블 공간은 증가한다.

② 입·출력 전송 시 큰 페이지가 더 효율적이다.

③ 페이지 크기가 클수록 디스크 접근 시간 부담이 감소된다.

④ 페이지 크기가 작아도 페이지 테이블의 단편화는 발생한다.

페이지 크기가 커지면 페이지의 수가 적어지므로 페이지 정보를 갖는 페이지 맵 테이블 공간은 줄어든다.

12. 실행 중인 프로세스는 일정 시간에 메모리의 일정 부분만을 집중적으로 참조한다는 개념을 의미하는 것은?

① Locality ② Monitor

③ Spooling ④ Fragmentation

모니터링(Monitoring)이란 어떤 대상을 감시, 관찰한다 라는 뜻으로 모니터링의 목적은 지속적인 감시, 감찰을 통해 대상의 상태나 가용성, 변화 등을 확인하고 대비하는 것이다. 즉, "어떤 대상의 상태나 상황을 지속적으로 감시, 관찰하여 예기치 못한 상황과 오류를 대비하고 극복한다"라고 할 수 있다.

스풀(Spool, 대기렬 관리)이란 컴퓨터 시스템에서 중앙처리장치와 입,

응용 SW 기초기술 활용 - SEC_07(가상기억장치 기타 관리 사항) 기출 및 출제 예상 문제

기출 및 출제 예상 문제(가상기억장치 기타 관리 사항)

13. 페이지 대체 문제에 관련된 사항 중 잘못된 것은?

① 스래싱(Thrashing) 현상이 일어나면 시스템의 처리율이 증가 한다.

② 시간 지역성이란 최근에 참조한 기억장소가 다시 참조될 가능성이 높다는 것이다.

③ 공간 지역성이란 참조된 기억장소에 대해 근처의 기억 장소가 다시 참조될 가능성이 높다는 것이다.

④ 어떤 프로세스가 빈번하게 참조하는 페이지들의 집합을 작업 셋이라 한다.

스래싱은 프로세스 처리 시간보다 페이지 교체 시간이 더 많아지는 현상이기에 스래싱이 발생하면 시스템의 처리율이 감소한다 라는 것을 기억하자.

14. 스래싱(Thrashing) 현상의 해결 조치로 틀린 것은?

① 부족한 자원을 증설한다.

② 일부 프로세스를 중단시킨다.

③ 성능 자료의 지속적 관리 및 분석으로 임계치를 예상하여 운영한다.

④ 다중 프로그래밍의 정도를 높여준다.

15. 페이징에서 페이지 크기에 관한 고려 사항으로 틀린 것은?

① 페이지 크기가 작을수록 페이지 테이블 크기가 커진다.

② 페이지 크기가 작을수록 좀더 알찬 워킹 셋을 유지할 수 있다.

③ 페이지 크기가 클수록 실제 프로그램 수행과 무관한 내용이 포함될 수 있다.

④ 페이지 크기가 클수록 전체적인 입·출력이 비효율적이다.

페이지의 크기가 클 경우 적은 수의 페이지가 존재하게 되어 작은 페이지 맵 테이블 공간을 필요로 하고, 참조되는 정보와 무관한 많은 양의 정보가 주기억장치에 남게 되며, 전체적인 입, 출력 효율은 증가된다.

16. 구역성(Locality)에 대한 설명으로 옳지 않은 것은?

① 실행 중인 프로세스가 일정 시간 동안에 참조하는 페이지의 집합을 의미한다.

② 시간 구역성과 공간 구역성이 있다.

③ 캐시 메모리 시스템의 이론적 근거이다.

④ Denning 교수에 의해 구역성의 개념이 증명되었다.

실행 중인 프로세스가 일정 시간 동안에 참조하는 페이지의 집합을 의미하는 것은 워킹 셋을 말한다.

Locality(국부성, 지역성, 구역성, 국소성)는 프로세스가 실행하는 동안

4.응용 SW 기초기술 활용-SEC_08(프로세스의 개요)

1) 프로세스(Process)의 정의

; 프로세스는 일반적으로 프로세서(처리기, CPU)에 의해 처리되는 사용자 프로그램, 시스템 프로그램, 즉 실행중인 프로그램을 의미하며, 작업(Job), 태스크(Task)라고도 한다.

프로세스는 다음과 같이 여러 형태로 정의할 수 있다.

- PCB를 가진 프로그램
- 실기억장치에 저장된 프로그램
- 프로세서가 할당되는 실체로서, 디스패치가 가능한 단위
- 프로시저가 활동 중인 것
- 비동기적 행위를 일으키는 주체
- 지정된 결과를 얻기 위한 일련의 계통적 동작
- 목적 또는 결과에 따라 발생되는 사건들의 과정
- 운영체제가 관리하는 실행 단위

PCB(PRINTED BOARD ASSEMBLY) : ASSY이라고도 불리며 인쇄회로기판에 필요한 부품을 탑재하여 납땜공정을 종료한 반제품.
IT용어에서의 **PCB(Process Control Block)**는 운영체제가 프로세스에 대한 중요한 정보를 저장해 놓는 곳으로, Task Control Block 또는 Job Control Block 이라고도 한다.
Dispatch(사전적 의미 : 보내다, 신속히 해치우다) '어딘가에 무엇을 보내는 행위' 또는 프로그램이 '어떤 메소드를 호출할 것인가를 결정하여 그것을 실행하는 과정'을 말한다.
프로시저 : 한 프로그램은 여러 개의 작은 프로그램으로 분할될 수 있는데, 이때 분할된 작은 프로그램을 의미하며, 부 프로그램(Sub-Routine)이라고도 한다.

4.응용 SW 기초기술 활용-SEC_08(프로세스의 개요)

2) PCB(Process Control Block, 프로세스 제어 블록)

; PCB는 운영체제가 프로세스에 대한 중요한 정보를 저장해 놓는 곳으로, Task Control Block 또는 Job Control Block이라고도 한다.

- 각 프로세스가 생성될 때마다 고유의 PCB가 생성되고, 프로세스가 완료되면 PCB는 제거된다.
- PCB에 저장되어 있는 정보는 다음과 같다.

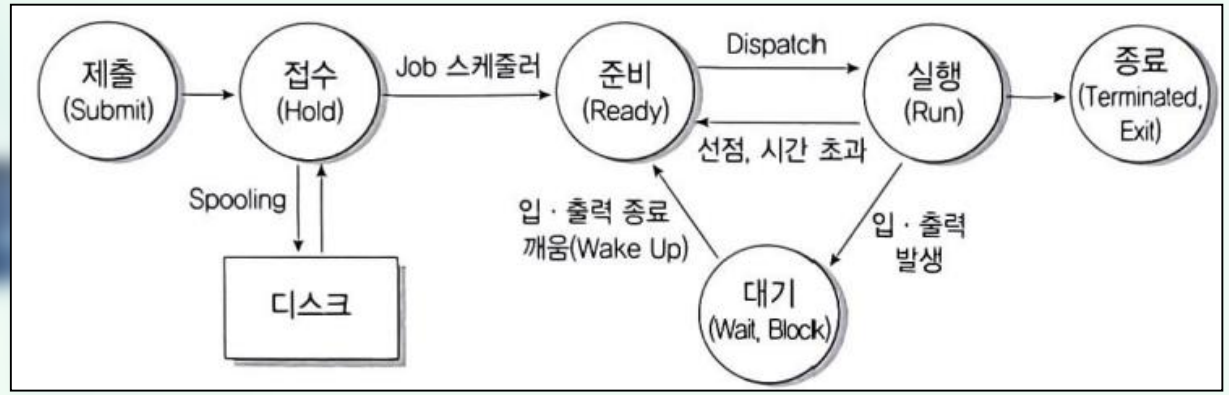
저장 정보	설명
프로세스의 현재 상태	준비, 대기, 실행 등의 프로세스 상태
포인터	•부모 프로세스에 대한 포인터 : 부모 프로세스의 주소 기억 •자식 프로세스에 대한 포인터 : 자식 프로세스의 주소 기억 •프로세스가 위치한 메모리에 대한 포인터 : 현재 프로세스가 위치한 주소 기억 •할당된 자원에 대한 포인터 : 프로세스에 할당된 각 자원에 대한 주소 기억
프로세스 고유 식별자	프로세스를 구분할 수 있는 고유의 번호
스케줄링 및 프로세스의 우선순위	스케줄링 정보 및 프로세스가 실행될 우선순위
CPU 레지스터 정보	Accumulator(누산기), 인덱스 레지스터, 범용 레지스터, 프로그램 카운터(PC)등에 대한 정보
주기억장치 관리 정보	기준 레지스터(Base Register), 페이지 테이블(Page Table)에 대한 정보
입·출력 상태 정보	입·출력장치, 개방된 파일 목록
계정 정보	CPU 사용 시간, 실제 사용 시간 한정된 시간

부모 프로세스/ 자식 프로세스 : 하나의 프로세스로 다른 프로세스를 생성할 수 있는데, 이때 생성되는 프로세스를 자식 프로세스 라고 하며, 기존에 있는 프로세스를 부모 프로세스라고 한다.
기준 레지스터 : 주기억장치가 분할된 영역으로 나뉘어 관리될 때, 프로그램이 한 영역에서 다른 영역의로 옮겨지더라도 명령의 주소 부분을 바꾸지 않고 정상적으로 수행될 수 있도록 하기 위한 레지스터이다.

4.응용 SW 기초기술 활용-SEC_08(프로세스의 개요)

3) 프로세스 상태 전이

; 프로세스 상태 전이는 프로세스가 시스템 내에 존재하는 동안 프로세스의 상태가 변하는 것을 의미하며, 프로세스의 상태를 다음과 같이 상태 전이도로 표시할 수 있다.



- 프로세스의 상태는 제출, 접수, 준비, 실행, 대기 상태로 나눌 수 있으며, 이 중 주요 세 가지 상태는 준비, 실행, 대기 상태이다.
- **제출(Submit)** : 작업을 처리하기 위해 사용자가 작업을 시스템에 제출한 상태이다.
- **접수(Hold)** : 제출된 작업이 스풀 공간인 디스크의 할당 위치에 저장된 상태이다.
- **준비(Ready)**
  - 프로세스가 프로세서를 할당 받기 위해 기다리고 있는 상태이다.
  - 프로세스는 준비상태 큐에서 실행을 준비하고 있다.
  - 접수상태에서 준비 상태로의 전이는 Job 스케줄러에 의해 수행된다.

4.응용 SW 기초기술 활용-SEC_08(프로세스의 개요)

3) 프로세스 상태 전이

- **실행(Run)**
  - 준비상태 큐에 있는 프로세스가 프로세서를 할당 받아 실행되는 상태이다.
  - 프로세스 수행이 완료되기 전에 프로세스에게 주어진 프로세서 할당 시간이 종료(Timer Run Out)되면 프로세스는 준비 상태로 전이된다.
  - 실행중인 프로세스에 입·출력(I/O) 처리가 필요하면 실행중인 프로세스는 대기 상태로 전이된다.
  - 준비 상태에서 실행 상태로의 전이는 CPU(프로세서) 스케줄러에 의해 수행된다.
- **대기(Wait), 보류, 블록(Block)** : 프로세스에 입·출력 처리가 필요하면 현재 실행 중인 프로세스가 중단되고, 입·출력 처리가 완료될 때까지 대기하고 있는 상태이다.
- **종료(Terminated, Exit)** : 프로세스의 실행이 끝나고 프로세스 할당이 해제된 상태이다.

준비상태 큐 : 여러 프로세스가 프로세서를 할당 받기 위해 기다리는 장소이다.

4.응용 SW 기초기술 활용-SEC_08(프로세스의 개요)

4) 프로세스 상태 전이 관련 용어

- **Dispatch** : 준비 상태에서 대기하고 있는 프로세스 중 하나가 프로세서를 할당 받아 실행 상태로 전이되는 과정이다.
- **Wake Up** : 입·출력 작업이 완료되어 프로세스가 대기 상태에서 준비 상태로 전이되는 과정이다.
- **Spooling** : 입·출력장치의 공유 및 상대적으로 느린 입·출력장치의 처리 속도를 보완하고 다중 프로그래밍 시스템의 성능을 향상시키기 위해 입·출력할 데이터를 직접 입·출력장치에 보내지 않고 나중에 한꺼번에 입·출력하기 위해 디스크에 저장하는 과정이다.
- **교통량 제어기(Traffic Controller)** : 프로세스의 상태에 대한 조사와 통보를 담당한다.

4.응용 SW 기초기술 활용-SEC_08(프로세스의 개요)

5) 스레드(Thread)

; 스레드는 프로세스 내에서의 작업 단위로서 시스템의 여러 자원을 할당 받아 실행하는 프로그램의 단위이다.

- 하나의 프로세스에 하나의 스레드가 존재하는 경우에는 단일 스레드, 하나 이상의 스레드가 존재하는 경우에는 다중(멀티, Multi) 스레드라고 한다.
- 프로세스의 일부 특성을 갖고 있기 때문에 경량(Light Weight) 프로세스라고도 한다.
- 스레드 기반 시스템에서 스레드는 독립적인 스케줄링의 최소 단위로서 프로세스의 역할을 담당한다.
- 동일 프로세스 환경에서 서로 독립적인 다중 수행이 가능하다.
- 스레드의 분류

사용자 수준의 스레드	•사용자가 만든 라이브러리를 사용하여 스레드를 운용한다. •커널 모드로의 전환이 없어 오버헤드가 줄어든다. •속도는 빠르지만 구현이 어렵다.
커널 수준의 스레드	•운영체제의 커널에 의해 스레드를 운용한다. •한 프로세스가 운영체제를 호출할 때 전체 프로세스가 대기하지 않으므로 시스템의 성능을 높일 수 있다. •여러 스레드가 커널에 동시에 접근할 수 있다. •스레드의 독립적인 스케줄링이 가능하다. •구현이 쉽지만 속도가 느리다.

4.응용 SW 기초기술 활용-SEC_08(프로세스의 개요)

5) 스레드(Thread)

● 스레드 사용의 장점

- 하나의 프로세스를 여러 개의 스레드로 생성하여 병행성을 증진시킬 수 있다.
- 하드웨어, 운영체제의 성능과 응용 프로그램의 처리율을 향상시킬 수 있다.
- 응용 프로그램의 응답 시간(Response Time)을 단축시킬 수 있다.
- 실행 환경을 공유시켜 기억장소의 낭비가 줄어든다.
- 프로세스들 간의 통신이 향상된다.
- 스레드는 공통적으로 접근 가능한 기억장치를 통해 효율적으로 통신한다.

응용 SW 기초기술 활용 - SEC_08(프로세스의 개요) 기출 및 출제 예상 문제

기출 및 출제 예상 문제(프로세스의 개요)

1. 다음 중 프로세스에 대한 설명 중 틀린 것은?

① 프로세서가 할당되는 실체로, 디스패치가 가능한 단위이다.

② 프로세스는 비동기적 행위를 일으키는 주체이다.

③ 프로세스는 스레드 내의 작업 단위를 의미하며, 경량 스레드라고도 불린다.

④ PCB를 가지며 PCB에는 프로세스의 현재 상태, 고유 식별자를 가지고 있다.

프로세스의 정의

프로세스는 일반적으로 프로세서(처리기, CPU)에 의해 처리되는 사용자 프로그램, 시스템 프로그램, 즉 실행중인 프로그램을 의미하며, 작업(Job), 태스크(Task)라고도 한다.

프로세스는 다음과 같이 여러 형태로 정의할 수 있다.

▶ PCB를 가진 프로그램

PCB(Process Control Block)는 운영체제가 프로세스에 대한 중요한 정보를 저장해 놓는 곳이다.

▶ 실기억장치에 저장된 프로그램

▶ 프로세서가 할당되는 실체로서, 디스패치가 가능한 단위

3. 프로세스 제어 블록(Process Control Block)에 대한 설명으로 옳지 않은 것은?

① 프로세스에 할당된 자원에 대한 정보를 갖고 있다.

② 프로세스의 우선순위에 대한 정보를 갖고 있다.

③ 부모 프로세스와 자식 프로세스는 PCB를 공유한다.

④ 프로세스의 현 상태를 알 수 있다.

PCB는 운영체제가 프로세스에 대한 중요한 정보를 저장해 놓는 곳으로, Task Control Block 또는 Job Control Block이라고 한다.

▶ 각 프로세스가 생성될 때마다 고유의 PCB가 생성되고, 프로세스가 완료되면 PCB도 함께 제거된다.

▶ PCB에 저장되어 있는 정보는 다음과 같다.

1. 프로세스의 현재 상태 : 준비, 대기, 실행 등의 프로세스의 상태
2. 포인터
   - 부모 프로세스에 대한 포인터 : 부모 프로세스의 주소 기억
   - 자식 프로세스에 대한 포인터 : 자식 프로세스의 주소 기억
   - 프로세스가 위치한 메모리에 대한 포인터 : 현재 프로세스가 위치한 주소 기억
   - 할당된 자원에 포인터 : 프로세스에 할당된 각 자원에 대한

응용 SW 기초기술 활용 - SEC_08(프로세스의 개요) 기출 및 출제 예상 문제

기출 및 출제 예상 문제(프로세스의 개요)

5. 프로세스 상태의 종류가 아닌 것은?

① Ready ② Running

③ Request ④ Exit

프로세스 상태는 제출(Submit), 접수(Hold), 준비(Ready), 실행(Running), 종료(Exit, Terminated)로 구분된다.

▶ **제출(Submit)** : 작업을 처리하기 위해 사용자가 작업을 시스템에 제출한 상태이다.

▶ **접수(Hold)** : 제출된 작업이 스풀 공간인 디스크의 할당 위치에 저장된 상태이다.

▶ **준비(Ready)**

- 프로세스가 프로세서를 할당 받기 위해서 기다리고 있는 상태이다.
- 프로세스는 준비상태 큐에서 실행을 준비하고 있다.
- 접수 상태에서 준비 상태로의 전이는 Job 스케쥴러에 의해 수행된다.

▶ **실행(Run)**

- 준비상태 큐에 있는 프로세스가 프로세서를 할당 받아서

7. 스레드(Thread)에 대한 설명으로 옳지 않은 것은?

① 한 개의 프로세스는 여러 개의 스레드를 가질 수 없다.

② 커널 스레드의 경우 운영체제에 의해 스레드를 운용한다.

③ 사용자 스레드의 경우 사용자가 만든 라이브러리를 사용하여 스레드를 운용한다.

④ 스레드를 사용함으로써 하드웨어, 운영체제의 성능과 응용 프로그램 의 처리율을 향상시킬 수 있다.

하나의 프로세스에 하나 이상의 스레드가 존재하는 경우에는 다중(멀티) 스레드라고 한다.

스레드(Thread)

스레드는 프로세스 내에서의 작업 단위로서 시스템의 여러 자원을 할당 받아 실행하는 프로그램의 단위이다.

▶ 하나의 프로세스에 하나의 스레드가 존재하는 경우에는 단일(싱글) 스레드, 하나 이상의 스레드가 존재하는 경우에는 다중(멀티) 스레드 라고 한다.

▶ 프로세스의 일부 특정을 갖고 있기 때문에 경량(Light Weight) 프로세스라고도 한다.

▶ 스레드 기반 시스템에서는 스레드는 독립적인 스케쥴링의 최소 단위

응용 SW 기초기술 활용 - SEC_08(프로세스의 개요) 기출 및 출제 예상 문제

기출 및 출제 예상 문제(프로세스의 개요)

9. 프로세스(Process)에 대한 설명으로 옳지 않는 것은?

① 트랩 오류, 프로그램 요구, 입·출력 인터럽트에 대해 조치를 취한다.

② 비동기적 행위를 일으키는 주체로 정의할 수 있다.

③ 실행중인 프로그램을 말한다.

④ 프로세스는 각종 자원을 요구한다.

트랩 오류, 프로그램 요구, 입·출력 인터럽트에 대해 조치를 취하는 것은 운영체제의 역할이다.

trap : 세그먼트의 크기보다 크다면 메모리 오류를 출력하고 해당 프로세스를 강제 종료한다. 이 때 발생하는 오류를 trap이라고 한다.

인터럽트(Interrupt)란 프로세스가 실행 도중 예기치 않은 상황이 발생할 때 발생한 상황을 처리한 후 실행 중인 작업으로 복귀하는 것을 의미한다.

주로 입출력 장치의 signal, data가 발생할 때까지 원래의 작업을 수행하다가 해당 기능을 처리하는 것이다.(외부 인터럽트)

인터럽트의 종류는 외부 인터럽트, 내부 인터럽트, 소프트웨어 인터럽트로 나뉜다.

비동기적 인터럽트 / 하드웨어 인터럽트

4.응용 SW 기초기술 활용-SEC_09(스케줄링)

1) 스케줄링(Scheduling)의 개요

; 스케줄링은 프로세스가 생성되어 실행될 때 필요한 시스템의 여러 자원을 해당 프로세스에게 할당하는 작업을 의미한다.

- 프로세스가 생성되어 완료될 때까지 프로세스는 여러 종류의 스케줄링 과정을 거치게 된다.
- 스케줄링의 종류에는 장기 스케줄링, 중기 스케줄링, 단기 스케줄링이 있다.

장기 스케줄링	•어떤 프로세스가 시스템의 자원을 차지할 수 있도록 할 것인가를 결정하여 준비상태 큐로 보내는 작업을 의미한다. •작업 스케줄링(Job Scheduling), 상위 스케줄링이라고도 하며, 작업 스케줄러에 의해 수행된다.
중기 스케줄링	•어떤 프로세스들이 CPU를 할당 받을 것인지 결정하는 작업을 의미한다. •CPU를 할당 받으려는 프로세스가 많을 경우 프로세스를 일시 보류시킨 후 활성화해서 일시적으로 부하를 조절한다.
단기 스케줄링	•프로세스가 실행되기 위해 CPU를 할당받는 시기와 특정 프로세스를 지정하는 작업을 의미한다. •프로세서 스케줄링(Processor Scheduling), 하위 스케줄링이라고도 한다. •프로세서 스케줄링 및 문맥 교환은 프로세서 스케줄러에 의해 수행된다

문맥 교환(Context Switching) : 하나의 프로세스에서 다른 프로세스로 CPU가 할당되는 과정에서 발생되는 것으로 새로운 프로세스에 CPU를 할당하기 위해 현재 CPU가 할당된 프로세스의 상태 정보를 저장하고 새로운 프로세스의 상태 정보를 설정한 후 CPU를 할당하여 실행되도록 하는 작업을 의미한다.

4.응용 SW 기초기술 활용-SEC_09(스케줄링)

2) 프로세스 스케줄링의 기법

비선점(Non-Preemptive) 스케줄링

- 이미 할당된 CPU를 다른 프로세스가 강제로 빼앗아 사용할 수 없는 스케줄링 기법이다.
- 프로세스가 CPU를 할당 받으면 해당 프로세스가 완료될 때까지 CPU를 사용한다.
- 모든 프로세스에 대한 요구를 공정하게 처리할 수 있다.
- 프로세스 응답 시간의 예측이 용이하며, 일괄 처리 방식에 적합하다.
- 중요한 작업(짧은 작업)이 중요하지 않은 작업(긴 작업)을 기다리는 경우가 발생할 수 있다.
- 비선점 스케줄링의 종류에는 FCFS, SJF, 우선순위(Priority), HRN, 기한부 등의 알고리즘이 있다.

비선점 : Non-Preemption(비우선 매수)이나 Non-Preemptive(비 선점)로 표현할 수 있다.
일괄 처리(batch processing) : 최종 사용자의 개입 없이 또는 (자원이 허가한다면) 실행을 스케줄링 할 수 있는 작업 (job)의 실행을 의미한다. 컴퓨터 프로그램 흐름에 따라 순차적으로 자료를 처리하는 방식이다. 초기의 일괄 처리 방식 은 사용자와 상호작용하는 것이 불가능했지만, 운영 체제가 발전함에 따라 프로그램 입출력을 통해 상호작용하는 것이 가능해졌다. 일괄 처리는 1950년대 전자 컴퓨팅 초기 시절 이후 메인 프레임 컴퓨터와 함께하고 있다.
FCFS(First-Come-First-Served) : FIFO (First-In-First-Out)와 같다. 말 그대로, 먼저 준비된 프로세서를 먼저 처리하는 방식이다.
SJF(Shortest Job First) : 프로세스의 실행 시간을 이용하여 가장 짧은 시간을 갖는 프로세스가 먼저 자원을 할당받는 방식이다.
우선순위(Priority) : 각 프로세스 별로 우선순위가 주어지고, 우선순위에 따라 CPU 할당되며, 우선순위가 같을 경우 FCFS 적용한다. 설정, 자원 상황 등에 따른 우선순위를 선정해 주요/긴급 프로세스에 대한 우선처리 가능하다.
HRN(Highest Response ratio Next) : 짧은 작업에 유리한 SJF의 단점을 개선 한 기법, 각 작업의 우선순위로 서비스 해주는 스케줄링이다.
기한부 스케줄링(deadline scheduling) : 작업들이 명시된 기간이나 기한 내에 완료되도록 계획한다.

4.응용 SW 기초기술 활용-SEC_09(스케줄링)

2) 프로세스 스케줄링의 기법

선점(Preemptive) 스케줄링

- 하나의 프로세스가 CPU를 할당 받아 실행하고 있을 때 우선순위가 높은 다른 프로세스가 CPU를 강제로 빼앗아 사용할 수 있는 스케줄링 기법이다.
- 우선순위가 높은 프로세스를 빠르게 처리할 수 있다.
- 주로 빠른 응답 시간을 요구하는 대화식 시분할 시스템에 사용된다.
- 많은 오버헤드(Overhead)를 초래한다.
- 선점이 가능하도록 일정 시간 배당에 대한 인터럽트용 타이머 클록(Clock)이 필요하다.
- 선점 스케줄링의 종류에는 Round Robin, SRT, 선점 우선순위, 다단계 큐, 다단계 피드백 큐 등의 알고리즘이 있다.

인터럽트용 타이머 클록 : 하나의 시스템 내에서 동작하는 장치들을 감시하기 위해 주기적인 신호를 발생하는 것으로 하나의 프로세스가 자원을 독점하지 못하도록 방지하기 위해 사용된다.
시분할 시스템(time-sharing) : 컴퓨터를 대화식으로 사용하려는 시도에서 탄생하였다. 시분할 운영 체제는 CPU스케줄링 과 다중 프로그래밍 을 이용해서 각 사용자들에게 컴퓨터 자원을 시간적으로 분할하여 사용할 수 있게 해 준다.
라운드 로빈(Round Robin) : 프로세스마다 같은 크기의 CPU 시간을 할당(시간 할당량/Time Slice), 보통 10 ~ 100ms 정도이다.프로세스가 할당된 시간 내에 처리완료 못하면 준비 큐 리스트의 가장 뒤로 보내지고, CPU는 대기중인 다음 프로세스로 넘어가며, 균등한 CPU 점유 시간을 보장하며, 시분할 시스템에 사용된다.
SRT(Shortest Remaining Time First) : 가장 짧은 시간이 소요되는 프로세스를 먼저 수행하며, 남은 시간이 더 짧다고 판단되는 프로세스가 준비 큐에 생기면 언제라도 프로세스가 점령된다.
다단계 큐(Multi Level Queue) : 작업들을 여러 종류 그룹으로 분할, 여러 개의 큐를 이용하여 상위 단계 작업이 선점한다. 각 큐는 자신 만의 독자적인 스케줄링을 가진다.
다단계 피드백 큐 (Multi Level Feedback Queue) : 입출력 위주와 CPU 위주인 프로세스의 특성에 따라 큐마다 서로 다른 CPU 시간 할당량 부여하며, FCFS와 라운드 로빈 기법을 혼합한 알고리즘이다. 유연성 뛰어나고 turnaround 시간과 response time에 최적화가 되어있다.

응용 SW 기초기술 활용 - SEC_09(스케줄링) 기출 및 출제 예상 문제

기출 및 출제 예상 문제(스케줄링)

1. 선점(Preemptive) 기법의 스케줄링에 해당하는 것은?

① FIFO ② SJF

③ HRN ④ RR

선점(Preemptive) 스케줄링

▶ 하나의 프로세스가 CPU를 할당 받아 실행하고 있을 때 우선순위가 높은 다른 프로세스가 CPU를 강제로 빼앗아 사용할 수 있는 스케줄링 기법이다.

▶ 우선 순위가 높은 프로세스를 빠르게 처리할 수 있다.

▶ 주로 빠른 응답 시간을 요구하는 대화식 시분할 시스템에 사용된다.

▶ 많은 오버헤드(Overhead)를 초래한다.

▶ 선점이 가능하도록 일정 시간 배당에 대한 인터럽트 타이머 클록(Clock)이 필요하다.

▶ 선점 스케쥴링의 알고리즘의 종류에는 Round Robin, SRT, 선점 우선순위, 다단계 큐, 다단계 피드백 큐 등의 알고리즘이 있다.

인터럽트 타이머 클록(Clock) : 하나의 시스템 내에서 동작하는 장치들을 감시하기 위해 주기적인 신호를 발생하는 것으로 하나의

3. 선점(Preemptive) 스케줄링 방식에 대한 설명으로 옳지 않은 것은?

① 대화식 시분할 시스템에 적합하다.

② 긴급하고 높은 우선순위의 프로세스들이 빠르게 처리될 수 있다.

③ 일단 CPU를 할당 받으면 다른 프로세스가 CPU를 강제적으로 빼앗을 수 없는 방식이다.

④ 선점을 위한 시간 배당에 대한 인터럽트용 타이머 클록(Clock)이 필요하다.

선점 스케줄링은 CPU 를 할당 받았을 때 다른 프로세스가 CPU를 차지할 수 있다.

4. 비선점(Non-preemptive) 스케줄링 방식에 해당하는 것으로만 짝지어진 것은?

① FCFS(First Come First Service), SJF(Shortest Job First)

② RR(Round-Robin), SRT(Shortest Remaining Time)

③ SRT(Shortest Remaining Time), SJF(Shortest Job First)

④ MQ(Multi-level Queue), FCFS(First Come First Service)

비선점 스케쥴링의 종류에는 FCFS, SJF, 우선순위, HRN, 기한부 알고리즘이 있다.

응용 SW 기초기술 활용 - SEC_09(스케줄링) 기출 및 출제 예상 문제

기출 및 출제 예상 문제(스케줄링)

5. 스케줄링의 목적으로 가장 거리가 먼 것은?

① 모든 작업들에 대해 공평성을 유지하기 위하여

② 단위 시간당 처리량을 최대화하기 위하여

③ 응답 시간을 빠르게 하기 위하여

④ 운영체제의 오버헤드를 최대화하기 위하여

스케쥴링은 프로세스가 생성되어 실행될 때 필요한 시스템의 여러 자원을 해당 프로세스에게 할당하는 작업을 의미하는 것으로, 운영체제의 오버헤드를 최소화하기 위해 사용한다.

6. 선점 스케줄링과 비선점 스케줄링에 대한 비교 설명 중 옳은 것은?

① 선점 스케줄링은 이미 할당된 CPU를 다른 프로세스가 강제로 빼앗아 사용할 수 없다.

② 선점 스케줄링은 상대적으로 과부하가 적다.

③ 비선점 스케줄링은 시분할 시스템에 유용하다.

④ 비선점 스케줄링은 응답시간의 예측이 용이하다.

① 선점 스케줄링은 이미 할당된 CPU 를 다른 프로세스가 강제로 빼앗아 사용할 수 있다.

7. 프로세스가 시스템의 자원을 차지할 수 있도록 할 것인가를 결정하여 준비상태 큐로 보내는 작업을 무엇이라고 하는가?

① 선점 스케줄링② 장기 스케줄링

③ 중기 스케줄링④ 단기 스케쥴링

장기 스케쥴링

▶ 어떤 프로세스가 시스템의 자원을 차지할 수 있도록 할 것인가를 결정하여, 준비상태 큐로 보내는 작업을 의미한다.

▶ 작업 스케줄링(Job Scheduling), 상위 스케쥴링이라고도 하며, 작업 스케줄러에 의해 수행된다.

중기 스케줄링

▶ 어떤 프로세스들이 CPU 를 할당 받을 것인지 결정하는 작업을 의미한다.

▶ CPU 를 할당 받으려는 프로세스가 많을 경우 프로세스를 일시 보류시킨 후 활성화 해서 일시적으로 부하를 조절한다.

단기 스케줄링

▶ 프로세스가 실행되기 위해 CPU 를 할당 받는 시기와 특정 프로세스 를 지정하는 작업을 의미한다.

▶ 프로세서 스케줄링(Processor Scheduling), 하위 스케쥴링이라고도

4.응용 SW 기초기술 활용-SEC_10(주요 스케줄링 알고리즘)

1) FCFS(First Come First Service, 선입선출) = FIFO(First In First Out)

; FCFS는 준비상태 큐(대기 큐, 준비 완료 리스트, 작업준비 큐, 스케줄링 큐)에 도착한 순서에 따라 차례로 CPU를 할당하는 기법으로, 가장 간단한 알고리즘이다.

- 먼저 도착한 것이 먼저 처리되어 공평성은 유지되지만 짧은 작업이 긴 작업을, 중요한 작업이 중요하지 않은 작업을 기다리게 된다.

4.응용 SW 기초기술 활용-SEC_10(주요 스케줄링 알고리즘)

1) FCFS(First Come First Service, 선입선출) = FIFO(First In First Out)

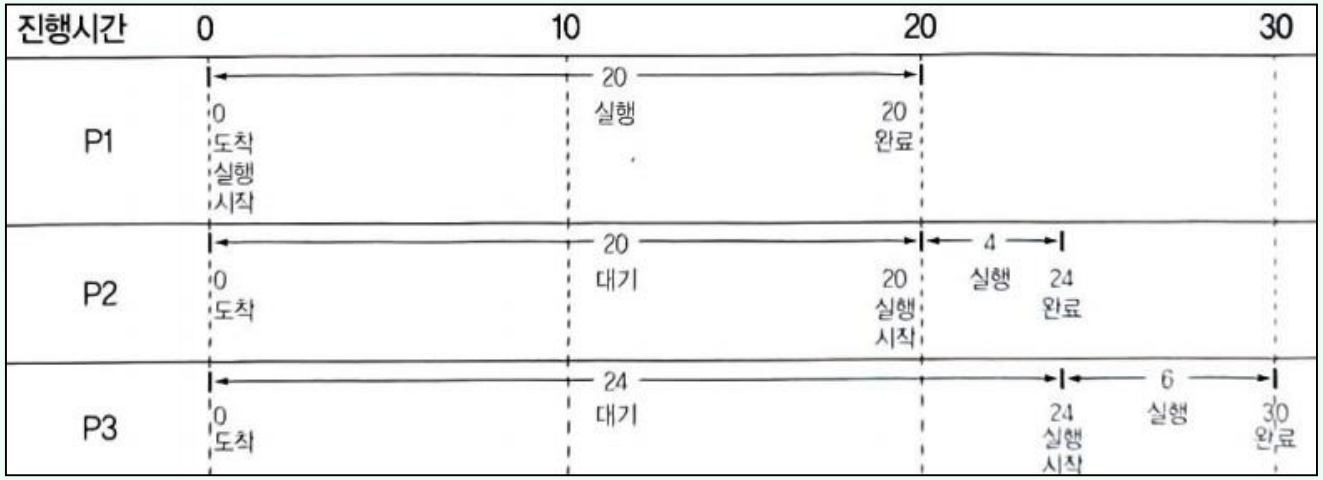
예제) 다음과 같은 프로세스들이 차례로 준비상태 큐에 들어왔다고 가정할 때, FCFS 기법을 이용하여 평균 실행 시간, 평균 대기 시간, 평균 반환 시간을 구하시오(제출시간은 없으며 시간의 단위는 초임).

프로세스 번호	P1	P2	P3
실행 시간	20	4	6

① 실행 시간을 이용하여 다음과 같이 각 프로세스의 대기 시간과 반환 시간을 구한다.

- 대기 시간 : 프로세스가 대기한 시간으로, 바로 앞 프로세스까지의 진행 시간으로 계산
- 반환 시간 : 프로세스의 대기 시간과 실행 시간의 합

② 실행 시간, 대기 시간 반환 시간의 평균은 '각 프로세스 시간의 합/프로세스의 개수'를 이용



- 평균 실행 시간 : (20 + 4 + 6) / 3 = 10
- 평균 대기 기간 : (0 + 20 + 24) / 3 = 14.6
- 평균 반환 시간 : (20 + 24 + 30) / 3 =24.6

4.응용 SW 기초기술 활용-SEC_10(주요 스케줄링 알고리즘)

2) SJF(Shortest Job First, 단기 작업 우선)

; SJF는 준비상태 큐에서 기다리고 있는 프로세스들 중에서 실행 시간이 가장 짧은 프로세스에게 먼저 CPU를 할당하는 기법이다.

- 가장 적은 평균 대기 시간을 제공하는 최적 알고리즘이다.
- 실행 시간이 긴 프로세스는 실행 시간이 짧은 프로세스에게 할당 순위가 밀려 무한 연기 상태가 발생될 수 있다.

4.응용 SW 기초기술 활용-SEC_10(주요 스케줄링 알고리즘)

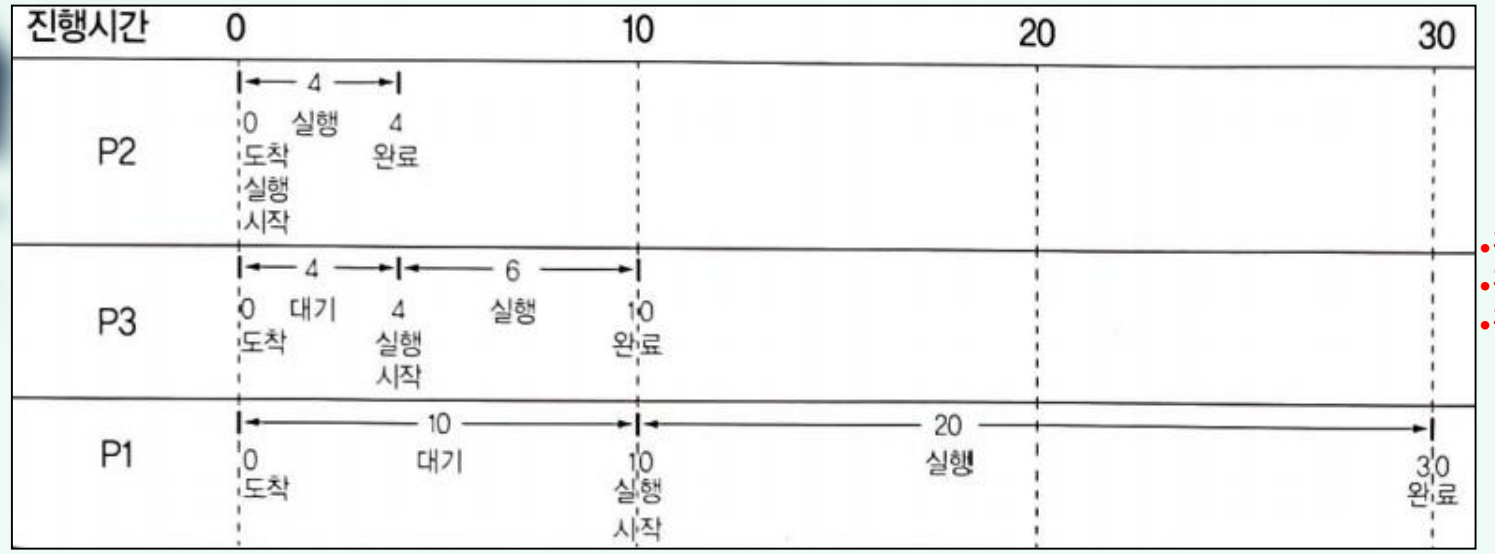
2) SJF(Shortest Job First, 단기 작업 우선)

예제1) 다음과 같은 프로세스들이 차례로 준비상태 큐에 들어왔다고 가정할 때, SJF기법을 이용하여 평균 실행 시간, 평균 대기 시간, 평균 반환 시간을 구하시오(제출시간이 없을 경우).

프로세스 번호	P1	P2	P3
실행 시간	20	4	6

① 아래와 같이 실행 시간이 짧은 프로세스를 먼저 처리하도록 이동시킨 후 각 프로세스의 대기 시간과 반환 시간을 구한다.

② 실행 시간, 대기 시간, 반환 시간, 각 시간의 평균은 FCFS의 예제와 동일한 방법으로 구한다.



- 평균 실행 시간 : (4 + 6 + 20) / 3 = 10
- 평균 대기 기간 : (0 + 4 + 10) / 3 = 4.6
- 평균 반환 시간 : (4 + 10 + 30) / 3 = 14.6

4.응용 SW 기초기술 활용-SEC_10(주요 스케줄링 알고리즘)

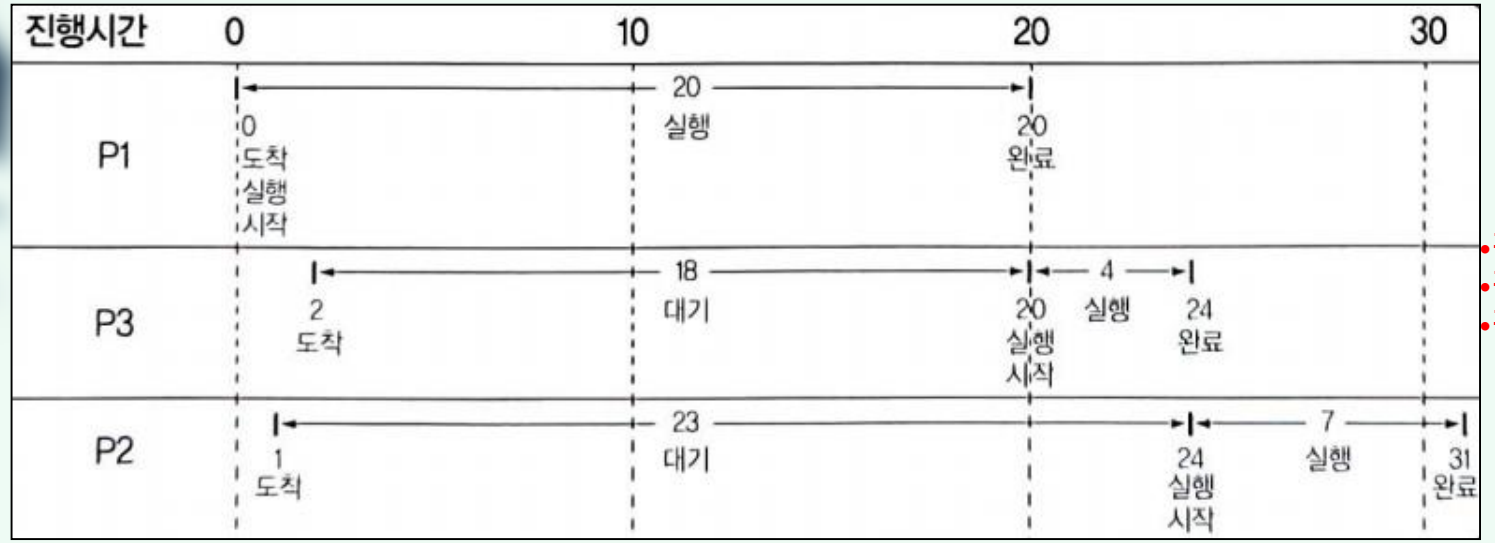
2) SJF(Shortest Job First, 단기 작업 우선)

예제2) 다음과 같은 프로세스들이 차례로 준비상태 큐에 들어왔다고 가정할 때, SJF기법을 이용하여 평균 실행 시간, 평균 대기 시간, 평균 반환 시간을 구하시오(제출시간이 있을 경우).

프로세스 번호	P1	P2	P3
실행 시간	20	7	4
제출 시간	0	1	2

① 가장 먼저 도착한 P1을 실행한 후 요구된 실행 시간이 적은 P3, P2 순으로 수행한다.

② 대기 시간은 현재 프로세스가 수행되기 전까지의 진행 시간에서 제출 시간을 차감하고, 반환 시간은 실행 시간과 대기 시간의 합으로 구한다.



- 평균 실행 시간 : (20 + 4 + 7) / 3 = 10.3
- 평균 대기 기간 : (0 + 18 + 23) / 3 = 13.6
- 평균 반환 시간 : (20 + 22 + 30) / 3 = 24

4.응용 SW 기초기술 활용-SEC_10(주요 스케줄링 알고리즘)

3) HRN(Hightest Response-ratio Next)

; 실행 시간이 긴 프로세스에 불리한 SJF기법을 보완하기 위한 것으로, 대기 시간과 서비스(실행) 시간을 이용하는 기법이다.

- 우선순위 계산 공식을 이용하여 서비스(실행) 시간이 짧은 프로세스나 대기 시간이 긴 프로세스에게 우선순위를 주어 CPU를 할당한다.
- 서비스 실행 시간이 짧거나 대기 시간이 긴 프로세스일 경우 우선순위가 높아진다.
- 우선순위를 계산하여 그 숫자가 가장 높은 것부터 낮은 순으로 우선순위가 부여된다.
- 우선순위 계산식

$$\text{우선순위 계산식} = \frac{\text{대기 시간+서비스 시간}}{\text{서비스 시간}}$$

4.응용 SW 기초기술 활용-SEC_10(주요 스케줄링 알고리즘)

3) HRN(Hightest Response-ratio Next)

예제) 다음과 같은 프로세스가 HRN기법으로 스케줄링 될 때 우선순위를 계산하시오.

프로세스 번호	P1	P2	P3
실행 시간	20	4	6
대기 시간	10	20	10
우선순위 계산	(20+10)/20=1.5	(4+20)/4=6	(6+10)/6=2.6
우선순위	P2 → P3 → P1		

$$\text{우선순위 계산식} = \frac{\text{대기 시간+서비스 시간}}{\text{서비스 시간}}$$

위의 공식대로 계산하면 우선순위는 P2 → P3 → P1 가 된다.

응용 SW 기초기술 활용 - SEC_10(주요 스케줄링 알고리즘) 기출 및 출제 예상 문제

기출 및 출제 예상 문제(주요 스케줄링 알고리즘)

1. 다음과 같은 프로세스가 차례로 큐에 도착하였을 때, SJF (Shortest Job First) 정책을 사용할 경우 가장 먼저 처리되는 작업은?

프로세스 번호	실행시간
P1	6
P2	8
P3	4
P4	3

① P1 ② P2

③ P3 ④ P4

SJF는 준비상태 큐에서 기다리고 있는 프로세스들 중에서 실행 시간이 가장 짧은 프로세스에게 먼저 CPU 를 할당하는 기법이다.

▶ **가장 적은 평균 대기 시간을 제공해 주는 최적의 알고리즘이다.**

▶ 실행 시간이 긴 프로세스는 실행 시간이 짧은 프로세스에게 할당 순위가 밀려 무한 연기 상태가 발생될 수가 있다.

하여, P4가 실행시간 3초이기에 가장 먼저 처리된다.

2. HRN 스케줄링에서 우선순위 계산식으로 올바른 것은?

3. HRN 방식으로 스케줄링 할 경우, 입력된 작업이 다음과 같을 때 처리되는 작업 순서로 옳은 것은?

작업	대기 시간	서비스(실행) 시간
A	5	20
B	40	20
C	15	45
D	20	2

① A→B→C→D ② A→C→B→D

③ D→B→C→A ④ D→A→B→C

HRN 방식의 우선순위 계산식 : (대기 시간 + 서비스(실행) 시간) / 서비스(실행) 시간

A 의 우선 순위 : (5 + 20) / 20 = 1.25

B 의 우선 순위 : (40 + 20) / 20 = 3

C 의 우선 순위 : (15 + 45) / 45 = 1.3333

D 의 우선 순위 : (20 + 2) / 2 = 11

숫자가 가장 높은 것부터 낮은 순으로 우선순위가 부여된다.

4. HRN(Highest Response-ratio Next) 스케줄링 방식에 대한 설명으로 옳지 않은 것은?

기출 및 출제 예상 문제(주요 스케줄링 알고리즘)

5. 다음에서 설명하는 프로세스 스케줄링은?

최소 작업 우선(SJF) 기법의 약점을 보완한 비선점 스케줄링 기법으로 다음과 같은 식을 이용해 우선순위를 판별한다.

우선순위 = (대기한 시간 + 서비스를 받을 시간) / 서비스를 받을 시간

① FIFO 스케줄링 ② RR 스케줄링

③ HRN 스케줄링 ④ MQ 스케줄링

'우선 순위'란 단어가 나오면 바로 HRN 스케줄링을 떠올려야 한다.

FCFS(First Come First Served) : **FIFO**(First Input First Out)와 같다. 말 그대로, 먼저 준비상태 큐에 들어온 순서대로 프로세서에 할당되어 처리되는 방식이다.

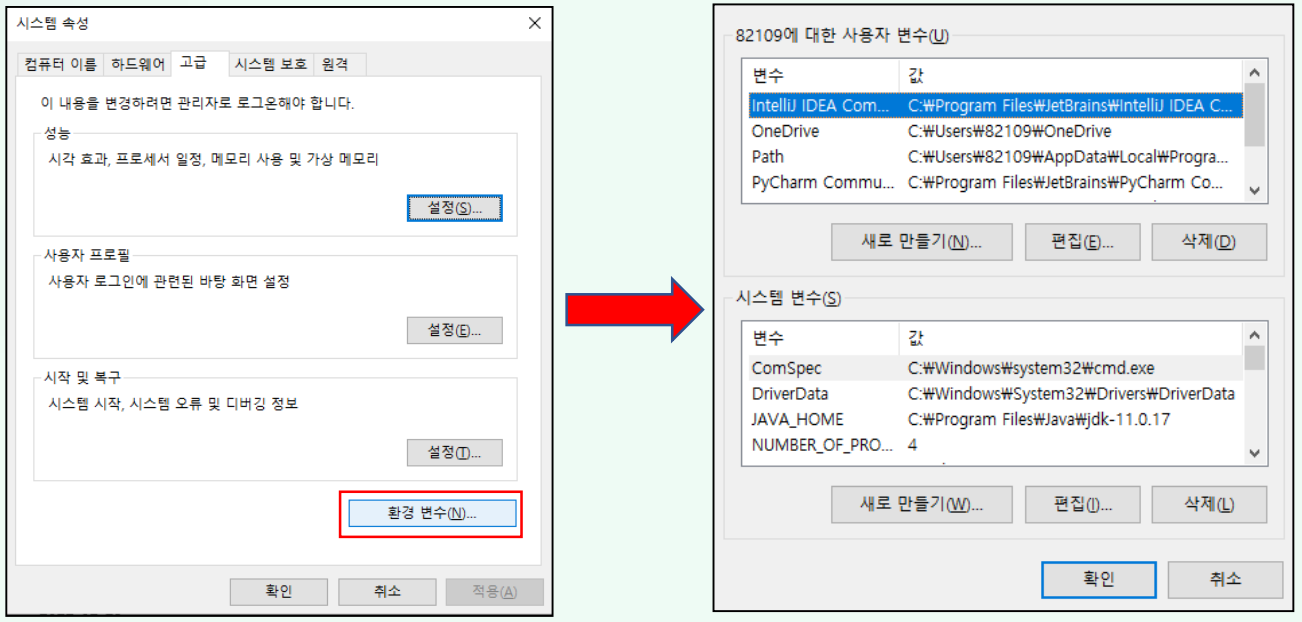
RR(라운드 로빈, Round Robin) : 프로세스 마다 같은 크기의 **CPU 시간을 할당(시간 할당량, Time Slice), 보통 10 ~ 100MS 정도** 이다. 프로세스가 할당된 시간 내에 처리 완료를 못하면 준비상태 큐 리스트의 가장 뒤로 보내지고, CPU 는 대기중인 다음 프로세스로 넘어가며, 균등한 CPU 점유 시간을 보장하며**, 시분할 시스템**에 사용된다.

4.응용 SW 기초기술 활용-SEC_11(환경 변수)

1) 환경 변수의 개요

; 환경 변수(Environment Variable)란 시스템 소프트웨어의 동작에 영향을 미치는 동적인 값들의 모임을 의미한다.

- 환경 변수는 변수명과 값으로 구성된다.
- 환경 변수는 시스템의 기본 정보를 저장한다.
- 환경 변수는 자식 프로세스에 상속된다.
- 환경 변수는 시스템 전반에 걸쳐 적용되는 시스템 환경 변수와 사용자 계정 내에서만 적용되는 사용자 환경 변수로 구분된다.



환경 변수를 이용하는 소프트웨어를 만든 경우 소프트웨어가 실행될 때 시스템에 저장된 환경 변수를 불러와 사용하게 된다. 예를 들어 'USERNAME' 이라는 환경 변수를 이용하여 "[USERNAME]님 안녕하세요"라는 메시지를 출력하는 소프트웨어를 만든 경우, 'USERNAME'에 'SKJ'가 저장된 PC에서 소프트웨어를 실행하면, "SKJ님 안녕하세요" 가 출력되는 것이다.

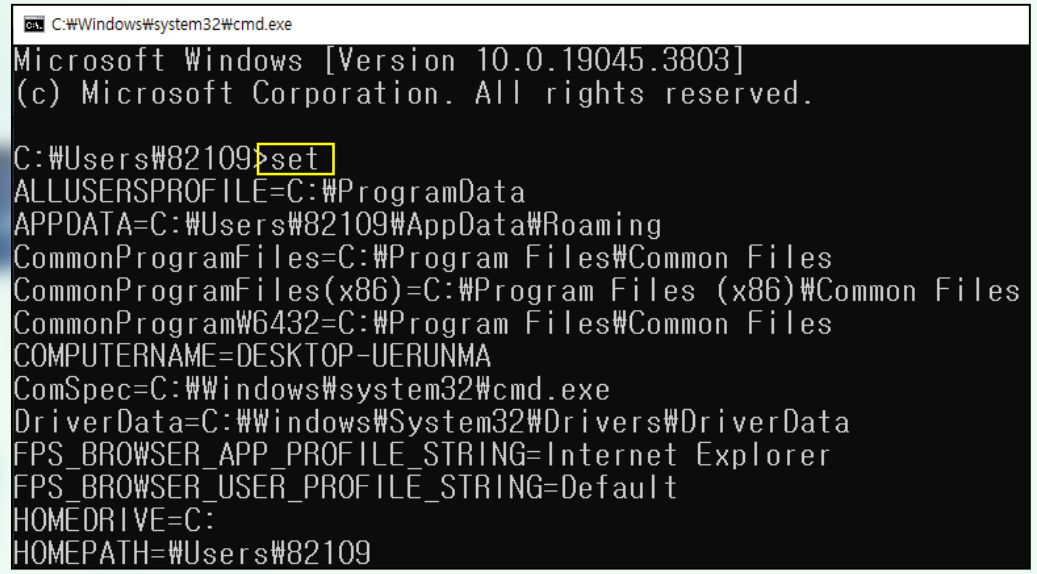
사용자 변수(User variables) : 로그인 한 계정 에 적용, 동일한 변수가 발생 시 1순위
시스템 변수(System variables) : OS 전체 적용, 동일한 변수가 발생 시 2순위

4.응용 SW 기초기술 활용-SEC_11(환경 변수)

2) Windows의 주요 환경 변수

; Windows에서 환경 변수를 명령어나 스크립트에서 사용하려면 변수명 앞뒤에 '%'를 입력해야 한다.

● Windows에서 명령 프롬프트 창에서 set을 입력하면 모든 환경 변수와 값을 출력한다.



환경 변수	용도
%ALLUSERPROFILE%	모든 사용자의 프로필이 저장된 폴더
%APPDATA%	설치된 프로그램의 필요 데이터가 저장된 폴더
%ComSpec%	기본 명령 프롬프트로 사용할 프로그램명
%HOMEDRIVE%	로그인한 계정의 정보가 저장된 드라이브
%HOMEPATH%	로그인한 계정의 기본 폴더
%LOGONSERVER%	로그인한 계정이 접속한 서버명
%PATH%	실행 파일을 찾는 경로
%PATHEXT%	cmd에서 실행할 수 있는 파일의 확장자 목록
%PROGRAMFILES%	기본 프로그램의 설치 폴더
%SYSTEMDRIVE%	Windows가 부팅된 드라이브
%SYSTEMROOT%	부팅된 운영체제가 들어 있는 폴더
%TEMP% 또는 %TMP%	임시 파일이 저장되는 폴더
%USERDOMAIN%	로그인한 시스템의 도메인명
%USERNAME%	로그인한 계정 이름
%USERPROFILE%	로그인한 유저의 프로필이 저장된 폴더명

Windows의 주요 환경 변수

4.응용 SW 기초기술 활용-SEC_11(환경 변수)

3) UNIX / LINUX의 주요 환경 변수

; UNIX나 LINUX에서 환경 변수를 명령어나 스크립트에서 사용하려면 변수명 앞에 '$'를 입력해야 한다.

● UINIX나 LINUX에서는 set, env, printenv, setenv 중 하나를 입력하면 모든 환경 변수와 값을 표시한다.

환경 변수	용도
$DISPLAY	현재 X 윈도 디스플레이 위치
$HOME	사용자의 홈 디렉터리
$LANG	프로그램 사용 시 기본적으로 지원되는 언어
$MAIL	메일을 보관하는 경로
$PATH	실행 파일을 찾는 경로
$PS1	쉘 프롬프트 정보
$PWD	현재 작업하는 디렉터리
$TERM	로긴 터미널 타입
$USER	사용자의 이름

X 윈도는 UNIX 계열의 운영체제에서 사용되는 GUI기반의 시스템 소프트웨어를 의미한다.

응용 SW 기초기술 활용 - SEC_11(환경 변수) 기출 및 출제 예상 문제

기출 및 출제 예상 문제(주요 스케줄링 알고리즘)

1. UNIX SHELL 환경 변수를 출력하는 명령어가 아닌 것은?

① configenv ② printenv

③ env ④ setenv

UNIX나 LINUX에서 모든 환경 변수와 값을 표시하려면 쉘(Shell) 에서 **set, env, printenv, setenv**를 입력해야 한다. configenv 명령어는 존재하지 않는다. Windows에서는 **set**을 입력해야 한다.

2. 다음 중 환경 변수에 대한 설명으로 가장 옳지 않은 것은?

① 시스템의 기본 정보를 저장한다.

② 기본적으로 부모 프로세스에서 상속 받아 사용한다.

③ 변수명과 값으로 구성된다.

④ 종류로는 시스템 환경 변수, 사용자 환경 변수, 인터페이스 환경 변수가 있다.

환경 변수(Environment Variable)라는 것은 시스템 소프트웨어의 동작에 영향을 미치는 동적인 값들의 모임을 의미한다.

▶ 환경 변수는 변수명과 값으로 구성된다.

▶ 환경 변수는 시스템의 기본 정보를 저장한다.

3. Windows에서 환경 변수를 명령어나 스크립트에서 사용하기 위해 변수명 앞, 뒤에 입력해야 하는 특수문자는?

① % ② $

③ # ④ &

Windows에서는 변수명 앞, 뒤에 %를 붙여야 되고, UNIX/LINUX에서는 변수명 앞에 $를 입력해야 한다는 것을 기억하자.

4. 다음 중 Windows에서 사용하는 환경 변수가 아닌 것은?

① PATH ② SYSTEMDRIVE

③ PWD ④ TEMP

PWD는 UNIX/LINUX에서 사용하는 환경 변수이다.

윈도우의 환경 변수

%ALLUSERPROFILE% : 모든 사용자의 프로필이 저장된 폴더

%APPDATA% : 설치된 프로그램의 필요 데이터가 저장된 폴더

%ComSpec% : 기본 명령 프롬프트로 사용할 프로그램명

%HOMEDRIVE% : 로그인 한 계정의 정보가 저장된 드라이브

%HOMEPATH% : 로그인 한 계정의 기본 폴더

%LOGONSERVER% : 로그인 한 계정이 접속한 서버명

%PATH% : 실행 파일을 찾는 경로

응용 SW 기초기술 활용 - SEC_11(환경 변수) 기출 및 출제 예상 문제

기출 및 출제 예상 문제(주요 스케줄링 알고리즘)

5. 다음 중 UNIX에서 사용하는 환경 변수가 아닌 것은?

① ProgramFiles ② DISPLAY

③ TERM ④ PS1

ProgramFiles는 Windows에서 기본 프로그램이 설치되어 있는 폴더를 저장하는 환경 변수이다.

UNIX/LINUX에서 사용하는 환경 변수

$DISPLAY : 현재 X 윈도 디스플레이 위치

$HOME : 사용자의 홈 디렉터리

$LANG : 프로그램 사용 시 기본적으로 지원되는 언어

$MAIL : 메일을 보관하는 경로

$PATH : 실행 파일을 찾는 경로

$PWD : 현재 작업하는 디렉터리

$TERM : 로긴 터미널 타입

$USER : 사용자의 이름

X 윈도는 UNIX 계열의 운영체제에서 사용되는 GUI 기반의 시스템 소프트웨어이다.

감사합니다.