1. **저장장치 및 파일 관리 개관**

개관

저장장치에 대해 더 자세히 알아보기 디스크의 효율적 관리가 매우 중요

파일 관리자는 시스템 내에 존재하는 파일을 제어하고 관리 학습목표

1. **저장장치의 종류를 이해한다.**
2. **다양한 디스크 스케줄링 알고리즘을 이해한다.**
3. **파일 관리자에 대해 이해한다.**
4. **디스크 공간 할당을 이해한다. 주요용어**

**순차접근 직접접근 FCFS**

# SSTF SCAN LOOK SLTF

순차 파일

직접 파일 인덱스된 순차 파일

* 1. **저장장치의 종류 [181]**
     1. **순차접근 저장장치**

데이터를 순차적으로 기록하고 판독하는 저장장치 예시

테이프 장치

테이프 위에 데이터를 일렬로 기록하고 읽을 때도 순차적으로 읽게 된다.

특징

초기 접근시간이 굉장히 오래 걸림

원하는 위치가 나올 때까지 테이프를 풀어야 함. 대량의 데이터 백업(backup)용으로 사용됨

* + 1. **직접접근 저장장치**

**임의접근 저장장치(random access storage devices)**

지정한 위치를 직접 찾아 데이터를 읽거나 쓸 수 있는 장치 종류

**자기 디스크(magnetic disk)**

자성을 띤 디스크의 표면에 데이터를 쓰거나 읽을 수 있는 직접접근 저장장치 구성(1)

**플래터(platter)**

**한 장 이상. 간격을 두고 쌓여 있는 형태 헤드(head)가 암(arm)에 연결**

각 플래터마다 기록하고 판독 암(arm)은 고정축에 연결

**구성(2)**

데이터가 저장되는 각 플래터의 표면 트랙

플래터의 중심축을 기준으로 한 동심원 형태 섹터

**일정한 용량을 저장할 수 있는 호(arc) 형태**

**실린더(cylinder)**

**플래터가 여러 장인 경우, 각 플래터의 중심축으로부터 같은 거리에 있는 트랙들을 모아 하나의 실린더를 구성 광디스크(optical disk)**

자성 대신에 빛을 이용하는 것

디스크 표면에 레이저를 쏘아 반사되는 빛의 차이를 이용하도록 데이터를 읽고 쓰는 직접접근 저장장치 광디스크는 동심원의 트랙들로 구성된 자기 디스크와는 다르게 나선형인 하나의 트랙으로 구성

예시

**CD-ROM, DVD-ROM, 블루레이 디스크**

# SSD

플래시 메모리처럼 읽고 쓰기가 가능하면서 전력공급이 없어도 데이터가 지워지지 않는 메모리를 이용하여 자기 디스크처 럼 만든 직접접근 저장장치

특징

자기 디스크보다 1) 속도가 빠름 2) 물리적인 장치가 없어 전력 소모가 적ㅇ므

용량 대비 가격이 비싸며 수명이 짧음

* 1. **디스크 스케줄링 알고리즘[184] 디스크 스케줄링(disk scheduling)**

디스크 접근 요구를 효율적으로 처리하는 순서를 결정하는 작업 디스크 접근 요구

디스크에서 데이터를 읽거나 쓰는 요구

다중 프로그래밍 환경에서는 많은 프로세스가 디스크 접근 요구를 발생시킬 수 있어, 이러한 요구들을 "디스크 큐"에 두고 관 리함.

디스크의 특징, 디스크 스케줄링의 필요성

디스크는 헤드의 이동, 디스크의 회전과 같은 "기계적인 움직임"에 의해 직접 접근을 하기 때문에, 디스크 스케줄러는 대기하고 있는 요구들 간의 위치적 관계를 조사하고 나서 최소한의 기계적 동작에 의해 접근 요구를 처리할 수 있도록 디스크 큐를 재배열 함.

**디스크 접근 요구를 처리하는 데 필요한 시간의 요소 탐구시간(seek time)**

기계적인 동작에 의해 암을 움직여 헤드를 원하는 트랙에 위치시키는 데 걸리는 시간 세 가지 요소 중 가장 느리다.

**최적화 대상이 됨. (탐색 시간이 그 다음 우선순위) 회전지연시간(rotational latency time) = 탐색 시간 (search time)**

헤드가 위치한 트랙에서 요구된 자료가 헤드 밑에 이를 때까지 디스크가 회전하는 데 걸리는 시간 탐색시간(search time)

현재 트랙의 헤드 위치에서 자료를 읽거나 쓰는 데 걸리는 시간 스케줄링 방법과 무관하게 항상 동일하게 필요한 부분

* + 1. **FCFS 스케줄링**

**먼저 도착한 접근 요구가 먼저 서비스를 받는 방법 예제에서 총탐구시간 1+5+15+20 = 41**

특징

실행 예정순서가 고정된다는 점에서 공평 총탐구시간이 커질 수 있다.

헤드의 현재 위치에 가까운 접근 요구라 하더라도 도착순서가 뒤에 있다면, 앞의 접근 요구를 먼저 처리한 후 접근 디스크의 부하가 낮을 때는 사용할 만하나, 부하가 높을수록 응답시간이 길어지는 단점

# SSTF(Shortest Seek Time First) 스케줄링

**SSTF(Shortest Seek Time First) 스케줄링 알고리즘 탐구시간이 가장 짧은 접근 요구 먼저 처리**

총 탐구시간

# 1+5+5+20 = 31

특징

장점

**FCFS 스케줄링보다 처리량이 많다**

평균응답시간이 비교적 짧다 일괄처리 운영체제에는 적합

단점

양 끝쪽에 위치하는 트랙은 기아상태에 빠질 수 있다.

응답시간의 편차가 크기때문에 시분할 시스템에서는 사용되지 않는다.

* 1. **디스크 스케줄링 알고리즘 (2) [187]**
     1. **SCAN 스케줄링**

양 끝 트랙 사이를 왕복하며 진행방향의 가장 가까운 접근 요구를 먼저 처리 예제 총 탐구시간

# 31 + 40 = 71

특징

장점

**(SSTF 스케줄링의) 응답시간 편차를 어느 정도 해소 단점**

헤드 바로 뒤에 도착한 접근 요구는 헤드가 디스크 마지막 트랙까지 갔다가 방향을 바꾸어 되돌아올 때까지 기다려야 하는 불공평한 문제

트랙의 양 끝은 헤드가 한 번 왕복할 때 한 번의 서비스 기회만 있다

# C-SCAN(Circular SCAN) 스케줄링

오로지 한쪽 방향으로만 진행방향의 가장 가까운 접근 요구를 처리한다는 것을 제외하면 SCAN 스케줄링과 동일 예제 총 탐구시간

특징

장점

양 끝 트랙에 대한 접근 요구의 차별을 제거 응답시간의 편차도 매우 적다

* + 1. **LOOK 및 C-LOOK 스케줄링**

**SCAN 스케줄링과 C-SCAN 스케줄링의 변형으로, 진행방향의 앞쪽에 더 이상 접근 요구가 없으면 바로 방향을 바꾸는 방법**

**LOOK 스케줄링 알고리즘**

**SCAN 스케줄링의 변형**

20번 트랙을 서비스한 후 앞을 보고 더 이상 접근 요구가 없으므로 바로 방향을 바꿈

**C-LOOK 스케줄링 알고리즘**

**C-SCAN 스케줄링의 변형**

* + 1. **SLTF 스케줄링**

**SLTF(Shortest Latency Time First) 스케줄링 알고리즘 회전지연시간 최적화**

(다른 것들은 총탐구시간 최적화)

동일 실린더의 여러 섹터에 대한 접근 요구에 대해 회전지연시간이 가장 짧은 것을 먼저 처리하는 방법 예제

헤드가 위치한 실린더(혹은 트랙)에 3개의 접근 요구가 있고 디스크의 회전방향을 고려할 때, 1번의 회전지연시간이 가장

**짧고, 그다음이 2번, 마지막이 3번이 된다.**

**따라서 3개의 접근 요구는 1, 2, 3의 순서대로 처리하게 된다.**

장점

높은 부하상태에서 유용한 방법

이론적인 최적해와 거의 일치하고 구현도 비교적 쉬움

* 1. **파일 관리 [192]**
     1. **파일 관리자의 요소**

**파일 관리자(file manager)**

**파일에 대한 1) 생성/삭제/수정 제어 2) 접근 제어 3) 파일에 의해 사용되는 자원 관리 파일 관리자의 요소 (X4)**

액세스 방식

파일에 저장되어 있는 데이터에 접근하는 방식 지정 파일 관리

파일을 저장, 참조, 공유할 수 있도록 하며, 안전하게 보호될 수 있도록 한다. 보조기억장치 관리

파일을 저장하는 데 필요한 공간 할당 파일 무결성 유지

파일 정보가 소실되지 않도록 보장 보조 기억장치 중에서 특히 디스크 장치를 관리

**2단계 계층구조 파일 시스템의 구조 루트(root)**

**디스크의 루트 디렉터리(root directory)가 시작되는 위치 루트 디렉터리**

**여러 개의 사용자 디렉터리(user directory)를 가짐 사용자 디렉터리**

그 사용자의 각 파일당 하나의 항목 항목

디스크상에 파일이 저장되어 있는 실제 위치

특징

**파일의 시스템 이름은 일반적으로 루트 디렉터리로부터 그 파일에 이르는 경로 이름(path name)으로 구성 계층구조 파일 시스템(hierarchically structured file system)에서 파일 시스템 이름**

**일반적으로 루트 디렉터리로부터 그 파일에 이르는 경로 이름(path name)으로 구성**

**예를 들어, OS, PL 등의 사용자를 갖는 2단계 파일 시스템에서 사용자 OS가 PROC 파일과 SCHE 파일을 가지고 있 을 대, PROC 파일의 경로 이름은 OS/PROC로 구성할 수 있고, 이 경우 PROC 파일의 시스템 이름도 OS/PROC이 된다.**

* + 1. **파일 관리자의 기능 파일 관리자의 역할**

보조기억장치를 활성화시켜 파일을 할당

파일의 기록을 갱신하는 동안에는 파일을 메모리에 적재

테이블을 갱신하거나 수정된 사항이 있다면 파일을 보조기억장치의 같은 장소에 다시 쓰고 그 파일을 해지 구체적인 파일 관리자의 기능

사용자가 파일을 생성, 수정 및 삭제할 수 있게 하는 기능 타인의 파일을 공동으로 사용할 수 있게 하는 기능

읽기, 쓰기, 실행 및 그 조합까지 여러 종류의 액세스 제어방법 제공 사용자가 각 응용에 적합한 구조로 파일을 구성할 수 있게 하는 기능 백업 및 복구 기능

물리적인 장치 이름 대신 기호화된 이름을 사용하여 파일을 참조 정보가 안전하게 보호되고 비밀이 보장되게 하는 기능

* 1. **파일 관리(2) [194]**
     1. **파일 구조와 접근방식 파일 구조**

파일을 구성하는 레코드들이 보조기억장치에 배치되는 방식 파일 구조의 종류

* + - 1. **순차 파일(sequential file)**

레코드가 물리적 순서에 따라 저장되어 있는 파일 논리적인 레코드의 순서와 물리적인 레코드의 순서가 동일 순차적 기록, 판독하는 테이프 장치에 가장 많이 이용

* + - 1. **인덱스된 순차 파일 (indexed sequential fil)**

각 레코드의 키를 기준으로 한 논리적 순서대로 레코드가 저장, 일부 주요 레코드의 실제 주소가 저장된 인덱스를 구성하여 관리

특징

키 순서에 의해 순차적으로 액세스

인덱스의 검색을 통해 직접 액세스할 수도 있음

* + - 1. **직접 파일 (direct file)**

각 레코드의 키를 이용하여 직접접근 저장장치의 물리적 주소를 통해 직접 액세스되는 파일 논리적인 키와 물리적 주소의 사상은 프로그래머가 직접 정의

**4, 디스크 공간 할당**

1. **연속 할당 기법**

보조기억장치의 연속된 가용공간에 파일 저장공간을 할당 사용자는 생성하려는 파일의 저장공간 크기를 미리 정해야 한다.

만약 보조기억장치의 남아 있는 공간들 중 그 크기 이상의 연속된 공간이 없을 경우 그 파일을 생성할 수 없다. 장점

액세스가 효율적: 논리적으로 연속된 레코드들이 보조기억장치에 물리적으로도 서로 인접하게 저장 파일의 디렉터리 구현이 수월: 디렉터리에는 각 파일의 시작주소와 파일의 길이만 유지하면 됨

단점

외부 단편화: 파일이 삭제되면 그 파일이 차지하고 있던 공간은 회수되지만, 새로 생성된느 파일의 크기와 같은 가능성이 희박하므로 메모리의 동적 분할과 같은 외부 단편화 문제 발생

주기적으로 보조기억장치에 대한 집약을 하여 새로운 파일을 넣을 수 있을 큰 공간으로 만들어야 함 파일의 크기 확장에 대한 대응이 비효율적

파일은 시간이 지남에 따라 커지거나 작아질 수 있다.

할당된 공간보다 파일의 크기가 커지면 그 파일을 저장할 수 있는 다른 공간으로 파일을 옮겨야 하는 불편이 있다.

1. **불연속 할당 기법**

섹터 또는 블록(정해진 수의 섹터) 단위로 공간을 할당 블록들은 포인터를 이용하여 연결

파일의 크기가 커지면 현재 파일이 저장되어 있는 블록으로부터 가장 가까운 거리에 있는 블록을 선택하여 그 파일에 추가 할당하고 포인터로 연결

장점

(연속 할당 기법의 문제였던) 단편화 문제 발생 X

파일 확장에 유연함 단점

파일 공간이 분산되어 논리적으로 연속된 레코드들을 검색하는 경우 성능저하가 발생

포인터 관리를 위한 연산 및 공간이 필요하여 추가비용 및 실제 데이터를 저장할 공간이 감소됨

요약

1. **저장장치는 순차접근 저장장치와 직접접근 저장장치로 나누니다.**
2. **직접접근 저장장치인 자기 디스크의 접근 요구 처리시간은 탐구시간, 회전지연시간, 전송시간으로 구성된다.**
3. **디스크 스케줄링은 디스크 접근 요구를 효율적으로 처리하는 순서를 결정하는 작업으로, 탐구시간을 최소화하는 것이 가장 중요 하다.**
4. **SSTF 스케줄링 알고리즘은 헤드 위치로부터 탐구시간이 가장 짧은 접근 요구를 먼저 처리하는 방법이다.**
5. **SCAN 스케줄링 알고리즘은 양 끝 트랙 사이를 왕복하여 진행방향의 가장 가까운 접근 요구를 먼저 처리하는 방법이다.**
6. **C-SCAN 스케줄링 알고리즘은 정해진 한 방향으로만 가장 가까운 접근 요구를 먼저 처리하는 방법으로, 트랙의 끝에 도달하면 반대편 트랙 끝으로 이동한 후 다시 같은 방향으로 진행하는 방법이다.**
7. **LOOK과 C-LOOK 스케줄링 알고리즘은 진행방향의 앞쪽에 더 이상 접근 요구가 없으면 바로 방향을 바꾸는 방법이다. 이때**

C-LOOK 스케줄링은 반대편 트랙 끝까지 이동하지 않고 가장 먼 접근 요구의 트랙까지만 이동한다.

1. **SLTF 스케줄링 알고리즘은 회전지연시간 최적화를 위한 알고리즘으로, 동일 실린더에서 회전지연시간이 가장 짧은 접근 요구를 먼저 처리하는 방법이다.**
2. **파일 관리자는 파일의 생성/수정/삭제, 파일의 공유 및 액세스 제어, 백업, 정보 보호 등의 기능을 수행한다.**