

6주차 3차시 파일 설계

【학습목표】

1. 내용 및 편성법에 따른 파일 구분을 할 수 있으며, 파일의 편성법의 원리를 설명할 수 있다.
2. 데이터의 기록 형태와 파일설계 이전에 검토 및 확정 사항에 대해 각각 설명할 수 있다.

학습내용1 : 파일의 종류 및 편성법

1. 정보의 단위

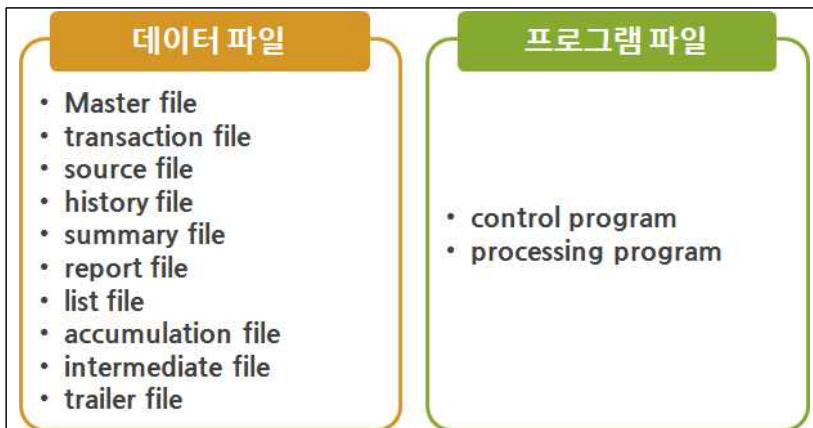
〈Bit ⇨Byte ⇨Word ⇨Logical Record(Record)⇨Physical Record(Block) ⇨File or Database〉

2. 파일의 종류

1) 수록매체에 따른 파일구분

- 자기 디스크 파일(Magnetic Disk File)
- 자기 테이프 파일(Magnetic Tape File)
- 디스켓 파일(Diskette File)
- 기타 파일

2) 내용에 따른 파일구분



* 작업 파일

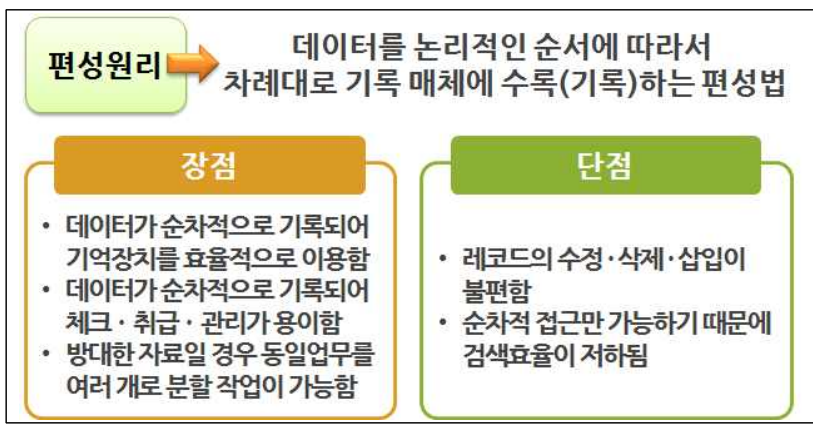
- input data file
- output data file
- work file
- temporary file
- check point file

3) 편성법에 따른 파일 구분

- ① 순차 파일(Sequential File)
- ② 색인순차 파일(Indexed Sequential File)
- ③ 랜덤 파일(Random File)
- ④ 인덱스 파일(Index File)
- ⑤ 직접 파일(Direct File)
- ⑥ 리스트 파일(List File)

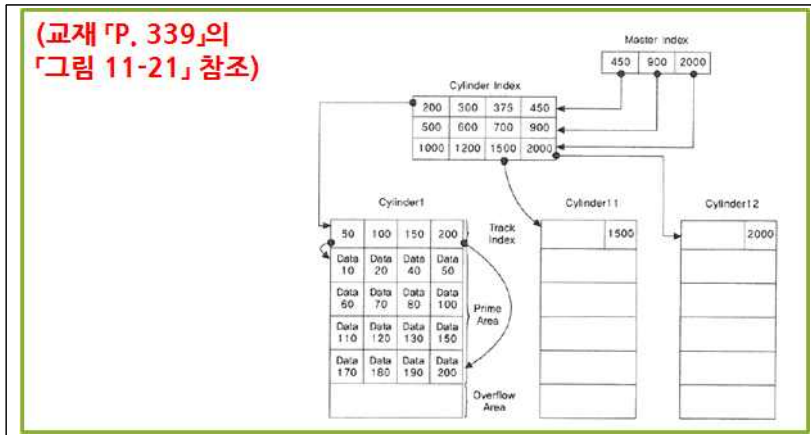
3. 파일의 편성법

1) 순차 편성(Sequential Organization)

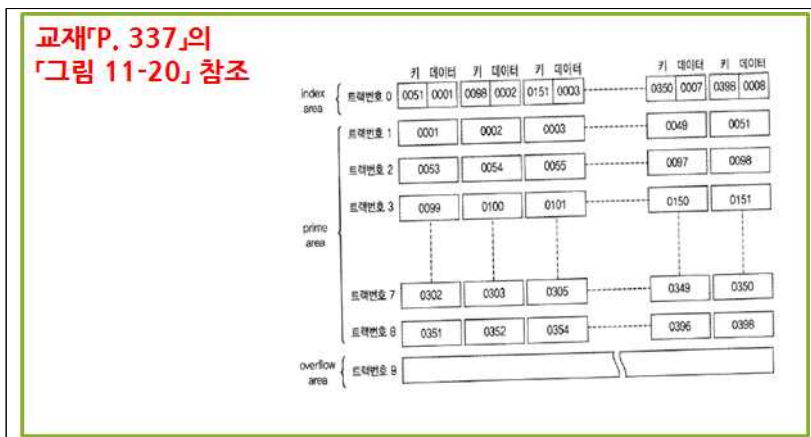


2) 색인순차 편성(Indexed Sequential Organization)

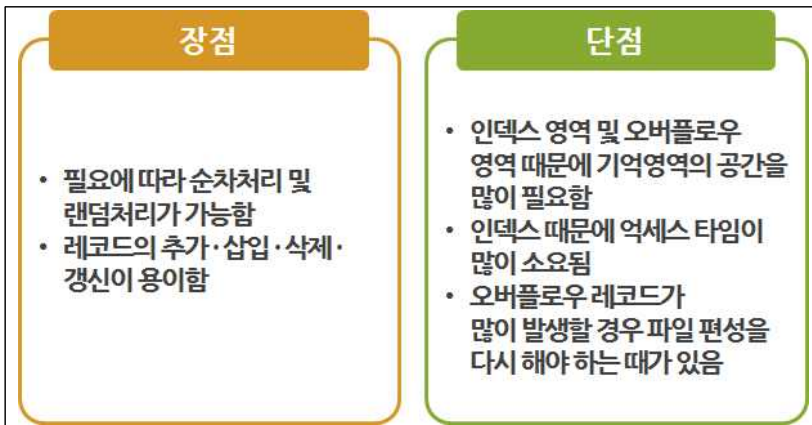
- 파일 편성 시에 색인(Index)을 붙여 편성해서 순차 액세스 및 랜덤액세스가 가능함
- 색인을 이용하여 순차적으로 액세스 하는 방법을 ISAM(indexed sequential access method)이라고 함
- 기본적 영역
 - Prime Area ➡ 데이터 수록(기록) 영역
 - Index Area
 - = Track Index
 - = Cylinder Index
 - = Master Index (교재 「P. 339」의 「그림 11-21」 참조)



- Overflow Area
 - = Cylinder Overflow Area
 - = Independent Overflow Area

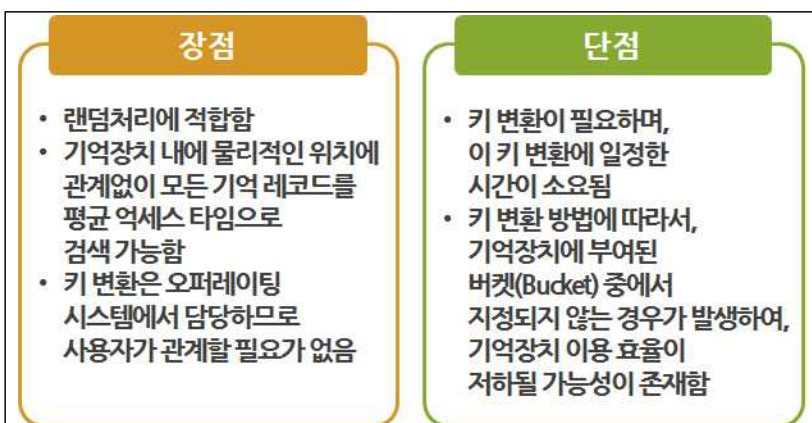
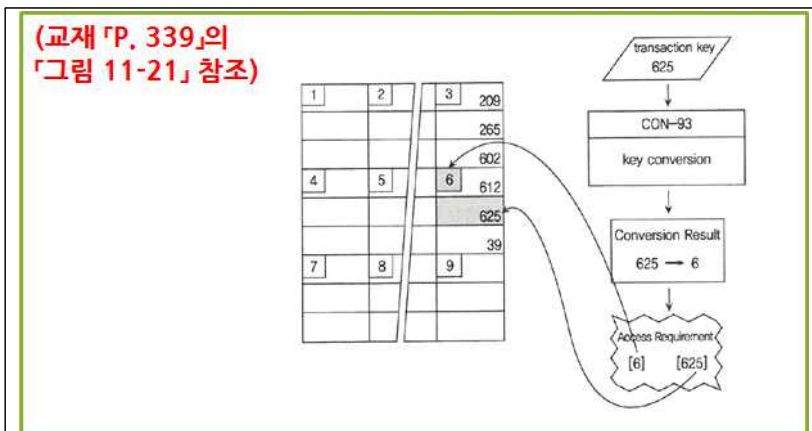


- 기본적 영역의 구성형태 교재「P. 337」의「그림 11-20」 참조



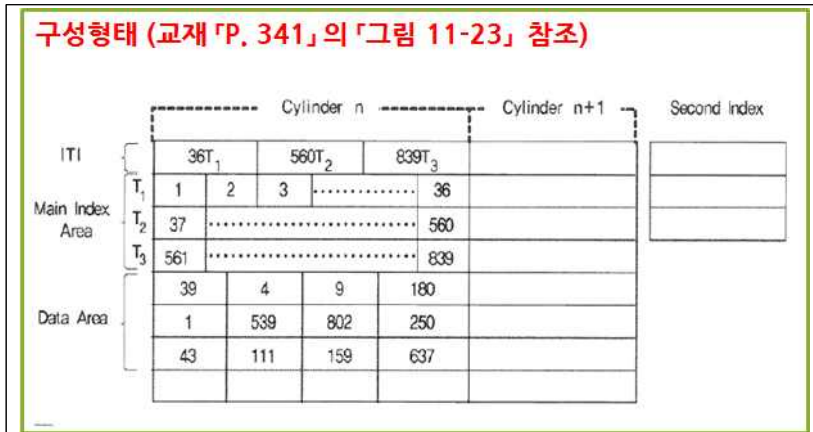
3) 랜덤 편성(Random Organization)

- 데이터 레코드를 키 항목(Key Item)을 중심으로 하여 랜덤처리가 가능하게 편성한 파일임
- 처리과정에서 키 변환(key Conversion)이 이루어짐
- 키 항목(key item)을 이용하여 데이터의 기록 위치 및 기록시킬 위치를 결정하거나 관리가 가능함
- 파일구성 형태 (교재 「P. 340」의 「그림 11-22」 참조)



4) 인덱스 편성(Index Organization)

- 파일을 「데이터 영역」과 「인덱스 영역」으로 구분하여 편성함
- 인덱스를 이용하여 랜덤처리 및 순차처리를 행함
- 구성형태 (교재 「P. 341」의 「그림 11-23」 참조)



- 인덱스 파일 구성 영역(Area)
 - Data Area
 - = 데이터가 수록되는 영역임
 - = 데이터를 키(key) 순으로 분류(sort)할 필요가 없음.
 - = 오버플로우 발생 시에 별도의 오버플로우 영역을 지정할 필요가 없음
 - Main Index Area
 - = 키(key) 항목의 값
 - = 그 키 항목의 데이터가 수록된 어드레스(address)에 대한 정보가 수록됨
 - ITI(index track index)
 - = Main Index Area가 1 트랙(track) 이상일 경우에 만들어짐
 - = ITI는 Main Index Area를 찾는데 효과적임
- * 인덱스 파일과 색인순차파일 비교시 차이점
 - 데이터를 키(Key) 항목 순으로 분류(Sort)하여 수록할 필요가 없음
 - 오버플로우(Overflow) 영역을 독립적으로 설정하지 않아도 됨
 - 인덱스 영역이 한 곳에 집중되어 있음
 - 필요하다면 여러 개의 인덱스를 설정할 수 있음

5) 직접 편성(Direct Organization)

<직접파일의 어드레스 지정법>



- * 간접 파일의 장점
 - 랜덤 액세스가 가능함
 - 평균 액세스 타임으로 검색됨
 - 레코드가 기억된 어드레스를 사용자가 직접관리 가능함

6) 직접 어드레스 방식

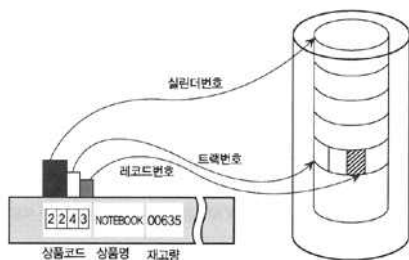
키 이용법

인덱스 탐색법

① 키(Key) 이용법

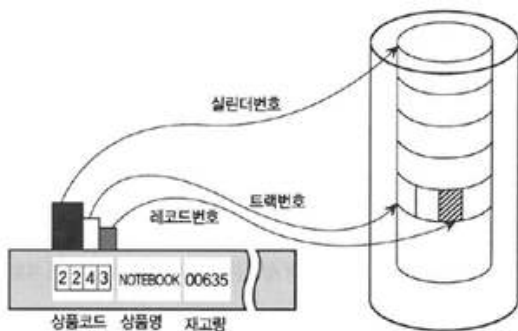
- 키항목(Key Item) 값으로 직접 기억장치 어드레스를 지정하는 방식 : 교재 「P. 343」의 「그림 11-24」 참조

교재 「P. 343」의 「그림 11-24」 참조



* 특징

- 키 항목 값이 다르기 때문에 시니넘(synonym)이 발생하지 않음
- 데이터가 키(key) 순으로 기록되어 순차처리가 가능함 : 키(key)에서 얻을 수 있는 레코드 어드레스로 랜덤 액세스가 가능함
- 키(key) 이용법의 문제점
 - 기억영역이 확장되면 ⇨데이터가 해당 영역에 기억된다고 보장 할 수 없음
 - 누락된 키 항목의 어드레스를 비워두면 공백이 많아질 수 있음 (교재 「P. 344」의 「그림 11-25」 참조)



- 키 항목의 값에 영문자나 특수문자가 포함되면 안됨
- 이 방식의 단점을 보완하기 위하여 「인덱스 탐색법」을 사용함

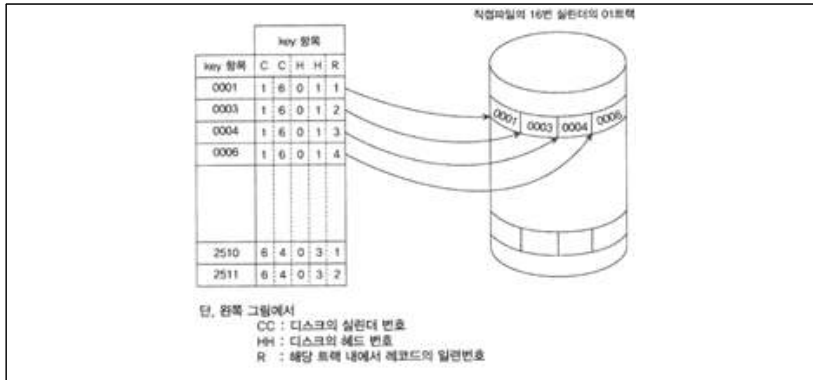
② 인덱스 탐색법(Index Search)

- 인덱스 탐색법

- 파일편성 과정에서 「데이터 레코드의 키 항목」과, 「그 데이터가 기록된 기억장치의 어드레스」를 모아서 「인덱스 테이블(Index Table)」을 만들

- 인덱스 테이블은 특정한 키(key) 값을 가지는 레코드가 어느 위치에 기억되는지 파악되기 때문에 필요한 처리가 쉬움

- 교재「P. 345」의「그림 11-26」 참조



* 특징

- 인덱스 테이블을 하나의 파일로 만들면 효과적으로 사용 가능함
- 데이터 개수가 많으면 인덱스 테이블 검색시간이 많이 소요됨
- 인덱스 테이블의 추가, 삭제 같은 관리가 복잡함
- 변동이 적고 데이터 양이 일정한 수준의 업무에 적합함

7) 간접 어드레스 방식

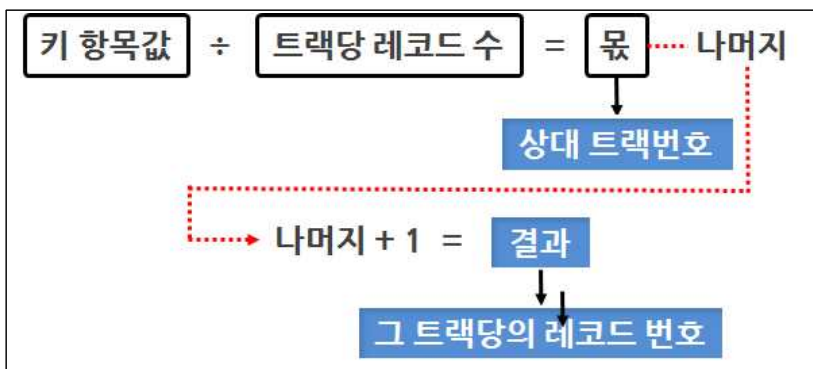
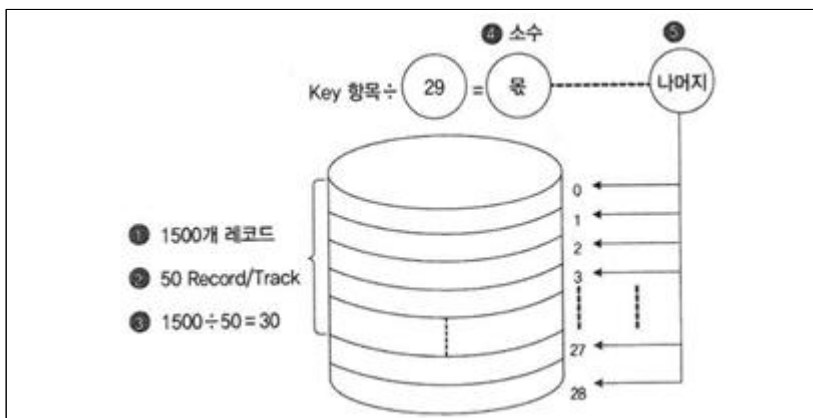
- ① 제산법
- ② 합산법
- ③ 접지법
- ④ 기수 변환법

① 계산법

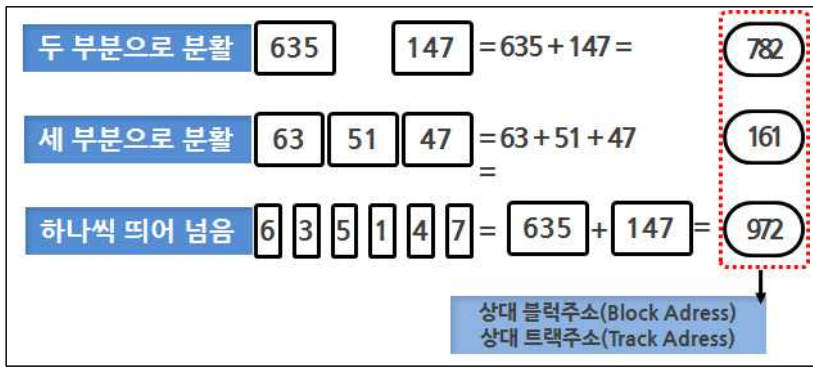


- 키 항목을 트랙당 레코드 수로 나누는 방식

- 원리 ⇨ 레코드에 「키 항목의 값」을 「트랙에 기록될 수 있는 레코드의 수」로 나누어
= 「몫」을 「상대트랙 번호」로 하고
= 「나머지에 1을 더한 값」을 해당 트랙 내에 기록되는 「레코드 번호」로 지정하는 방식임
- 나머지에 1을 더하는 이유는 자기 디스크의 Track Descriptor Record를 피하기 위한 조치임
- 해당 파일을 구상하는 레코드의 개수(個數)를 파악함 (예를 들어 1,000 개의 레코드)
- 트랙(track)당 기록 가능한 레코드 개수를 파악함 (예를 들어 20 레코드/트랙으로 가정함)
- 소요 트랙 수에서 가장 가까운 소수(prime number)를 구함 ⇨ 50에 가장 가까운 소수는 「47」임
- 「임의의 키 값」 ÷ 「소수」 = 「몫」 「소수」
- 위의 단계에서 구한 「나머지」 「상대트랙 번호」임
- 교재 「P. 347」의 「그림 11-29」 및 교재 「P. 370」 설명 내용 참조



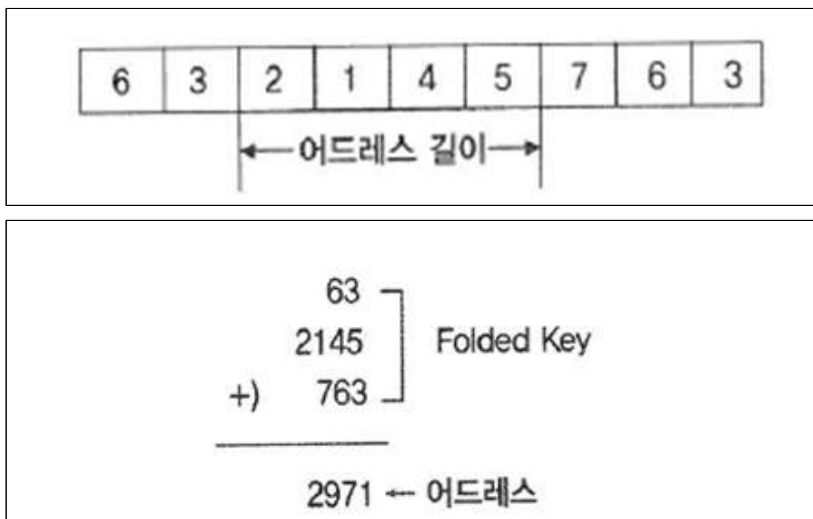
② 합산법(Addition Transformation)



③ 접지법(Folding Transformation)

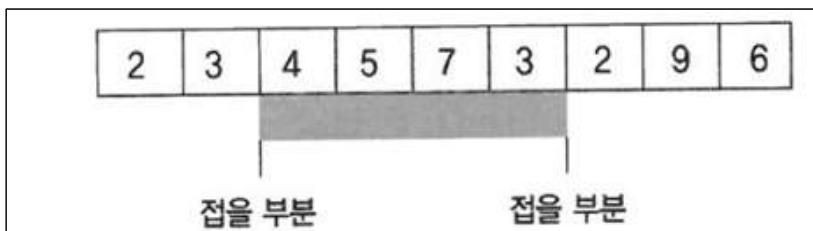
* Fold Shifting 방식

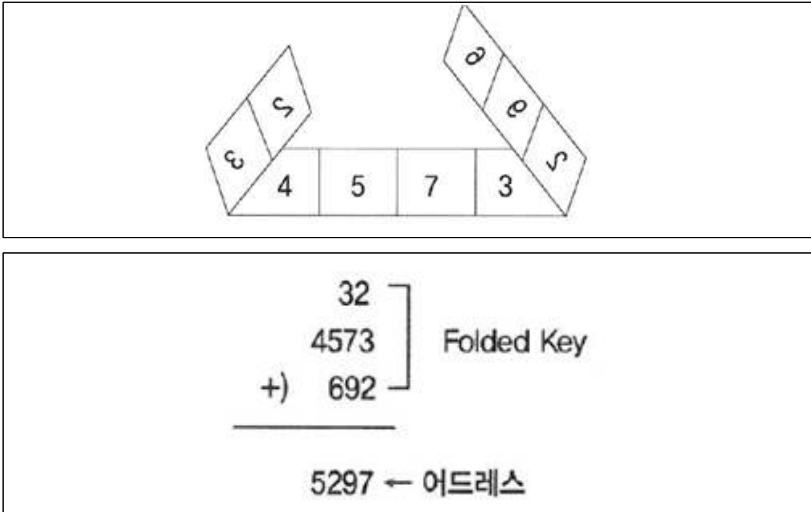
- 「키 항목의 값」을「어드레스 길이로 허용되는 자릿수」를 고려하여, 「몇 개 부분으로 분할」한 다음에, 「분할 된 각 부분」의 오른쪽 끝자리를 맞추어 더한 값을「홈어드레스」로 정하는 방법
- 교재 「P. 349」의 「그림 11-31 및 32」 참조



* Fold Boundary 방식

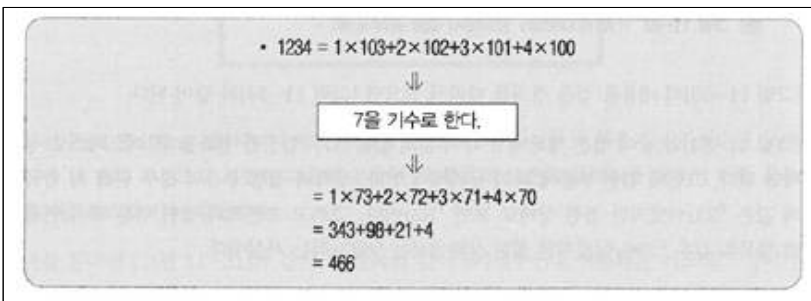
- 「키 항목의 값」을 「허용되는 어드레스의 자릿수」를 고려하여, 「몇 개의 부분으로 나누어」, 「각 부분의 경계선을 접어서」, 「각 부분의 자릿수를 맞추어 합산 결과」를 「홈 어드레스」로 정함
- 접힌 부분의 값이 바뀔에 주의해야 함 (교재 「PP. 349~350」의 「그림 11-33과 34, 35」 참조)





④ 기수 변환법

- 키 항목을 나타내는 숫자의 기수(Base)를 바꾸어 나타내는 방법
- 교재 「PP. 350~351」 내용 참조

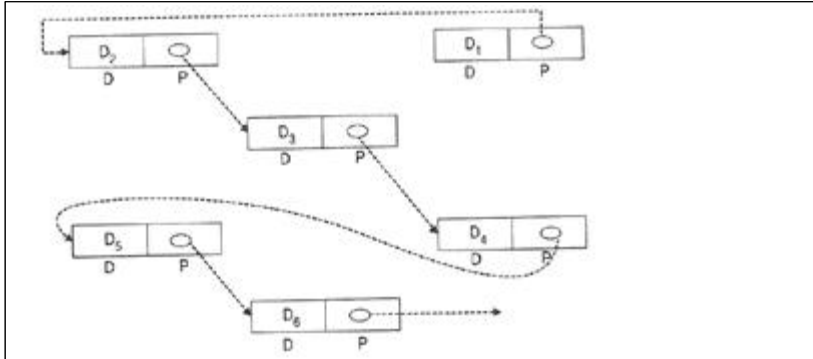


8) 간접주소 지정방식의 장단점

- 키 항목의 값을 사용자가 원하는 범위 내의 자릿수로 압축 가능함
- 키 항목의 값이 동일 어드레스로 변환되는 『충돌(Collision) 현상』 나타나며, 이를 『시니님(Synonym)』이라고 함
- 시니님 대책
 - 오버플로우 영역을 설정하여 여기에 기억시킴.
 - 기억영역을 블록으로 구분해서 각 블록에 『여분 영역』을 확보했다가 여기에 기억시킴

9) 리스트 편성(List Organization)

- 데이터 기록형태가 랜덤파일처럼 여러 곳에 분산되어 있지만 데이터 레코드에 붙어 있는 포인터(Pointer)가 차례대로 데이터 레코드의 어드레스를 지정하여 파일이 유지됨
- 교재 「P. 352」의 「그림 11-36」 참조



장점

- 데이터 레코드의 갱신·삭제가 용이함
- 데이터 레코드의 삽입이 간단함
- 왜냐하면 포인터의 내용을 변경하는 간단한 조작만 하면 되기 때문임

단점

- 처리효율이 저하됨
- 포인터로 인해서 데이터 레코드의 길이가 길어짐
- 파일구조가 복잡함

학습내용2 : 데이터의 기록형태 및 파일 설계 추진

1. 데이터의 기록형태

1) 데이터 기록형태

고정길이 레코드

- 비블록화 고정길이 레코드 (unblocked fixed length record)
- 블록화 고정길이 레코드 (blocked fixed length record)

가변길이 레코드

- 비블록화 가변길이 레코드 (unblocked variable length record)
- 블록화 가변길이 레코드 (blocked variable length record)

2) 입출력 기준

블록화 레코드
(blocking record)

비블록화 레코드
(unblocking record)

2. 파일설계 이전에 검토 및 확정 사항

① 기초사항의 확정

- 파일명(File Name) 결정
- 파일의 작성 목적 정의
- 파일의 종류 확정
- 사용 범위 결정

② 데이터 항목의 분석 · 검토

- 항목명(Item Name) 부여
- 자릿수 결정
- 항목의 배열 순서 결정
- 항목의 속성(Mode) 결정
- 레코드 형태 : 고정길이 혹은 가변길이
- 레코드 · 블록의 길이

③ 파일의 특성 · 조작

- 처리방식 : 순차처리 혹은 랜덤처리
- 처리 주기 · 시기
- 갱신형태 : 추가 · 삭제 · 삽입 등의 갱신을
- 처리시간 : 액세스 타임 파악
- 데이터 량

④ 신뢰성 확보 방안

- 기술적 측면
- 설비적 측면
- 운용 · 관리적 측면

3. 파일 매체 설계

1) 파일 매체 선정 시 고려사항

- ① 매체의 기억용량
- ② 매체 · 매체화 소요비용
- ③ 파일 처리에 소요시간
- ④ 조작의 용이성

2) 파일에서 요구되는 실제적 조건

- 파일이 요구하는 처리방식(순차 · 랜덤 · 겸용)에 따라 매체가 달라짐
- 요구되는 액세스 타임에 따라 매체가 달라짐
- 데이터 량에 따라 매체의 용량이 결정되며, 용량 산출법은 교재 「PP. 356~358」 참조
- 조작의 용이성을 고려해야하며, 데이터 량 · 파일종류 · 파일개수·사용빈도 등의 대소에 따라 매체를 결정해야함
- 이상의 조건이 결정된 이후 특정 매체 선정, 매체 개수, 장치 대수 등이 결정됨

4. 파일설계의 요약

1) 파일설계 결과는 「파일설계 기준항목의 문서화」가 되어야함
(교재「PP. 359~ 360」의「그림 11-41 및 42」 참조)

[illegible]

5. 데이터베이스

1) 별도의 데이터베이스 과목에서 취급하므로 생략함

학습내용3 : 프로세스 설계

1. 프로세스 설계(Process Design)

〈업무처리 효율화를 위하여 처리내용을 구분 혹은 분리하거나,구분된 단위업무의 연결 등을 고려하는 설계과정〉

2. 처리 패턴

1) 처리 패턴(Processing Pattern)

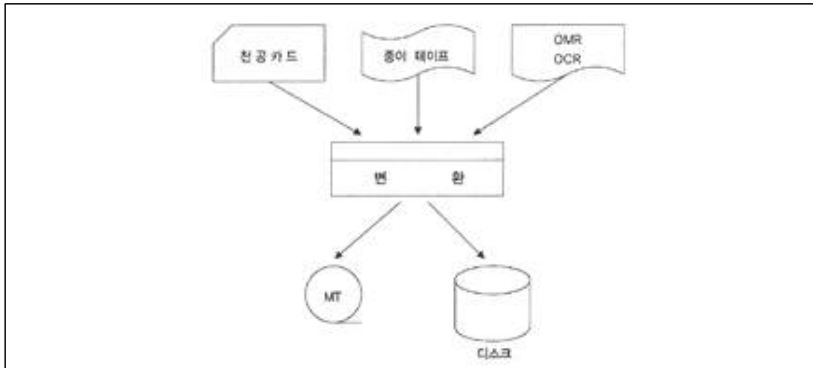
- 일괄처리 시스템
- 온라인 시스템

2) 일괄처리 시스템(Batch Processing System)

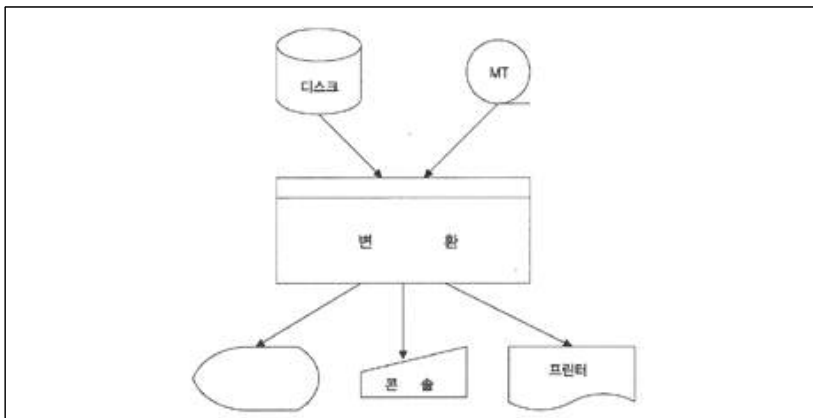
- 대상업무의 종류 · 내용은 다양하지만 「프로세스 설계과정」을 정형화하는 것이 가능함
- 프로세스의 기본적인 패턴(Pattern)은 다음과 같음

① 변환 (Conversion)

- 컴퓨터 업무처리의 가장 기본적인 처리임
- 입력변환 : 데이터 기록매체에 기록된 내용을 파일매체에 기억시키는 처리를 의미함
(교재「P. 369」의「그림 11-46」 참조)



- 출력변환 : 파일매체에 기록된 데이터를 출력 기록매체로 변환 하는 처리를 의미함
(교재「P. 369」의「그림 11-47」 참조)

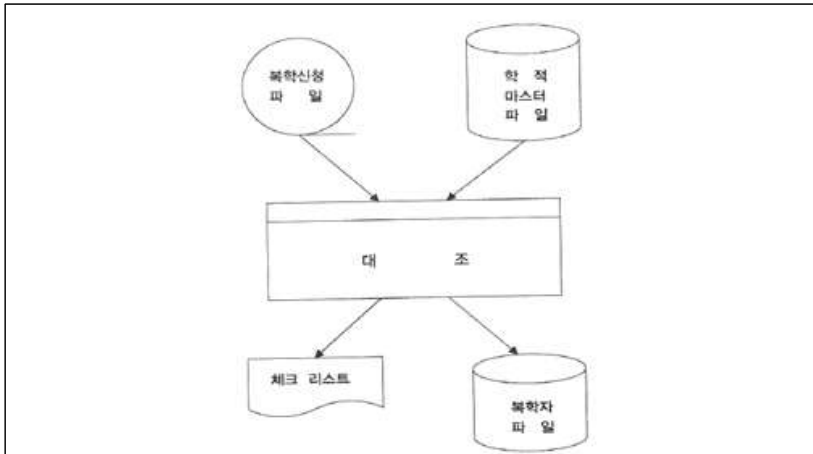


② 분류 (Sort : 정렬)

- 오름차순 분류(Ascending Order Sort)
- 내림차순 분류(Descending Order Sort)

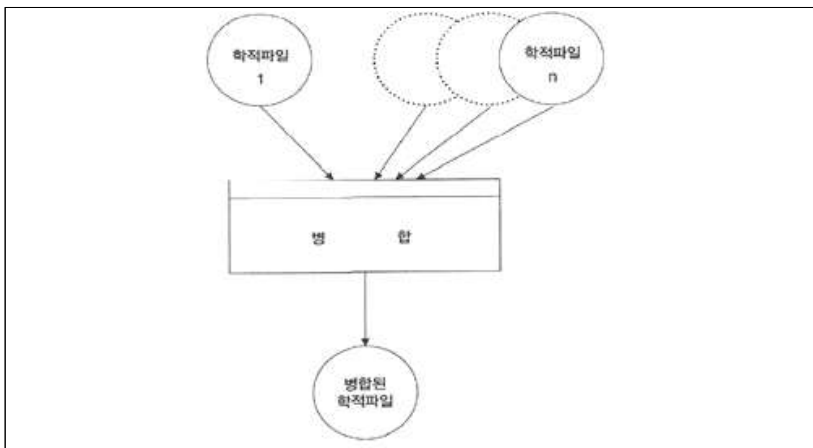
③ 대조 (Matching)

- 두 개 이상의 파일을 비교하면서 기록 순서나 내용을 검사하는 처리를 의미함
- 비교결과 이상이 발견되면 체크 리스트(check list)에 출력함
- 교재 「P. 371」의 「그림 11-49」 참조



④ 병합 (Merge)

- 동일한 형식의 두 개 이상의 파일을 처리 효율 · 보관 등을 감안하여 하나의 파일로 통합하는 처리를 의미함
- 통합 대상 파일들은 동일한 논리로 분류(sort) 되어 있어야 함
- 병합방법 : 오름차순 병합 혹은 내림차순 병합이 있음
- 교재 「P. 372」의 「그림 11-50」 참조



⑤ 갱신 (Update)

- 마스터 파일의 정보를 거래파일에 기록되어 있는 정보에 따라서 추가 (Addition) · 삭제(Delete) · 갱신(Update) 등의 처리를 하는 것을 의미함

⑥ 추출 (Extract)

- 파일에 기록된 정보 중에서 특정한 조건을 만족하는 정보만을 발췌하는 처리를 말함

⑦ 분배 (Distribution)

- 파일에 기록된 정보에 특정한 조건을 부여하고 첫째로 조건에 맞는 정보, 둘째로 조건을 만족시키지 못 하는 정보를 구분하는 처리임

⑧ 생성 (Generate)

- 1~n 개의 파일을 입력하여 가공 · 변환 · 처리하여 새로운 파일 을 작성하는 처리를 의미함

⑨ 보고서 (Reporting) 작성

- 필요한 처리를 한 다음에 인쇄용지나 기타 매체에 결과를 출력하는 처리를 의미함

2) 온라인 시스템(online system)

① 온라인 일괄처리 방식(Online Batch Processing)

- Online Data Collection System
- Remote Batch Processing System

② 즉시처리방식(Real Time Processing System)

- Online Real Time System
 - Inquiry
 - Transaction Data Processing
 - Message Switching
- Time Sharing System(시분할 방식)

3. 프로세스 설계 원칙과 고려사항

1) 기본적인 처리과정을 명확히 표현해야 함

- 입력에서 처리 과정까지

2) 업무의 흐름을 정확히 나타낸 「프로세스 차트(Process Chart)」를 작성하고 차트에 의거하여「논리설계(Logic Design)」를 해야 함

3) 프로세스 설계에서 발생 가능한「에러」에 대해서 여하히「체크시스템(check system)」을 도입할 것인지를 결정해야 함

4) 프로세스 설계 시

- 처리절차 · 정보흐름·처리과정 등을 이해하기 쉬운 「표준화 방안」을 고려해야 함

- 5) 하드웨어 구성 · 성능, 오퍼레이팅을 중심으로 소프트웨어 능력을 고려해야 함
- 6) 시스템 조작의 용이성, 오퍼레이터 개입의 최소화 방안을 고려해야 함
- 7) 프로그래머 · 관련 요원의 수준을 고려해야 함

4. 프로세스 설계과정

1) 요건 규명

- * 프로세스 설계에 요구되는 요건
 - 정보의 발생장소 · 발생빈도 · 발생시간
 - 정보의 수집 방법 · 수집시간
 - 정보의 사용장소 · 사용시간
 - 정보 사용자의 성격 · 사용자수
 - 처리내용 · 처리주기
 - 사용할 컴퓨터의 성능 · 구성

2) 처리방식의 확정

- 사용 컴퓨터 대수의 측면에서 집중처리 방식, 분산처리 방식
- 처리시점 측면에서 일괄처리, 즉시처리
- 통신의 측면에서 온라인 시스템, 오프라인 시스템

3) 프로세스 설계단계의 작업

- 시스템을 분할하여 「계층구조」로 작성해야 함
- 서브시스템의 「구비요건」 결정해야 함
- 각 서브시스템의 처리방식
 - 처리에 필요한 기기
 - 처리순서
 - 필요한 파일
- 서브시스템의 「입력 ⇨처리 ⇨출력」의 대응관계를 정의해야 함

〈서브시스템의 규모가 크면 「재분할」해야 하며, 환언하면「수행의 분할」을 함〉

- 단일 프로그램으로 요구사항을 구현하면 과부하가 걸리기 때문에 프로그램 작성이 어렵고 논리가 대단히 복잡해짐
- 처리할 데이터의 발생시점 차이로, 하나의 프로그램에서 구현이 불가능함
- 컴퓨터 시스템 구성상의 제약으로 하나의 프로그램으로 요구사항을 모두 구현하는 것이 불가능함
- 요구되는 처리사항과 데이터의 구성이 일치하지 않음

〈「서브시스템」 · 「수행」들은 「기본적 처리 패턴」으로 표준화시켜서 「논리적 순서」에 의거하여 전체적인 「프로세스 차트(Process Chart)」를 작성해야 함〉

【학습정리】

1. 파일의 종류 및 편성법을 학습한다.
2. 데이터의 기록형태 및 파일 설계 추진을 알아본다.
3. 프로세스 설계를 설명할 수 있다.