1 순차 자료구조의 문제점과 연결 자료구조의 개념

1 순차 자료구조의 특징

순차 자료구조 구현 방식은 논리적인 순서와 물리적인 순서가 같기 때문에 원소 위치를 찾아 액세스하기 쉬움

2 순차 자료구조의 문제점

- 삽입 연산이나 삭제 연산 후에 연속적인 물리 주소를 유지하기 위해서 원소들을 이동시키는 추가 작업과 시간 소요
  - 원소들의 이동 작업으로 인한 오버헤드로 원소의 개수가 많고 삽입·삭제 연산이 많이 발생하는 경우에 성능상의 문제 발생(작업시간이 더 많이 걸림)

2 순차 자료구조의 문제점

- 순차 자료구조는 배열을 이용해 구현하기 때문에
  배열이 갖고 있는 메모리 사용의 비효율성 문제를
  그대로 가짐(메모리 할당을 연속적으로 잡아야 함)

- 순차 자료구조에서의 연산 시간에 대한 문제와
  저장 공간에 대한 문제를 개선한 자료 표현 방법 필요

3 순차 자료구조 개선한 자료구조

- 순차 자료구조의 연산시간과 저장공간에 대한 문제를 개선한 자료표현 방법에는 연결 자료구조 또는 비순차 자료구조가 있음

4 연결 자료구조(Linked Data Structure)

- 자료의 논리적인 순서와 물리적인 순서가 불일치
  - 각 원소에 저장되어 있는 다음 원소의 주소에 의해 순서가 연결되는 방식
    - 물리적인 순서를 맞추기 위한 오버헤드가 발생하지 않음
  - 여러 개의 작은 공간을 연결하여 하나의 전체 자료구조를 표현
    - 크기 변경이 유연하고 더 효율적으로 메모리를 사용

5 연결 자료구조 - 연결 리스트

- 연결 리스트
  - 리스트를 연결 자료구조로 표현한 것을 연결 리스트라 함
  - 연결하는 방식에 따라 단순 연결 리스트와 원형 연결 리스트, 이중 연결 리스트, 이중 원형 연결 리스트가 있음

6 순차 자료구조와 연결 자료구조의 예

순차 자료구조의 예

- 한 장으로 만든 이불과 한 덩어리로 된 소시지처럼 하나로 된 고정크기의 메모리 공간을 사용함

[이불]

[소시지]

※ 출처 : IT CookBook, C로 배우는 쉬운 자료구조(개정3판), 이지영저, 한빛미디어

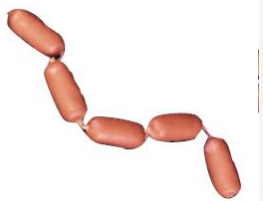
6 순차 자료구조와 연결 자료구조의 예

연결 자료구조의 예

- 퀼트 구조로 여러 개의 조각을 연결하여 만든 이불이나 작은 소시지를 여러 개 연결하여 만드는 줄줄이 소시지처럼 작은 공간을 여러 개 연결하여 전체를 표현하므로 유연하게 변경할 수 있고 메모리를 좀 더 효율적으로 사용할 수 있음

※ 출처 : IT CookBook, C로 배우는 쉬운 자료구조(개정3판), 이지영저, 한빛미디어

[퀼트 이불]



[줄줄이 소시지]

7 연결 자료구조 노드의 논리적 구조

- 연결 자료구조의 원소는 연결될 다음 원소에 대한 주소를 저장
- 노드(node) 구조

<원소, 주소>단위로 구성

7 연결 자료구조 노드의 논리적 구조

구성

데이터 필드(Data Field)	링크 필드(Link Field)
▪ 원소 값을 저장	▪ 다음 노드의 주소를 저장, 포인터 변수를 사용하여 주소값을 저장, 다른 말로 포인터 또는 링크 또는 참조(Reference)라고도 함

[노드의 논리적 구조]

8 순차 자료구조와 연결 자료구조의 비교

구분	순차 자료구조	연결 자료구조
메모리 저장 방식	필요한 전체 메모리 크기를 계산하여 할당하고, 할당된 메모리의 시작 위치부터 빈자리 없이 자료를 순서대로 연속하여 저장한다.	노드 단위로 메모리가 할당되며, 저장 위치의 순서와 상관없이 노드의 링크 필드에 다음 자료의 주소를 저장한다.
연산 특징	삽입·삭제 연산 후에도 빈자리 없이 자료가 순서대로 연속 저장되어, 변경된 논리적인 순서와 저장된 물리적인 순서가 일치한다.	삽입·삭제 연산 후 논리적인 순서가 변경되어도 링크 정보만 변경되고 물리적 위치는 변경되지 않는다.
프로그램 기법	배열을 이용한 구현	포인터를 이용한 구현

2 선형 리스트와 연결 리스트 비교

2 | 선형 리스트와 연결 리스트 비교

1 연결 리스트의 이해 : 기차놀이

규칙

- 각자 이름을 쓴 이름표를 만듦
- 상자 안에 전체 이름표와 X표를 담아 섞음
- 한 명씩 상자에서 이름표를 뽑고 이름표를 적힌 사람과 연결해 인간 기차를 만듦
- 인간기차는 이름표를 든 방향으로 움직이고, X표를 뽑은 사람은 기차 마지막 칸이 됨

2 | 선형 리스트와 연결 리스트 비교

1 연결 리스트의 이해 : 기차놀이

규칙

① 이름표 뽑기 ② 뽑은 사람을 찾아 연결 : 승희 → 상원 → 수영

※ 출처 : IT CookBook, C로 배우는 쉬운 자료구조(개정3판), 이지영저, 한빛미디어

2 연결 리스트의 이해 : 기차놀이와 연결 리스트

기차놀이	연결 리스트
기차놀이 하는 아이들 (승희, 상원, 수영)	연결 리스트의 노드
아이들이 가지고 있는 이름표	노드의 link 필드

3 선형 리스트 week을 순차 리스트와 연결 리스트로 표현한 경우 비교

- 리스트 week=(월, 화, 수, 목, 금, 토, 일)
  순차 리스트 표현
  - 배열 원소를 일곱 개 사용하여 연속적인 순차 구조를 메모리에 표현

※ 출처 : IT CookBook, C로 배우는 쉬운 자료구조(개정3판), 이지영저, 한빛미디어

3 선형 리스트 week을 순차 리스트와 연결 리스트로 표현한 경우 비교

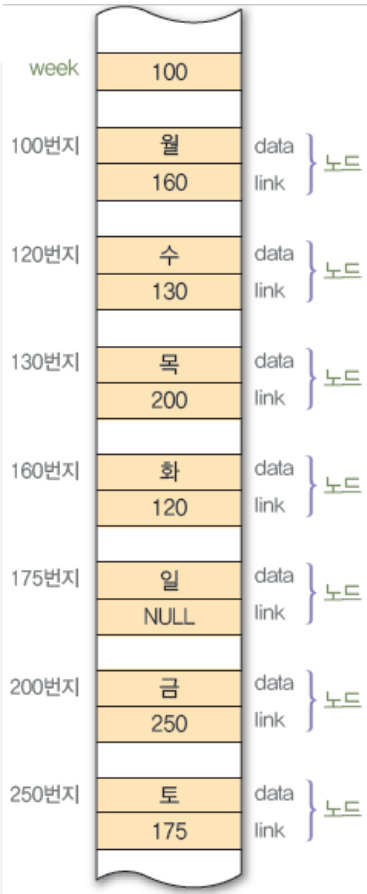
- 리스트 week=(월, 화, 수, 목, 금, 토, 일)의 **연결 리스트** 표현
  - 노드 일곱 개와 시작 포인터 week를 사용하여 비연속적인 연결 구조를 메모리에 표현

2 | 선형 리스트와 연결 리스트 비교

3 선형 리스트 week을 순차 리스트와 연결 리스트로 표현한 경우 비교

- 리스트 week=(월, 화, 수, 목, 금, 토, 일)의 **연결 리스트** 표현
  - 연결구조는 포인터를 담을 저장 공간이 추가로 필요한 대신, 삽입이나 삭제 연산을 하고 난 다음에 논리적 순서와 물리적 순서를 맞추기 위해 추가로 작업하지 않아도 됨

(a) 논리적 구조



(b) 물리적 구조

※ 출처 : IT CookBook, C로 배우는 쉬운 자료구조(개정3판), 이지영저, 한빛미디어

4 선형 리스트와 연결 리스트의 비교

- 리스트 이름 week
  : 연결 리스트의 시작을 가리키는 포인터 변수

  week는 연결 리스트의 첫 번째 노드를
  가리킴과 동시에 연결된 리스트 전체 의미

- 연결 리스트의 마지막 노드의 링크 필드
  : 노드 끝을 표시하기 위해 NULL(널) 저장

- 공백 연결 리스트
  : 포인터 변수 week에 NULL 저장(널 포인터)

5 연결리스트 week의 노드에 대한 구조체 정의

- 노드의 자료표현은 구조체로 정의
  - 구성은 값을 저장할 필드와 다음 노드를 연결할 링크 필드가 필요
  - 링크 필드는 연결을 표현하는 포인터이어야 하고 타입은 노드와 같은 구조체 타입으로 함

- typedef : 기존의 형태에 별명을 붙일 때에 사용
  - 형식에 대한 새로운 식별자를 정의할 수 있음
  - 새로운 유형 식별자를 정의한 후 기존 형식을 사용할 수 없게 되는 것은 아님

5 연결리스트 week의 노드에 대한 구조체 정의

- 구조체 타입에서도 typedef 사용 하여 타입을 간단하게 사용가능
  - typedef struct Node LinkedList; //typedef로 타입 재정의
  - struct Node node; → LinkedList node; //가능

5 연결리스트 week의 노드에 대한 구조체 정의

```
struct Node{
    char data[4];
    struct Node* link;
};
```

[리스트 week 노드의 구조체]

```
typedef struct Node{
    char data[4];
    struct Node* link;
}ListNode;
```

[typedef를 사용하여 ListNode 타입 정의]

6 연결 리스트의 노드와 점 연산자 표현

- 각 노드의 필드에 저장한 값은 점 연산자를 사용해 액세스

week.data	week.link
포인터 week가 가리키는 노드 데이터 필드 값 “월”	포인터 week가 가리키는 노드 링크 필드에 저장된 주소값 “120”

※ 출처 : IT CookBook, C로 배우는 쉬운 자료구조(개정3판), 이지영저, 한빛미디어

6 연결 리스트의 노드와 점 연산자 표현

- (a)의 대화에 담긴 진호와 상원이의 관계를 (b)의 연결 리스트의 노드로 표현하고, (c)와 같이 점 연산자(.)를 사용하여 액세스

(a)

(b)

(c) 진호.link.link.link.link.link.link.link.link.data

※ 출처 : IT CookBook, C로 배우는 쉬운 자료구조(개정3판), 이지영저, 한빛미디어

3 단순 연결리스트의 개념

3 | 단순 연결리스트의 개념

1 단순 연결 리스트(Singly linked list)의 개념

- 노드가 하나의 링크 필드에 의해서 다음 노드와 연결되는 구조를 가짐
- 연결 리스트, 선형 연결 리스트, 단순 연결 선형 리스트 라고도 함
- 앞에서의 기차놀이와 같이 이름표 하나를 가지고 한쪽으로만 연결된 인간기차가 단순 연결 리스트임

1 단순 연결 리스트(Singly linked list)의 개념

- 연결 리스트에서 각 요소는 노드(Node)로 이루어짐
- 노드는 구조체로 구현하며, 데이터와 다음 노드를 가리키는 포인터로 이루어져 있음
- 단순 연결 리스트는 이전 노드 안에 있는 포인터가 다음 노드를 가리키는 구조로 되어 있음
- 마지막 노드의 링크는 NULL을 가리키며 뒤로 돌아갈 수 없음
- 자주 변경되는 상황에서는 배열 리스트보단 연결 리스트가 비용적인 측면에서 효율적임