

## 제5장 연결 리스트

### 1. 리스트의 개념

- 리스트의 의미
  - '일정한 순서'의 나열
  - 어떤 정의에 의해서 결정된 '논리적인 순서'의 나열
- 리스트의 '순서'는 데이터가 저장되는 물리적인 위치와 상관없이 사람들의 머릿속에 인식되는 '논리적인 순서', 혹은 리스트에 나타나는 원소들 간의 '의미적인 순서'를 의미함
  - 배열은 인덱스로 표현되는 '순서'가 배열 원소의 메모리 공간(주기억장치, DDR)에서의 물리적인 위치를 의미함

### 2. 배열을 이용한 리스트의 구현

- '배열로 구현된 리스트'는 원소의 순서가 연속적인 물리적 주소에 저장됨
  - 원소를 삽입하거나 삭제하기 위해서는 해당 원소의 위치 뒤에 있는 모든 원소를 뒤로 물리거나 앞으로 당겨야만 됨
  - 리스트 원소값의 이동은 원소수가 많을수록 프로그램의 수행시간을 증가시킴
- 리스트의 원소 삽입은 프로그램의 실행 중에 메모리 할당을 필요로 하는 경우도 발생시킴
- 자료의 삽입과 삭제가 빈번히 발생하는 상황에서 리스트를 배열로 구현하는 것은 빈번한 자료 이동으로 인한 비효율적인 컴퓨팅 성능을 유발함

### 3. 포인터를 이용한 리스트의 구현

- 노드(node): 리스트의 원소(값) + 다음 원소를 가리키는 정보
  - 노드는 데이터 요소(값)와 리스트의 다음 원소를 지시하는 포인터(주소)로 구성됨

### 4. 포인터 변수

### 5. 연결 리스트에서 노드의 삽입과 삭제

- 리스트의 원소 삭제 연산 단계
  - ① 삭제할 노드의 선행 노드의 링크 필드를 삭제할 노드의 후행 노드를 가리키게 한다.
  - ② 삭제할 노드를 메모리에 반환한다.

○ 리스트의 원소 삽입 연산 단계

- ① 메모리 공간을 할당받고 삽입할 내용을 저장하여 삽입할 x노드를 생성합니다.
- ② x노드의 링크부분이 후행 노드가 될 j노드를 가리키게 합니다.
- ③ 삽입될 x노드의 선행 노드가 될 i 노드의 링크 필드가 x노드를 가리키게 합니다.

6. 연결 리스트의 여러 가지 연산 프로그램