

# **C Programming (CSE2035)**

## **(Chap12. Lists)**



---

**Sungwon Jung, Ph.D.**

**Bigdata Processing & DB LAB**  
**Dept. of Computer Science and Engineering**  
**Sogang University**  
**Seoul, Korea**  
**Tel: +82-2-705-8930**  
**Email : [jungsung@sogang.ac.kr](mailto:jungsung@sogang.ac.kr)**



## 실습 1

- 수업시간에 배운 linked list 구조를 이용하여, 학생정보를 list로 관리하는 프로그램을 작성한다.
  - 프로그램이 시작되면 메뉴가 주어진다.
  - 1번 메뉴는 데이터를 추가하는 메뉴로 학생정보를 / 로 구분해서 추가할 수 있다.
  - 학생 정보는 StudentID와 StudentName으로 구성되어 있다.
  - 2번 메뉴는 linked list의 원소를 모두 출력한다.
  - list의 nptr이 NULL인 경우 linked list의 마지막을 의미한다.
  - 1번 메뉴로 추가한 학생 정보는 순서대로 기존의 list 뒤에 추가하도록 한다.
  - 1번 메뉴를 통해 StudentID와 StudentName을 입력 받을 때마다, node를 동적으로 할당 받고, list에 추가하도록 한다.
  - 모든 출력은 main함수에서 한다.
  - StudentName은 30글자 이하로 가정한다. (정적 초기화를 해도 상관없음)
  - Node \*head 는 항상 list의 맨 앞을 가리키도록 한다.



## 실습 1

- void create\_node(int id, char \*name)
  - 새로운 node를 만들어 원래의 list에 추가한다.
  - Parameter: id = Student id, name = Student name을 가리키는 포인터
- void PrintList()
  - 모든 노드를 출력한다.
- 새로운 node와 Student name을 위한 공간은 항상 동적 할당을 한다.

```
void create_Node(int id, char *name) {
    Node *new;

    new = (Node *)malloc(sizeof(Node));
    new->st_name = (char *)malloc(sizeof(char)*10);

    // TODO
    // assign id to new->st_id
    // ...

    if (head == NULL) {
        /*TODO*/
    } else {
        /*TODO*/
    }
}
```

```
typedef struct _NODE {
    int id;
    char name[10];
    struct _NODE *link;
} NODE;

NODE *head;
```



## 실습 1

### ■ 프로그램 실행 예시)

[Select Menu]

1 Insert a new student

2 Print list

> 1

Enter a student: 20200001/홍길동

> 2

[20200001/홍길동] -> NULL

> 1

Enter a student: 20200002/둘리

> 2

[20200001/홍길동] -> [20200002/둘리] -> NULL



## 실습 2

- 강의 시간에 배운 linked list insertion을 실제로 구현해 보자. Node를 list의 원하는 위치에 삽입하는 방법을 배운다.
  - 데이터는 char형 변수 하나만 사용한다.

```
typedef struct _NODE {  
    char data;  
    struct _NODE *link;  
} NODE;  
  
NODE *head;
```

- 구현 함수 :

InsertNodeAt(char, int) / PrintList()



## 실습 2

- void InsertNodeAt(char a, int index)
  - 새로운 node를 생성하여 list에서 index에 삽입되도록 한다.
  - Parameter: a = 새로운 node의 data값
  - Parameter: index = 새로운 node를 삽입할 위치 값인 인덱스
    - head가 NULL인 경우 index의 값과 상관없이 0번째 위치에 node를 삽입한다.
    - list의 길이가 주어진 index보다 작을 경우 list의 맨 마지막에 위치에 node를 삽입한다.
    - list의 위치를 의미하는 인덱스는 0번 부터 시작해서 1씩 증가한다.
- void PrintList()
  - Linked list의 모든 값을 출력한다.
- 본 문제에서는 전역변수를 정의해서 사용한다.



## 실습 2

- Main함수를 아래와 같다.

```
int main() {  
    InsertNodeAt('p', 0)  
    InsertNodeAt('p', 0)  
    InsertNodeAt('e', 100)  
    InsertNodeAt('a', 0)  
    InsertNodeAt('l', 3)  
    PrintList();  
}
```

- 프로그램 수행 결과

```
PrintList(): apple
```

# 실습 2

- Insert operation은 아래와 같은 순서로 이루어진다.

