

과제 2. 연결 리스트

한국외국어대학교
고 석 훈

연결 리스트(Linked List)

- 연결 리스트(linked list)
 - 리스트를 연결 자료구조로 표현한 구조
- 노드(node)
 - 연결 자료구조에서 하나의 원소를 표현하기 위한 단위 구조
 - 데이터 필드와 링크 필드로 구성



```
typedef struct _Node {  
    char data[20];  
    struct _Node* link;  
} Node;
```

연결 리스트 생성 방법

```
#include <stdio.h>
#include <stdlib.h>

typedef struct _Node {
    char data[20];
    struct _Node* link;
} Node;

Node *newNode(char data[]) {
    Node *p = (Node*)malloc(sizeof(Node));
    strcpy(p->data, data);
    p->link = NULL;
    return p;
}

Node *addFirstNode(Node *list, Node *p) {
    p->link = list;
    return p;
}

void main() {
    Node *list = NULL;

    list = addFirstNode(list, newNode("Wed"));
    list = addFirstNode(list, newNode("Tue"));
    list = addFirstNode(list, newNode("Mon"));
}
```

과제 2-1. 리스트 출력

- printList 함수를 만들어 보자.

```
. . .  
  
void printList(Node *list) { . . . }  
  
void main() {  
    Node *list = NULL;  
  
    printList(list);  
    list = addFirstNode(list, newNode("Wed"));  
    printList(list);  
    list = addFirstNode(list, newNode("Tue"));  
    printList(list);  
    list = addFirstNode(list, newNode("Mon"));  
    printList(list);  
}
```

```
NULL list  
[Wed]  
[Tue]->[Wed]  
[Mon]->[Tue]->[Wed]
```

과제 2-2. 마지막 노드 추가

- addLastNode 함수를 만들어 보자.

. . .

```
Node *addLastNode(Node *list, Node *p) { . . . }
```

```
void main() {  
    Node *list = NULL;  
  
    printList(list);  
    list = addFirstNode(list, newNode("Wed"));  
    printList(list);  
    list = addFirstNode(list, newNode("Tue"));  
    printList(list);  
    list = addFirstNode(list, newNode("Mon"));  
    printList(list);  
  
    list = addLastNode(list, newNode("Thu"));  
    printList(list);  
    list = addLastNode(list, newNode("Fri"));  
    printList(list);  
}
```

```
NULL list  
[Wed]  
[Tue]->[Wed]  
[Mon]->[Tue]->[Wed]  
[Mon]->[Tue]->[Wed]->[Thu]  
[Mon]->[Tue]->[Wed]->[Thu]->[Fri]
```

리스트 제거, freeList

```
. . .  
  
void freeList(Node *list) {  
    Node *p;  
  
    while (list != NULL) {  
        p = list;  
        list = list->link;  
        free(p);  
    }  
}  
  
void main() {  
    Node *list = NULL;  
  
    printList(list);  
    list = addFirstNode(list, newNode("Wed"));  
    printList(list);  
    list = addFirstNode(list, newNode("Tue"));  
    printList(list);  
    list = addFirstNode(list, newNode("Mon"));  
    printList(list);  
  
    list = addLastNode(list, newNode("Thu"));  
    printList(list);  
    list = addLastNode(list, newNode("Fri"));  
    printList(list);  
  
    freeList(list);  
}
```

```
NULL list  
[Wed]  
[Tue]->[Wed]  
[Mon]->[Tue]->[Wed]  
[Mon]->[Tue]->[Wed]->[Thu]  
[Mon]->[Tue]->[Wed]->[Thu]->[Fri]
```

과제 제출 방법

- e-class에 소스코드를 제출한다.
 - 오로지 소스코드 파일 *.c만 제출한다.
 - 소스코드 첫 줄에 제출자의 학번, 이름을 주석으로 기재한다.
 - 소스코드에 설명이 필요한 부분에 주석을 기재한다.
 - ◆ 작성하는 함수의 코드 직전에 함수에 대해 설명
 - ◆ 함수 내부의 주요 기능 별로 구분하여 설명 후 코드 작성
- 제출 마감
 - 2019년 3월 24일 (일) 자정
 - 제출 시간 엄수!