

실습6

Structure & List 연습프로그램



The Index is like this.

INDEX

1. 단순 연결 리스트

- 실습 목적
- 실습 문제

단순 연결 리스트

실습 목적 및 단순 연결 리스트 이해

■ 실습 목적

- 연결 리스트에 대한 이해도를 높인다.
- 노드의 생성, 삽입, 삭제 등 연결 리스트에서 사용되는 기능들을 이해하고 활용한다.

■ 단순 연결 리스트

- 단순 연결 리스트에서 노드들은 하나의 링크 필드를 가집니다. 이 링크 필드를 이용하여 모든 노드들이 연결되어 있습니다.
- 노드 = 데이터 필드 + 링크 필드



- 데이터에는 저장하고 싶은 정보를 저장
- 링크에는 연결되는 노드의 주소를 저장

- 헤드 포인터와 노드



- 연결 리스트의 시작 노드의 주소를 저장
- 포인터 방식으로 주소를 이용하여 시작 노드에 접근

실습 목적 및 단순 연결 리스트 이해

■ 실습 목적

- 연결 리스트에 대한 이해도를 높인다.
- 노드의 생성, 삽입, 삭제 등 연결 리스트에서 사용되는 기능들을 이해하고 활용한다.

■ 단순 연결 리스트

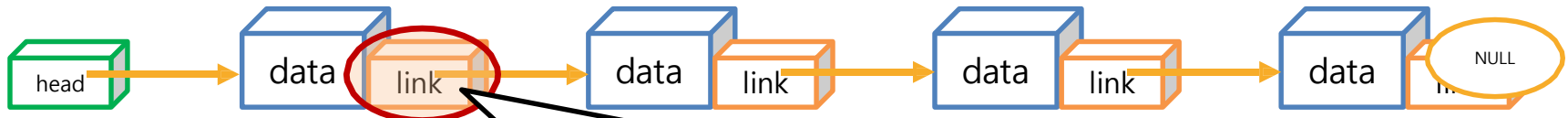
- 단순 연결 리스트에서 노드들은 하나의 링크 필드를 가집니다. 이 링크 필드를 이용하여 모든 노드들이 연결되어 있습니다.

- 노드 = 데이터 필드 + 링크 필드



- 데이터에는 저장하고 싶은 정보를 저장
- 링크에는 연결되는 노드의 주소를 저장

- 헤드 포인터와 노드



- 링크는 해당 노드와 연결되는 다음 노드의 주소를 저장
- 포인터 방식으로 주소를 이용하여 시작 노드에 접근

실습 문제 - 단순 연결 리스트 예제

- 실습 문제 : 리스트의 기능을 이해하고 사용해본다.
 - 노드를 생성하고, 생성된 노드를 리스트에 삽입한다.
 - 특정 노드를 리스트에서 삭제한다.
 - 리스트를 병합한다.
 - 리스트에서 특정 데이터를 탐색한다.
- 주어진 함수
 - `void insert_node(ListNode **phead, ListNode *p, ListNode *new_node);`
 - `void remove_node(ListNode **phead, ListNode *p, ListNode *removed);`
 - `void display(ListNode *head);`
 - `void display_recur(ListNode *head);`
 - `ListNode *search(ListNode *head, element x);`
 - `ListNode *concat(ListNode *head1, ListNode *head2);`
 - `ListNode *reverse(ListNode *head);`
 - `ListNode *create_node(element data, ListNode *link);`
 - 리스트 노드 구조체와 함수에 대한 설명은 강의노트 "Chapter6.List"를 참조하세요.

실습 문제 - 단순 연결 리스트 예제

- 실습 문제 : 리스트의 기능을 이해하고 사용해본다.
 - 노드를 생성하고, 생성된 노드를 리스트에 삽입한다.
 - 특정 노드를 리스트에서 삭제한다.
 - 리스트를 병합한다.
 - 리스트에서 특정 데이터를 탐색한다.

```
C:\WINDOWS\system32\cmd.exe
30->20->10->
20->10->
80->70->60->
80->70->60->20->10->
10->20->60->70->80->
탐색 성공: 20
계속하려면 아무 키나 누르십시오 . . .
```

- 리스트 함수를 적절히 활용하여, 주석에 맞게 동작하도록 코딩하세요.
- 결과는 위와 같도록 하세요.

```
int main()
{
    ListNode *list1=NULL, *list2=NULL;
    ListNode *p;

    // list1에 10,20,30 추가
    display(list1);

    // list1 에서 첫번째 노드 제거
    display(list1);

    // list2에 60, 70, 80 추가
    display(list2);

    // list2 = list2 + list1
    display(list2);

    // list2를 역순으로
    display(list2);

    // list1에서 20 탐색
    printf("탐색 성공: %d\n", p->data);

    return 0;
}
```

실습 문제 - 단순 연결 리스트 예제

- 실습 문제 : 리스트의 기능을 이해하고 사용해본다.
 - 노드를 생성하고, 생성된 노드를 리스트에 삽입한다.
 - 특정 노드를 리스트에서 삭제한다.
 - 리스트를 병합한다.
 - 리스트에서 특정 데이터를 탐색한다.

• insert_node 함수와 create_node 함수 활용

• remove_node 함수 이용

• insert_node 함수와 create_node 함수 활용

• concat 함수 이용

• reverse 함수 이용

• search 함수 이용

```
int main()  
{  
    ListNode *list1=NULL, *list2=NULL;  
    ListNode *p;  
  
    //list1에 10,20,30 추가  
    display(list1);  
  
    // list1 에서 첫번째 노드 제거  
    display(list1);  
  
    // list2에 60, 70, 80 추가  
    display(list2);  
  
    // list2 = list2 + list1  
    display(list2);  
  
    // list2를 역순으로  
    display(list2);  
  
    // list1에서 20 탐색  
    printf("탐색 성공: %d\n", p->data);  
  
    return 0;  
}
```


Thank You
Q&A