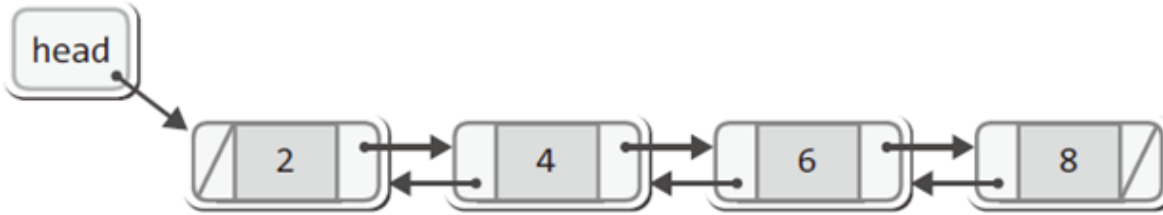


Chapter 05.

연결리스트3

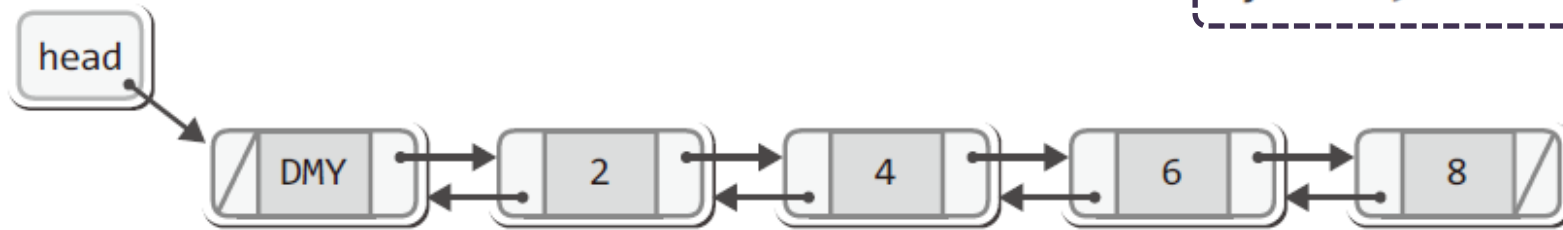
7주차

양방향연결리스트 (다양한 양방향 연결리스트)

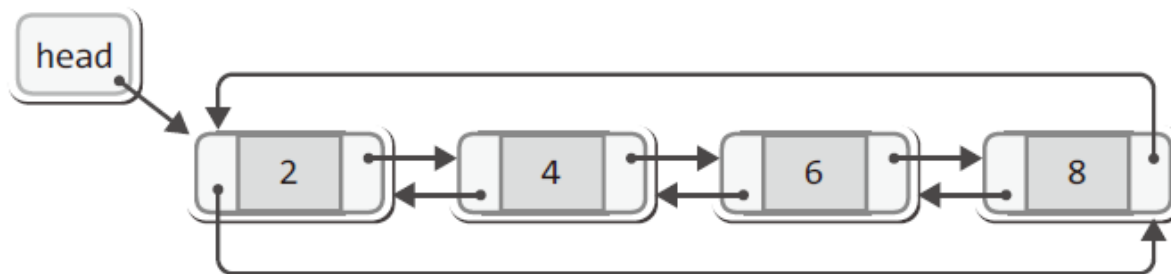


▶ [그림 05-17: 기본적인 양방향 연결 리스트]

```
typedef struct _node
{
    Data data;
    struct _node * next;
    struct _node * prev;
} Node;
```



▶ [그림 05-18: 더미 노드 양방향 연결 리스트]



▶ [그림 05-19: 원형 연결 기반의 양방향 연결 리스트]

노드구조체 :
링크부에 링크를 2개 사용
하여 **선행노드도 보관.**
-> **양방향으로 이동 가능.**

양방향연결리스트

(구조체 정의)

```
#define _CRT_SECURE_NO_WARNINGS
#include <stdio.h>

#define TRUE    1
#define FALSE   0

typedef struct _node {
    int data;
    struct _node * next;
    struct _node * prev;
} Node;

typedef struct _dbLinkedList {
    Node * head;
    Node * cur;
    int count;
} List;
```

양방향연결리스트

(리스트 초기화)

```
void ListInit(List * plist)
{
    plist->head = NULL;
    plist->count = 0;
}
```

양방향연결리스트 (머리에 추가)

```
void LInsert(List * plist, int data)
{
    Node * newNode = (Node*)malloc(sizeof(Node));
    newNode->data = data;

    newNode->next = plist->head;
    if (plist->head != NULL)
        plist->head->prev = newNode;

    newNode->prev = NULL;
    plist->head = newNode;

    plist->count++;
}
```

양방향연결리스트

(데이터 조회)

```
int LFirst(List * plist, int * pdata) {
    if (plist->head == NULL)
        return FALSE;

    plist->cur = plist->head;
    *pdata = plist->cur->data;

    return TRUE;
}

int LCount(List * plist)
{
    return plist->count;
}
```

양방향연결리스트 (데이터 조회)

```
int LNext(List * plist, int * pdata) {  
    if (plist->cur->next == NULL)  
        return FALSE;  
  
    plist->cur = plist->cur->next;  
    *pdata = plist->cur->data;  
  
    return TRUE;  
}
```

```
int LPrevious(List * plist, int * pdata) {  
    if (plist->cur->prev == NULL)  
        return FALSE;  
  
    plist->cur = plist->cur->prev;  
    *pdata = plist->cur->data;  
  
    return TRUE;  
}
```

양방향연결리스트 (main함수 실행 결과)

8 7 6 5 4 3 2 1 2 3 4 5 6 7 8

양방향연결리스트 (main함수)

```
int main(void) {
```

```
List list;  
int data;  
ListInit(&list);
```

양방향 연결 리스트의
생성 및 초기화

```
LInsert(&list, 1); LInsert(&list, 2);  
LInsert(&list, 3); LInsert(&list, 4);  
LInsert(&list, 5); LInsert(&list, 6);  
LInsert(&list, 7); LInsert(&list, 8);
```

8개의 데이터 저장

```
if (LFirst(&list, &data)) {  
    printf("%d ", data);  
    while (LNext(&list, &data))  
        printf("%d ", data);  
    while (LPrevious(&list, &data))  
        printf("%d ", data);  
    printf("\n\n");  
}  
return 0;
```

저장된 데이터의 조회

```
}
```