

# 운영체제 Lab04

2015140124

전자공학과

진우빈

## 1. 프로그램의 구조

```
#include "DoublyLinkedList.h"
#include "DoublyIterator.h"
#include "LinkedQueue.h"
#include "LinkedQueueIterator.h"
#include <conio.h>
#include <Windows.h>
using namespace std;

struct Block {
    int num; // block 번호
    bool free = false; // block 상태
    bool delay = false; // block이 delay 상태인지 여부

    Block() {
        num = 0;
    }

    Block(int n) {
        num = n;
    }

    string Display() {};
    bool operator==(const Block& data) {};
};

int HeaderSize;
LinkedQueue<Block>* Hashqueue;
DoublyLinkedList<Block> FreeList;
```

Buffer Cache실습을 위해 사용되는 Block의 구조체는 위와 같습니다. Block의 번호를 알기위해 Block의 번호와, Block이 현재 어떤 상태인지 알려주는 free, delay된 상태인지를 알려주는 delay boolean을 통해 구조체의 상태를 표현하였습니다. 또한 Hashqueue에는 해당 block의 번호가 나머지에 맞는 수

에 따라서 들어갈 수 있게 하였고 LinkedQueue 구조체를 사용하였습니다. FreeList는 Doublylinkedlist로 구현하였습니다. LinkedQueue와 Doublylinkedlist 두 ADT 모두 자료구조 시간에 설계한 ADT입니다. Queue에 별도의 iterator를 두어 index에 접근할 수 있게 하였습니다.

## 2. 사용 함수

`void Init()` // 초기 입력을 받는 함수

`void Display()` // 해쉬 대기행렬 헤더와 자유리스트의 상태를 출력하는 함수

`void ChangetoDelayedWrite(int num)` // block의 상태를 delay\_write로 변환해주는 함수

`void ChangeState(int num)` // block의 state를 사용 중으로 변경하는 함수

`int getblk(int num)` // getblk가 작동하는 5가지 경우에 따라 block을 해방하는 함수

## 3. 실행 화면

**Pdf 1페이지에 있는 사진을 예시로 하여 진행 하였습니다.**

```
C:\Users\Wjin\source\repos\OS_Lab04\Debug\OS_Lab04.exe
Hash Queue Header 사이즈를 입력해주세요 : 4

No 0 Queue의 block의 갯수를 입력해주세요 : 3
No 0 Queue에 할당될 block 번호를 입력해주세요.
1번 째 block 번호 : 28
2번 째 block 번호 : 4
3번 째 block 번호 : 64

No 1 Queue의 block의 갯수를 입력해주세요 : 3
No 1 Queue에 할당될 block 번호를 입력해주세요.
1번 째 block 번호 : 17
2번 째 block 번호 : 5
3번 째 block 번호 : 97

No 2 Queue의 block의 갯수를 입력해주세요 : 3
No 2 Queue에 할당될 block 번호를 입력해주세요.
1번 째 block 번호 : 98
2번 째 block 번호 : 50
3번 째 block 번호 : 10

No 3 Queue의 block의 갯수를 입력해주세요 : 3
No 3 Queue에 할당될 block 번호를 입력해주세요.
1번 째 block 번호 : 3
2번 째 block 번호 : 35
3번 째 block 번호 : 99

FreeList에 할당할 block의 갯수를 입력해주세요 : 6
FreeList에 할당할 block의 번호 6개를 입력해주세요 : 3 5 4 28 97 10
```

<해쉬대기행렬 헤더와 자유리스트의 정보를 입력받는 Init함수>

```
blk no 0 mod 4 [ 64 4 28 ]
blk no 1 mod 4 [ 97 5 17 ]
blk no 2 mod 4 [ 10 50 98 ]
blk no 3 mod 4 [ 99 35 3 ]
Free List [3 5 4 28 97 10 ]
```

1. change block status to delayed write
2. getblk
3. exit

<해쉬대기행렬 헤더와 자유리스트의 정보를 출력하는 Display 함수>

(이 때 Queue의 정보를 출력할 때 0 mod 4를 예를 들면 28, 4, 64 순서가 아니라 64, 4, 28 순서로 출력합니다. 조교님의 과제 pdf와 비교했을 때 순서가 반대이지만 진행 상의 문제는 없었습니다)

```
Command를 입력해주세요 : 1
Delayed Write로 변경시킬 block 번호를 입력해주세요 : 64
blk no 0 mod 4 [ 64_delayed_write 4 28 ]
blk no 1 mod 4 [ 97 5 17 ]
blk no 2 mod 4 [ 10 50 98 ]
blk no 3 mod 4 [ 99 35 3 ]
Free List [3 5 4 28 97 10 ]
```

1. change block status to delayed write
2. getblk
3. exit

<해쉬대기행렬 헤더에 존재하는 block을 Delay Write 상태로 변경하는 함수>

```
Command를 입력해주세요 : 2
할당 받을 block 번호를 입력해주세요 : 3
Block 3을 할당 받았습니다.
blk no 0 mod 4 [ 64_delayed_write 4 28 ]
blk no 1 mod 4 [ 97 5 17 ]
blk no 2 mod 4 [ 10 50 98 ]
blk no 3 mod 4 [ 99 35 3 ]
Free List [5 4 28 97 10 ]
```

1. change block status to delayed write
2. getblk
3. exit

<Free List에 있는 Block 3를 할당받는 getblk 함수>

```
Command를 입력해주세요 : 2
할당 받을 block 번호를 입력해주세요 : 98
Sleep...0
Sleep...1
Sleep...2
```

```
Block 98을 할당 받았습니다.
blk no 0 mod 4 [ 64_delayed_write 4 28 ]
blk no 1 mod 4 [ 97 5 17 ]
blk no 2 mod 4 [ 10 50 98 ]
blk no 3 mod 4 [ 99 35 3 ]
Free List [5 4 28 97 10 ]
```

```
1. change block status to delayed write
2. getblk
3. exit
```

<Free List에 있지 않은 block 98을 3초 동안 Sleep후 할당받는 getblk 함수>

```
Command를 입력해주세요 : 1
Delayed Write로 변경시킬 block 번호를 입력해주세요 : 4
blk no 0 mod 4 [ 64 4_delayed_write 28 ]
blk no 1 mod 4 [ 97 5_delayed_write 17 ]
blk no 2 mod 4 [ 10 50 98 ]
blk no 3 mod 4 [ 99 35 3 ]
Free List [5_delayed_write 4_delayed_write 28 97 10 ]
```

```
1. change block status to delayed write
2. getblk
3. exit
```

<FreeList에 존재하는 block을 DelayWrite로 변경하는 함수>

Exe 실행 파일은 첨부가 되지 않아 소스코드를 첨부합니다. (Visual Studio2019에서 실행함)