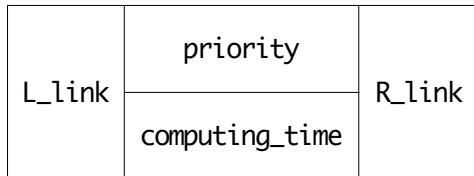


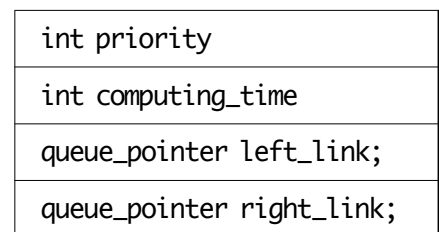
1. Design

1.1 Data Structure Modeling

Priority Queue Management 프로그램을 구현하기 위해 먼저 해야 할 일은 자료구조를 정의하는 것이다. 여기에서는 우선순위 큐가 둘 이상의 큐를 갖는 구조인 Multi-level Queue(MLQ)를 이용할 것이며, <Figure 1>과 같은 Data Structure를 큐의 노드로 정의한다. 이것은 논리적인 Doubly linked list의 구조이며, 메모리에 저장되는 물리적 구조는 <Figure 2>와 같다.



<Figure 1> Logical structure of queue element



<Figure 2> Physical structure

위에서 설계한 자료구조를 C언어로 표현하면 다음과 같다.

```
typedef struct priority_queue *queue_pointer;
struct priority_queue {
    int priority;
    int computing_time;
    queue_pointer left_link, right_link;
};

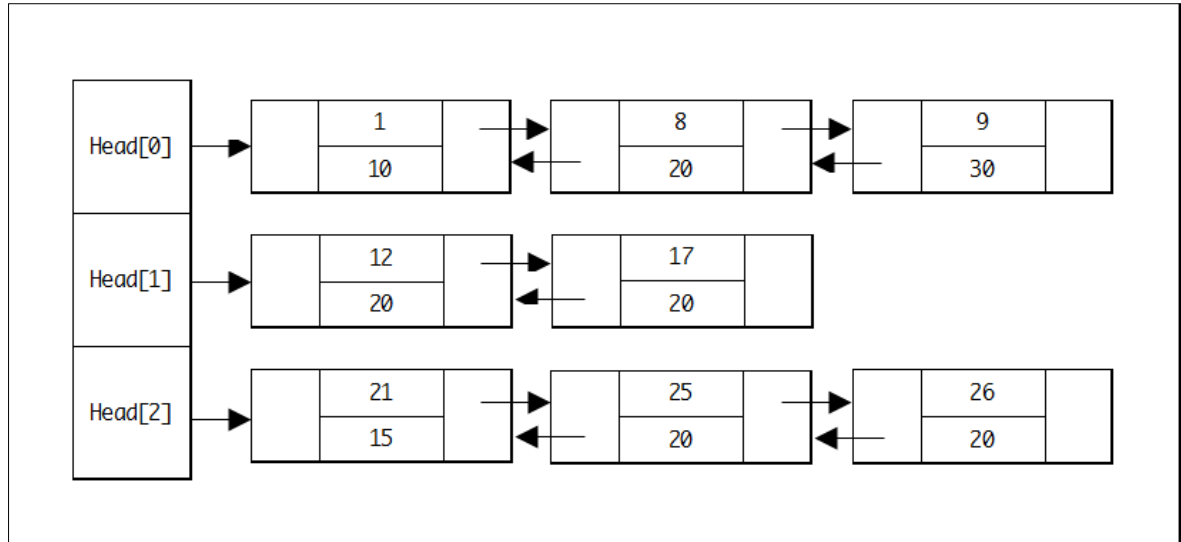
typedef struct queue_head *head_pointer;
struct queue_head {
    queue_pointer link;
};
```

<Figure 3> Definition of the structure for element of Queue in C

<Figure 3>에서 아래에 있는 구조체 queue_head는 큐의 노드를 가리키는 포인터 변수를 멤버로 갖는 구조체이다. 이것은 단지 큐를 가리키는 헤더의 용도로만 사용한다.

<Figure 4>는 3개의 레벨을 가진 priority queue의 예를 보여준다. 각 레벨의 큐는

priority 순으로 정렬되어 있으며 각 레벨의 priority 범위는 Head[0]은 1~10까지, Head[1]은 11~20까지, Head[2]는 21~30까지이다. 이 프로그램에서는 큐의 레벨을 3으로 고정한다.



<Figure 4> Example : Logical structure of a priority queue with 3-level

1.2 Operation Modeling

Priority Queue Management 프로그램에서 제공하는 Operation을 정의해 보면 여러 가지가 있겠지만 여기서 연산들은 다음과 같은 연산들로만 제한하고 정의하여 설계하기로 한다.

- initialize_queue() : 우선순위 큐를 초기화하는 함수로 head pointer를 NULL로 초기화한다.
- insert_queue() : Queue ID를 판단하여 적절한 큐에 삽입한다.
- delete_queue() : 해당 Queue ID를 판단하여 거기서 제일 앞 노드를 삭제한다.
또한, 제거되는 element의 정보를 출력한다.
- print_queue() : 우선순위 큐 안에 있는 모든 element를 출력한다.
- select_QueueID() : 입력으로 받는 priority가 어느 Queue ID에 속하는지 판단한다.
- destroy_Queue() : 큐를 만들기 위해 할당했던 메모리를 반납한다.

1.2.1 Definition of Priority Queue Operation

□ *initialize_queue*

형 식	void initialize_queue(void)
리턴값	없음.
설 명	head_pointer 타입의 변수를 NULL로 초기화한다. head_pointer 타입은 큐를 가리키는 포인터가 된다.
복잡도	큐의 레벨이 n 개일 때, $O(n)$

□ *insert_queue*

형 식	int insert_queue(int priority, int computing_time)
리턴값	성공시 0, 실패시 -1을 리턴
설 명	매개변수로 priority와 computing_time을 받아서 새로운 element를 생성한다. 내부적으로 Queue ID를 찾아서 적합한 레벨의 큐에 삽입하는 연산을 수행한다.
복잡도	큐의 element가 m 개이고, 레벨이 n 개일 때, $O(m+n)$

□ *delete_queue*

형 식	int delete_queue(int priority)
리턴값	성공시 0, 실패시 -1을 리턴
설 명	매개변수로 받은 priority값이 속한 레벨의 큐에서 최상위 우선순위를 갖는 element를 삭제하고, Queue ID, priority, computing_time을 출력한다.
복잡도	큐의 레벨이 n 개일 때, $O(n)$

□ *print_queue*

형 식	void print_queue(void)
리턴값	없음.
설 명	우선순위 큐의 레벨별로 모든 element를 출력한다.
복잡도	큐의 element가 m 개일 때, $O(m)$

□ *select_QueueID()*

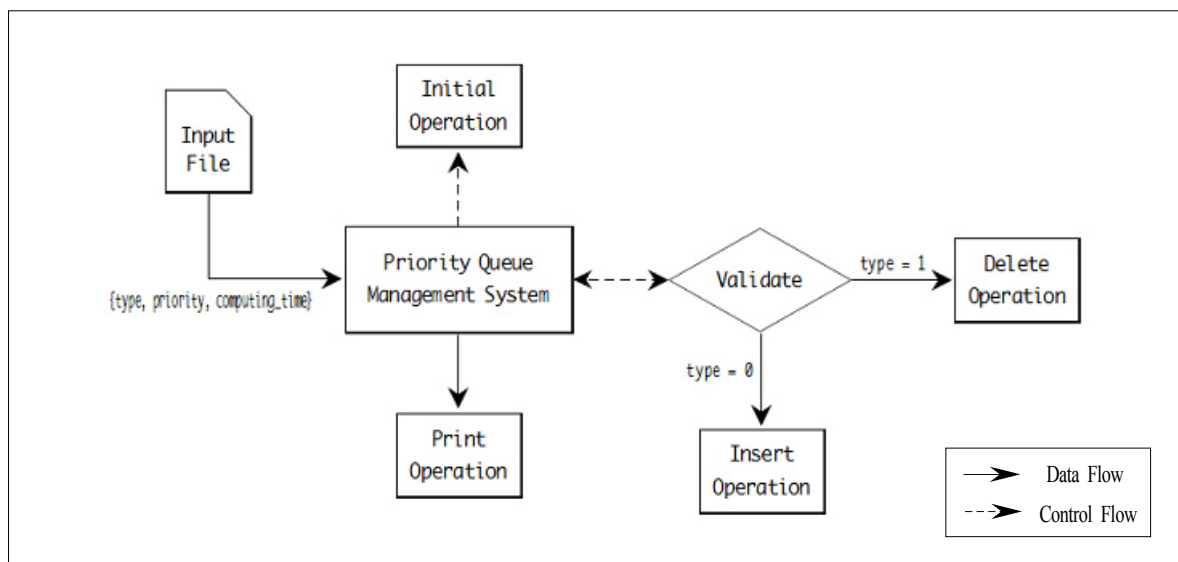
형 식	int select_QueueID(int priority)
리턴값	priority가 속한 큐의 level 또는 priority가 범위를 넘었을 때 -1 리턴
설 명	매개변수 priority에 해당하는 큐의 레벨을 선택하여 리턴한다.
복잡도	큐의 레벨이 n 개일 때, $O(n)$

□ *destroy_Queue()*

형 식 void *destroy_Queue*(int *queue_level_size*)
리턴값 없음.
설 명 매개변수로 큐의 레벨 수(int *queue_level_size*)를 넘겨받아 priority queue에 할당된 모든 메모리를 반환한다.
복잡도 큐의 element가 *m*개일 때, $O(m)$

1.3 Flow-Chart Modeling

Priority Queue Management 프로그램이 동작하는 순서도는 <Figure 5>에 묘사되어 있다. priority queue에 대한 입력이 파일로 주어지며 각각의 실선 화살표는 데이터 흐름을 나타내고 점선 화살표는 제어 흐름을 나타낸다. (각 독립 모듈(module)에서 제어 흐름은 Priority Queue Management System으로 반환된다고 가정하고 별도로 표시하지 않았다.)



<Figure 5> Flow-Chart of a Priority Queue management program

2. Execution

1.1 Contents of *input.txt* file

*input.txt*의 내용은 <Figure 5>와 같다. 예외처리에 대한 결과를 확인하기 위해 중간 중간에 타입이 3이거나 우선순위가 50인 원소 등 유효하지 않은 값을 포함한다. 각각의 유효하지 않은 값들의 처리는 <Figure 6>에서 확인할 수 있다.

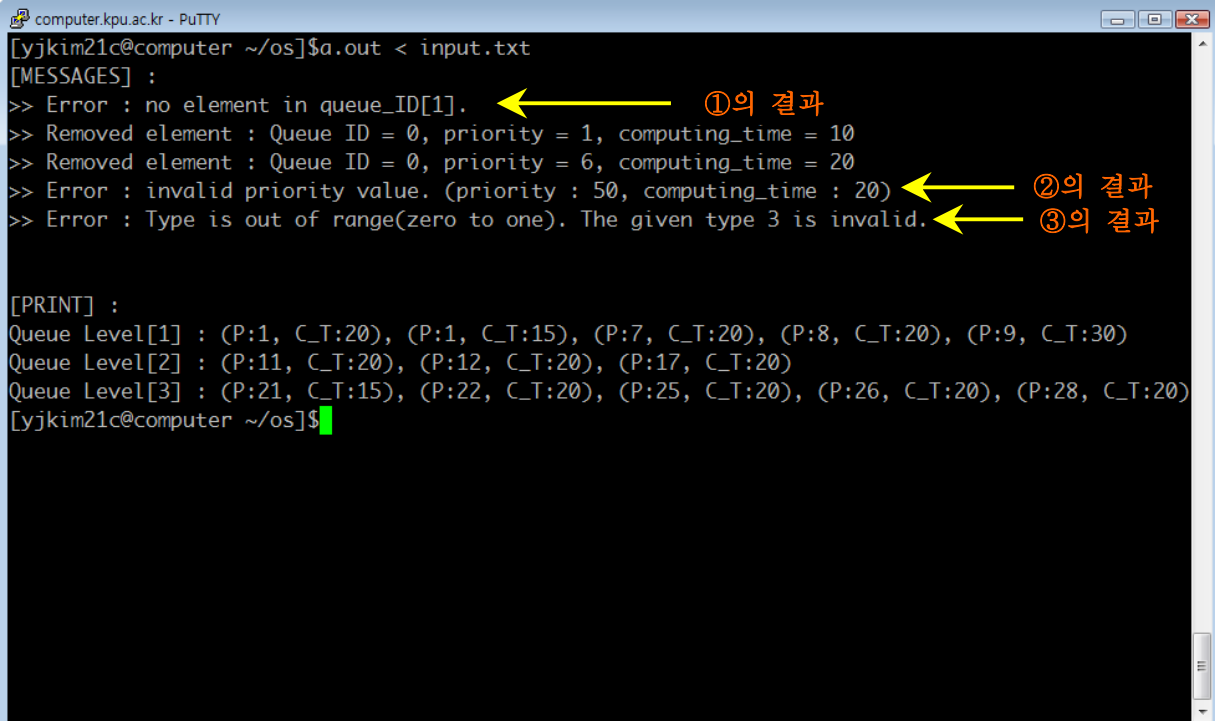
```
computer.kpu.ac.kr - PuTTY
0 8 20
1 11 15
0 1 10
0 11 20
0 17 20
0 12 20
0 9 30
0 6 20
1 10 15
1 10 15
0 26 20
0 22 20
0 28 20
0 25 20
0 7 20
0 21 15
0 50 20
0 1 15
0 1 20
3 4 30
~
~
~
"input.txt" 20 lines, 152 characters
```

<Figure 5> *input.txt*

1.2 Executed result

input.txt 파일의 내용을 입력으로 넣었을 때 <Figure 5>의 결과는 아래와 같이 출력된다. 여기서 유효하지 않은 값들에 대한 처리가 어떻게 되었는지 확인할 수 있다.

또한, 실행된 결과는 아래와 같이 MESSAGE 부분과 PRINT 부분으로 나뉘어 출력되며 MESSAGE에는 프로그램 실행 중 발생한 결과나 에러에 대한 결과를 출력하고 PRINT에는 우선순위 큐 안에 있는 모든 원소들을 출력한다.



```
computer.kpu.ac.kr - PuTTY
[yjkim21c@computer ~/os]$a.out < input.txt
[MESSAGES] :
>> Error : no element in queue_ID[1].
>> Removed element : Queue ID = 0, priority = 1, computing_time = 10
>> Removed element : Queue ID = 0, priority = 6, computing_time = 20
>> Error : invalid priority value. (priority : 50, computing_time : 20)
>> Error : Type is out of range(zero to one). The given type 3 is invalid.

[PRINT] :
Queue Level[1] : (P:1, C_T:20), (P:1, C_T:15), (P:7, C_T:20), (P:8, C_T:20), (P:9, C_T:30)
Queue Level[2] : (P:11, C_T:20), (P:12, C_T:20), (P:17, C_T:20)
Queue Level[3] : (P:21, C_T:15), (P:22, C_T:20), (P:25, C_T:20), (P:26, C_T:20), (P:28, C_T:20)
[yjkim21c@computer ~/os]$
```

<Figure 6> Result of "*a.out < input.txt*"

Priority Queue Management Program

- Design & Execution -



Dept. Computer Engineering

2005152011

YongJoong Kim