

운영체제 과제물1

2015년도 1학기

Due Date: 2015. 5. 6. 수업시간까지

1. 제목: Priority Queue Management 프로그램

2. 프로그램 개요

- 입력으로 priority 필드와 computing_time 필드를 받아 priority queue를 관리하는 프로그램을 작성한다.
- priority queue는 3개의 queue로 분리되며, priority 값이 1 - 10사이가 가장 높은 단계의 queue(Queue ID = 1)에서 관리된다. Priority 값이 11 - 20사이는 중간 단계의 queue(Queue ID = 2)에서 관리되며, priority 값이 21 - 30인 경우에는 가장 낮은 단계의 queue(Queue ID = 3)에서 관리된다.
- 각 queue에서는 priority 가 높을수록(숫자가 적을수록 priority 가 높다) head 가까이 에 위치된다.
- priority queue 초기화 function(initialize_queue())은 queue의 초기화 기능을 수행한다.
- priority queue 삽입 function(insert_queue())은 priority에 따라서 원소를 적절한 queue에 삽입한다.
- priority queue 삭제 function(delete_queue())은 해당 queue에서 priority 보다 높거나 같은 priority 값을 가지는 원소 중 최상위 priority 값을 가지는 원소를 삭제하며, Queue ID, priority 값과 computing_time을 출력한다. (삭제를 원하는 priority와 같은 원소를 삭제하는 것이 아님을 유의하기 바람)
- 입력은 input_file에서 읽어오며, {type, priority, computing_time} 필드를 입력으로 가진다. 이때 type 값이 0이면 insert, 1이면 delete를 의미한다. 그리고 type -1은 입력이 완료되었음을 의미한다.
- 입력 완료 이후에는 현재 priority queue의 3개 queue의 모든 원소들을 priority가 높은 것부터 출력하는 function(print_queue())을 이용하여 각 원소가 속한 Queue ID, priority 값과 computing_time을 출력한다.

3. 프로그램 구조

```
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;
```

```

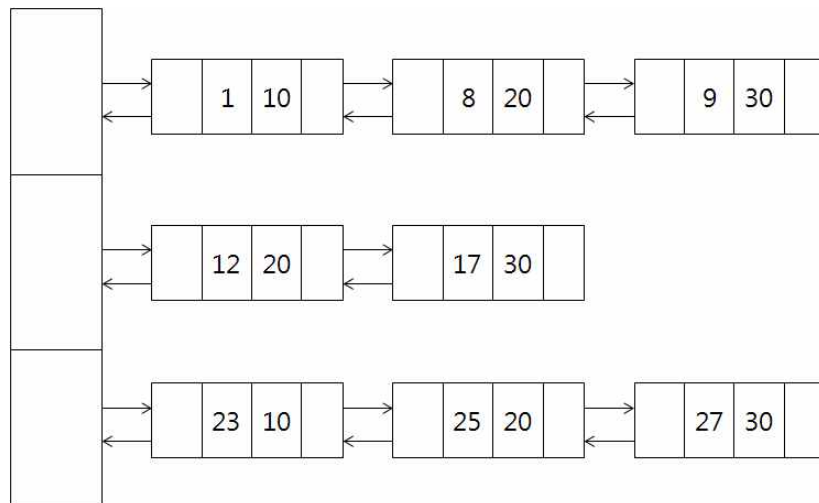
struct queue_head {
    queue_pointer left_link;
    queue_pointer right_link;
};

void initialize_queue(void);
int insert_queue(int priority, int computing_time);
int delete_queue(int priority);
void print_queue(void);

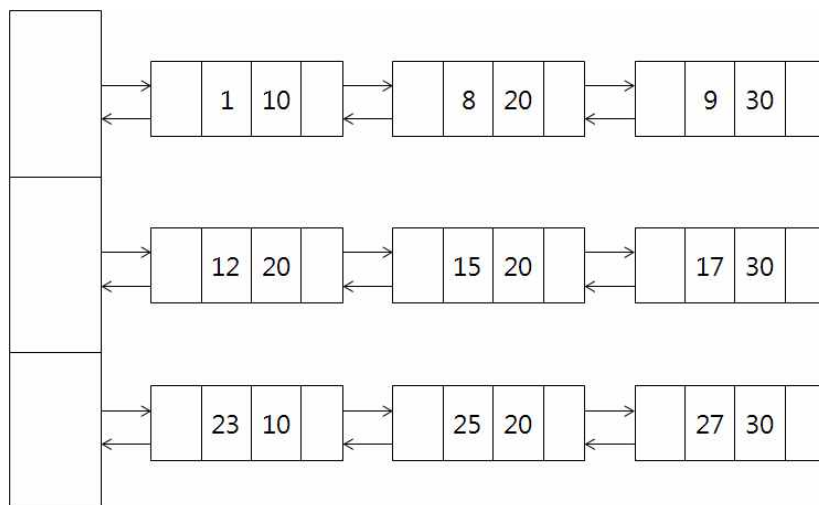
```

4. 프로그램 수행 예제

(1) 예제 구성

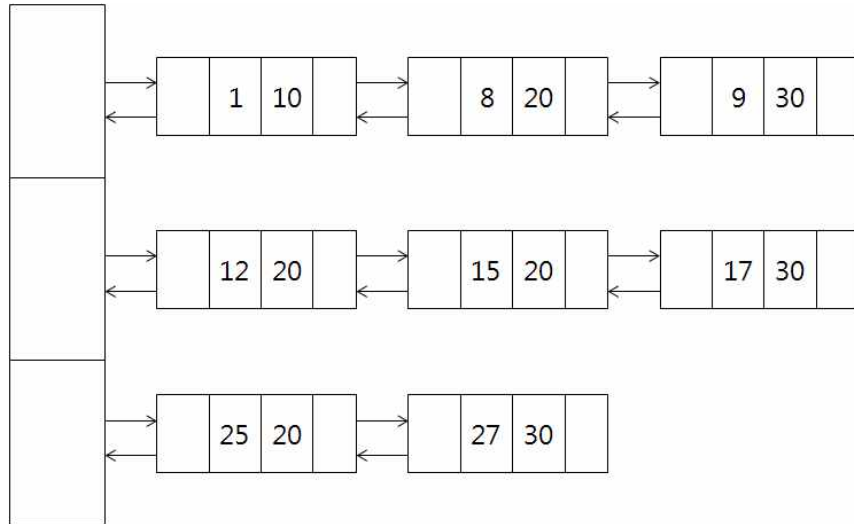


(2) priority = 15, computingtime = 20이 insert 되는 경우



(3) priority = 25가 delete 되는 경우

(실제로는 해당 Queue의 priority가 가장 높은 21을 가진 원소가 delete 된다)



5. 프로그램의 수행

- 입력은 현재 작업 디렉토리의 input_file로부터 가져오며, {type, priority, computing_time} 필드를 입력 받아 insert와 delete 기능을 수행한다. 최종적으로 print_queue()를 수행시켜 현재 큐에 남아 있는 모든 원소들의 priority와 computing_time을 출력한다.

(1) 입력 예제(type, priority, computing_time): input_file의 내용

```
0 8 20
0 1 10
0 23 10
0 25 20
0 17 30
0 9 30
0 12 20
0 27 30
0 15 20
1 25 20
0 28 10
1 9 30
1 9 30
0 2 20
1 15 20
-1
```

(2) 출력 예제

Queue ID	priority	computing_time
3	23	10
1	1	10
1	8	20
2	12	20
1	2	20
1	9	30
2	15	20
2	17	30
3	25	20
3	27	30
3	28	10

6. 관련된 프로그램

- C 자료 구조의 이중연결리스트만 잘 이해하면 쉽게 작성 가능한 프로그램입니다.

7. 과제물 제출 방법

- e-class 과제 게시판에 C 프로그램 소스 프로그램을 업로드 한다.
- 프로그램 이름은 OS학번.c로 한다. 즉 학번이 2015051130이면 OS2015051130.c라는 이름으로 업로드 한다.
- 프로그램은 UNIX 환경에서 gcc 컴파일러를 이용해서 컴파일되고, 테스트 가능해야 한다.

8. 기타 유의 사항

- 과제물1은 운영체제 설계과제물(Term Project)을 수행하기 위해 필수적인 내용입니다.
- 프로그래밍은 개인별로 수행합니다. 타 학생의 내용을 참고하지 말고 독자적으로 수행하기 바라며, 자신의 프로그램으로 작성하기 바랍니다.
- C 프로그래밍과 자료구조가 약한 학생들은 조교의 도움을 받기 바랍니다.
- 과제물 제출이 delay가 된 경우 감점합니다.