

Lebanese University
Faculty of Science
Section I

BS - Computer Science
2018-2019

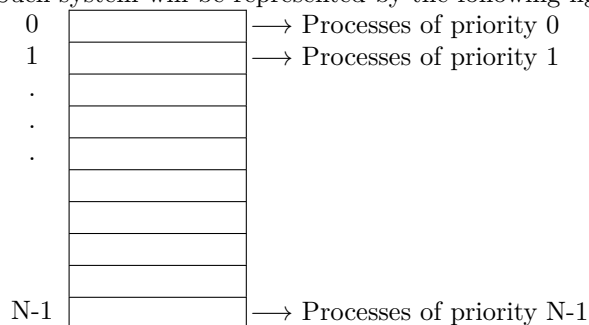
I2206

Data Structures

LS 5 : Priority Queue

We want to manage a set of processes in a computer system. Each process has a unique **id** and a **priority**. It is obvious that the processes having highest priority will be executed first. The processes with equal priority will be executed in FIFO. The maximum priority in the system is a constant N : The highest priority is therefore $N-1$, the lowest is 0. Assume that the execution of a process consists of printing his id.

Such system will be represented by the following figure :



First check `Type_Queue.h` file listed below :

```
#define N 5
#define M 10
#include<stdio.h>
#include<time.h>

typedef int element;
typedef element process;
typedef struct
{
    element data[M]; /* queue content */
    int front , rear;
} queue;

typedef queue systemQ[N];
```

Then, check `prog.c` and then write the following functions in `function.c` allowing to :

1. Create the previously described structure;
2. Display the content of the structure;
3. Add a process;
4. Delete the process with the highest priority;
5. Calculate the number of processes for a given priority;
6. Modify the priority of processes of priority i to priority j .

In order to test your functions, you can eventually generate random numbers (corresponding to the id or to a priority) by calling the function `int rand()` that returns random integer. Before calling this function, you should initialize the generator process of random numbers by calling `srand((unsigned) time(NULL))`.

EXAMPLE :

```
srand((unsigned) time(NULL))
/*Displaying 10 random numbers */
for(i=0;i<10;i++)
    printf("%6d\n",rand());
```