

Assignment 3 – Bo Henderson and Burde Prerana Kamath

Question 1

The doubly linked list was implemented with four functions. The FIFO operations which comprises of enqueue and dequeue was implemented followed by a priority enqueue function and a display function.

The structure consists of process id and the key value. The priority enqueue function was implemented using the key value as the priority. It also consisted of next and previous pointers that were local to every node and three global pointers that was current, front and rear.

Xinu uses arrays as the base structure whereas the program uses structures as the base structure. Since XINU uses relative pointers and has an implicit data structure such as the queuetab array, it takes less memory space and processor time. Since our implementation consists of structures using pointers there are some downfalls such as:

- 1 – Pointers always take up extra memory space.
- 2 – Since arrays have a better cache locality, they perform in a shorter time than our implementation and hence, their performance and memory requirements is better.
- 3 – Linked lists cannot be accessed randomly. As seen in the implementation, they need to be accessed using the front pointer always which doesn't hold true for arrays since they have already been assigned a particular amount of memory space since its static memory allocation and can be accessed through their indices.

Question 2:

In XINU, a valid queue id is defined in the queue.h file.

Valid queue id's are:

- It must be positive
- It must lie between 0 and NQENT – 2 (both inclusive)

$NQENT = NPROC + 4 + NSEM + NSEM$

NPROC – Number of processes.

NSEM – Number of semaphores.

These are hardcoded values in XINU.