

Highest Available Priority Queue

Jeff Klow - CSC311 Systems Software

Cornell College

03 December 2015

The Priority Array

```
QueuePointer priorityArray[PRIORITY_LEVELS];  
for( i = 0; i < PRIORITY_LEVELS; i++ ) {  
    priorityArray[i] = createQueue();  
}  
double elapsedTime = 0.0;  
for( i = 0; i < numberOfProcesses; i++ ) {  
    ProcessPointer pp = createProcess();  
    elapsedTime += pp->interarrivalTime;  
    pp->arrivalTime = elapsedTime;  
    enqueue( priorityArray[ (*pp).priority ], pp );  
}
```

The Algorithm

```
double elapsedTime = 0.0;
if( (*pp).arrivalTime >= elapsedTime ) {
    for( j = i; j < len; j++ ) {
        pp = peek( arr[j] );
        if( (*pp).arrivalTime > elapsedTime ) continue;
        else {
            pp = dequeue( arr[j] );
            elapsedTime += (*pp).serviceTime;
            enqueue( outputQ, pp ); break; }
    } // end low-priority for
    if( j = len-1 ) elapsedTime += WAIT_TIMER; }
```

Very Convenient

Within the source code:

```
#include "SimulationConstants.ini"
```

Within this file , there are the following variables named and explained:

```
#define MEAN_SERVICE_TIME 2.0  
#define MEAN_INTERARRIVAL_TIME 3.0  
#define PRIORITY_LEVELS 5  
#define PROCESS_COUNT 2000  
#define WAIT_TIMER 0.25
```

Wait Time Statistics

Using 5 priority levels and 2000 processes:

- ▶ The average wait time: 4.8707
- ▶ Average at priority 0: 1.2581 (26% of average)
- ▶ Average at priority 1: 2.1955 (45% of average)
- ▶ Average at priority 2: 2.7896 (57% of average)
- ▶ Average at priority 3: 5.8395 (120% of average)
- ▶ Average at priority 4: 11.9538 (245% of average)