

Spring 2020 Ins: Dr. Aftab Ahmad

Due by 4 PM on 4/06/2020

Implement an OS scheduler to compare average waiting times for priority queue versus non-priority queue.

Note: see the grading rubric at the end on the next page.

1. Imagine n jobs in an Operating System queue as an array A of integers, each element being a random number between 1 and 100. $A[i]$ is the size of job i in microseconds.
2. Copy the array A into an array B . Sort B using **quicksort**. Measure the time in microseconds that it takes to sort the array B . Divide this time by n . This is the sorting time per job.
3. Reverse-copy B into a third array C . In other words, $C[i] = B[n-1-i]$ for $i = \{0..n-1\}$. Now you have three arrays A , B and C . Array A has the jobs listed in the order of arrival, B listed in sorted order of the job size and C in reverse-sorted order.
4. The waiting time for job i is the sum of the sizes of all jobs $A[0]$ through $A[i-1]$. The queueing time of job i is the waiting time plus the job size $A[i]$. The average queueing time is the sum of queueing times of all the n jobs divided by n . Calculate the average queueing times for all the three cases; non-priority, priority (which is same as max-priority), and reverse-priority (which is same as min-priority). For priority and reverse-priority queues (B and C) add the sorting time per job to the calculated average waiting/queueing time. We are assuming that reverse-copying time is negligible.
5. Find the average queueing times for the three cases for $n = 10$ through 1010 with a step of 100. Print the actual (not average) **waiting times** (not queueing times) of three jobs for all the three cases for element numbers $n/3$, $2n/3$ and n . Print the three values for A , B and C in an output file <LastNameFirstInitial_4.dat>. This is something similar to what we did in assignment # 1. This time, we are not using random indexes but deterministic indexes so that we can compare the three cases.
6. Use gnuplot to plot the average **queueing times** using lines of different colors and proper labels, such as max-priority, no-priority, min-priority. The plot title should be 'Average queueing time for priority queueing'. There should be a key for all the three plot titles. Copy this multiplot and the data from *.dat file above into a PDF file using same naming convention. Add your observation from (5) and from the plots in a few sentences (two sentences will be good enough).
7. Compress the three files <YourLastNameFirstInitial_4.cpp>, <YourLastNameFirstInitial_4.dat> and <YourLastNameFirstInitial_4.PDF> into one file <YourLastNameFirstInitial_4.zip> or something such and submit this compressed file as your assignment.

In this assignment, you are not required to use HEAP to implement the queues, but you are welcomed to do so.

Grading Rubric

<i>Running code with correct results in *.dat file</i>	<i>20</i>
<i>Comments explaining data structures and algorithms</i>	<i>10</i>
<i>Graph depicting the queueing times for the three cases</i>	<i>10</i>
<i>Correct explanation of results</i>	<i>5</i>
<i>Submission format followed correctly</i>	<i>5</i>