CS 3220-AM,BM,CM: Operating Systems

Assignment 2

Date Assigned: Apr 22nd, 2021 Date Due: Apr 25th, 2021

(**Total Score**: 32 points)

(**Questions**: 2)

Instructions:

Labels of all figures and tables are mentioned below the figures and tables respectively. Show all your working where required. Provide all numerical results with two digits of precision only.

Q 1. In a real-world application of a Bank ATM server that processes all customers' requests placed upon their respective ATM machines, provide names and quantities of all possible threads that might be needed in a process, where a process is associated with a single user using a single ATM machine. **(16 points)**

Q 2. In this problem you are to compare reading a file using a single-threaded file server and a multi-threaded file server. It takes 20 msec to get a request for work, dispatch it, and do the rest of the necessary processing, assuming that the data needed are in the block cache. If a disk operation is needed, as is the case one-third of the time, an additional 50 msec is required, during which time the thread sleeps. How many requests/sec can the server handle if it is single-threaded? If it is multi-threaded? **(16 points)**

A 1.

S#	Thread	Quantity
1	User Identification	30
2	Account Selection	30
3	Balance Inquiry	15
4	Cash Withdrawal	15
5	Bill Payment	5
6	Money Transfer	5
7	Money Deposit	3
8	PIN Change	2

A 2. If required data is acquired from Cache (two-third of total time):

Total time to read a file (get a request for work, dispatch it, and do the rest of the necessary processing)

$$= t_C = 20msec = 20 \times 10^{-3} seconds$$

If required data is acquired from Hard Disk (one-third of total time):

Total time to read a file (get a request for work, dispatch it, and do the rest of the necessary processing)

$$= t_D = 20msec + 50msec = 70msec = 70 \times 10^{-3} seconds$$

If the Server is single-threaded:

Average time to read a file (get a request for work, dispatch it, and do the rest of the necessary processing)

$$= \frac{2t_C}{3} + \frac{t_D}{3} = \frac{2 \times 20 \times 10^{-3} seconds}{3} + \frac{70 \times 10^{-3} seconds}{3}$$
$$= 13.33 \times 10^{-3} seconds \; + \; 23.33 \times 10^{-3} seconds = 36.66 \times 10^{-3} seconds$$
$$= 36.66msec$$

Total number of requests per second that the server can handle $= \frac{1}{36.66 \times 10^{-3} seconds} = 27.28 \approx 27$

If the Server is multi-threaded:

Average time to read a file (get a request for work, dispatch it, and do the rest of the necessary processing)

$$= min\ (t_C, t_D) = min\ (20 \times 10^{-3} seconds, 70 \times 10^{-3} seconds) = 20 \times 10^{-3} seconds = 20msec.$$

Since during the $50msec$ time slot, where the Thread waiting for data from Hard Disk would be sleeping, other Threads (either the Thread reading a file from Cache, or the Thread trying to read a file from Cache but not receiving it and ultimately going for the Hard Disk) would be keeping the CPU busy, and hence the I/O time would not be counted towards the total time that reading a file from Hard Disk will take. Hence, the Average time to read a file would be the time it takes to read a file from Cache only.

Total number of requests per second that the server can handle $= \frac{1}{20\times10^{-3}seconds} = 50$