**CS 305 001, Operating Systems**

**Homework 2 (50 points)**

**Name: Write your Name**

**Question 1 (5 points):** Describe briefly (within 8 to 10 sentences) in your own words**\*** how the states of a process are switched between new, ready, running, and waiting in the execution.

**The first state a process enters is the ‘new’ state. The new state is when a process is first generated. From here, the process switches to the ‘running’ state. The running state is when the process’s instructions are performed. Next, the process switches to the ‘waiting’ state. The waiting state is when the process is pausing for something to happen (like receiving input) before it can be completed. After the event occurs that satisfies the waiting state, the process switches to the ‘ready’ state. The ‘ready state’ is when the process is equipped to be sent to a processor to be executed. After this, the process is ‘terminated’, meaning that is has completed.**

**Question 2 (5 points):** Two different processes can communicate indirectly via mailboxes or ports. Describe briefly (within 8 to 10 sentences) in your own words**\*** how two processes can communicate with each other via ports.

**In order to communicate with each other, the processes must share a communication link. This can be done using sockets, where each process must have one socket. A socket is a point from which data is transmitted to/from a process. Sockets are denoted by the combination of an IP address and a port (port number). This type of communication is usually handled through a client-server system. The server monitors a particular port. It is awaiting/expecting a request from the client to access this port. When the server gets the request, it accepts the connection from the client socket and finalizes the link.**

**\*Do not copy from the book, read appropriate sections in the book, and write what you understood in your own sentences.**

**Question 3 (10 points):** Figures 1 and 2 represent how servers and clients communicate over a network.

Analyze and the code, and then answer the following two questions:

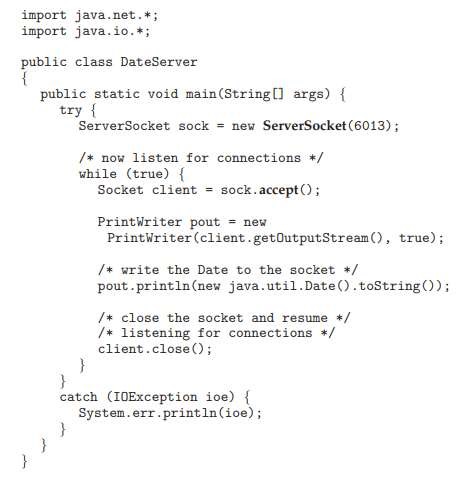


Figure 1 Server in Inter-process communication over network

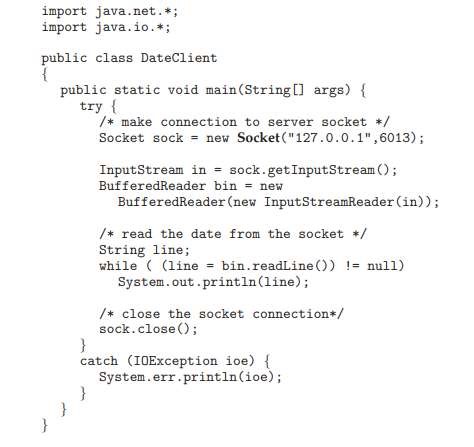


Figure 2: Client in Inter-process communication over network

1. **(5 points)** Describe briefly (5 to 10 sentences) how this server-client architecture is utilizing interprocess communication via port over a network. Refer to section 3.8.1, “Sockets” if you need to know more about inter-process communication over a network.

**The server creates a socket that monitors a certain port. The server awaits a request from the client to access this port. When the server gets the request, it accepts the connection from the client socket and the connection is complete. If processes are communicating on the same machine, the sockets will have the same IP address. If processes are communicating on different machines on same network, the sockets will have different IP addresses. Both processes are communicating with each other via a port, whether on the same machine or on a network.**

1. **(5 points)** In this architecture, the server only provides information to only one client. Describe briefly (5 to 10 sentences) how the server can introduce multi-threading ideas so that it can provide information/ service to multiple clients in parallel.

**The server should handle clients via multiple threads. This way, instead of having the server occupied with only the main thread, the server can service multiple client threads while the socket is listening for requests. The server will create a thread for each client as they arrive. After the server receives connection from the client it will provide service to the client via the created thread. The server returns to the main thread as it awaits another client.**

There are two different ways we design multithreaded programs, data parallelism, and task parallelism. Data parallelism distributes data across multiple threads so that each thread works on a different subset of the data but performs the same operation on the different input data. In contrast, task parallelism distributes tasks across different threads, each performing different operations on the same data. The following two questions (Question 4 and Question 5) involve implementing data parallelism and task parallelism in a multithreaded environment.

**Question 4 (15 points):** Write a multithreaded program (in Java/another programming language) to determine minimum, maximum, average, and standard deviation on a subset of the data items. There will be four different worker threads in this implementation. 1st worker thread will compute the minimum, maximum, average, and standard deviation, 2nd, 3rd, and 4th worker threads will do the same. But the list of data items they would work on would be different. Write the main function to generate a list of numbers, provide a subset of the list to each thread, invoke, and manage the worker threads. Each thread can print the minimum, maximum, average, and standard deviation values; they do not need to return it.

Example: if the primary InputList is [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12], and it will be divided into 4 smaller inputLists for each thread. So, thread#1 will work on the subInputList1 = [1, 2, 3], thread#2 will work on subInputList2 = [4, 5, 6], thread#3 will work on subInputList3 = [7, 8, 9], and thread#4 will work on subInputList4 = [10, 11, 12] and calculate minimum, maximum, average, and standard deviation respectively on it’s corresponding subInputList.

**This is labelled ‘CS 305 Homework Two question 4’ in zip file.**

**Question 5 (15 points)**: Write a multithreaded program (in Java/another programming language) to determine minimum, maximum, average, and standard deviation on a list of data items. There will be four different worker threads in this implementation. 1st worker thread will compute the maximum number, 2nd worker thread will compute the minimum number, 3rd worker thread will compute the average number, and the 4th worker thread will compute the standard deviation of the same list of data items. Write the main function to provide a list of numbers to threads, invoke, and manage the worker threads. Each thread can print the corresponding minimum, maximum, average, or standard deviation value; they do not need to return it.

For example: if the primary InputList is [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12], then all the 4 worker threads will work on the primary InputList, but thread#1 will find out the maximum, whereas thread#2 will find out the minimum, thread#3 will find out the average, and thread#4 will find out the standard deviation of the primary inputList respectively.

**Note:** Comment your code rigorously, if possible, line by line to ensure readers understand your intention of different actions and the purpose of each line/s in your implementation. Also, your code should be error-free so that I can run it and generate the output.

**This is labelled ‘CS 305 Homework Two question 5’ in zip file.**

**Point breakdown of questions 4 and 5:**

1. (8 points) Sound Logic, algorithm, and appropriate function calls for worker threads and main function.
2. (5 points) Executable code without any error and generates the correct output.
3. (2 points) Well-commented code.

**What to Submit:** Submit a zip file containing the following three items:

1. A text/ word/ pdf document for the answers to question 1, 2, and 3.
2. A java/ relevant file containing code for the implementation of question 4.
3. A java/ relevant file containing code for the implementation of question 5.