Name: _____

CS 6/73201 **Exam #1** **Advanced OS**

Wednesday 24 February 1999

1. Most computer systems discussed in this class are MIMD computers.

a. What is a "MIMD" computer? (Explain more than just the acronym.) (6 points)

MIMD is an acronym for "Multiple Instruction, Multiple Data". A MIMD computer has multiple processors, each of which can be doing different things (hence the "multiple instruction") on different pieces of data (hence the "multiple data").

b. Are MIMD computers parallel or distributed machines? Explain your answer. (10 points)

Both. If the MIMD computer has tightly coupled processors, sharing a single memory, it is referred to as a "multiprocessor" and considered to be a parallel machine. If it has loosely coupled processors, each with its own memory, it is referred to as a "multicomputer" and considered to be a distributed machine (the exception is when the processors are connected by a switched network; in this case it is a parallel machine).

2. The Internet Protocol (IP) is a very common protocol for communication at the network layer.

a. What are the main responsibilities of this protocol / layer? (7 points)

Providing routing needed to transfer data (packets) between systems. (For a connection-oriented protocol, the network layer would also be responsible for establishing, maintaining, and terminating the connection, but IP is connectionless, so this is not necessary.)

b. In what ways is IP "unreliable", and how is this unreliability overcome? (7 points)

IP is unreliable in the sense that packets can be lost, duplicated, or delivered out of order. It is the responsibility of the next higher level — the transport layer (perhaps using TCP) — to handle these errors in some way and reliably reassemble a whole message out of individual packets.

3. An important consideration in implementing message passing is the size of the buffer.

a. Is it possible to have a buffer of zero capacity, and if so, what are the advantages and disadvantages of this method? (6 points)

Yes, in this situation there is no buffer in the link, so the sender has to keep the message until it can be passed to the receiver. There aren't any real advantages to this method, other than perhaps a simple implementation, or the fact that the sender knows the receiver has gotten the message. The primary disadvantage is that it doesn't support non-blocking sends or asynchronous communication.

b. Is it possible to have a buffer with single-message capacity, and if so, what are the advantages and disadvantages of this method? (6 points)

Yes, and again the main advantage would be simplicity. There would also be a slight advantage in that it permits a non-blocking send (allowing the sender to send the message and then immediately proceed), but the corresponding disadvantage is that it permits only a single non-blocking send (before the buffer fills up, and subsequent sends have to block).

4. List, and briefly explain, some of the issues that are handled by an RPC mechanism, that would otherwise have to be handled by the programmer if message-passing were used instead. (15 points)

Your answer should explain how the RPC mechanism handles I/O protection, parameter marshalling, conversion between internal representations, error handling, binding to a server, maintaining state, etc.

5. What are the (i) advantages and (ii) disadvantages of user-level threads over kernel-level threads? (15 points)

Advantages: doesn't require OS modification, simple representation maintained by user process, simple thread management by user process, fast switching between threads since OS isn't involved, and very flexible.

Disadvantages: OS doesn't know about the threads, so it doesn't give the process more time, and if one thread blocks it blocks the entire process

(Note that I did not ask for the advantages and disadvantages of kernel-level threads. Some people wasted considerable space explaining this as well.)

6. What purpose does the ORB serve in a CORBA system? (8 points)

The ORB (object request broker) is responsible for distributing objects. It also acts as middleware, allowing software developers to write code that is portable between different OS and hardware vendors. Like RPC, it handles the issues addressed in question 4 on this exam.

7. Explain the difference between a symmetric multiprocessor (SMP) (an UMA machine) and a distributed shared memory (DSM) system, in terms of (i) overall architecture, (ii) access of each processor to memory, and (iii) caching. (20 points)

Architecture and access to memory: SMP has multiple processors connected by a bus, and each of those processors has hardware access to a globally-shared memory. DSM has multiple

processors connected via a network, each of those has its own memory, and processors can only access each other's memory through software (e.g., messages).

Caching: Each SMP processor has its own hardware cache, which must be kept consistent with the other caches (usually using "snooping"). Whether or not each processor in a DSM system caches data is a design issue for the operating system, and any caching is directly under OS (software) control.