Name: _____

CS 6/73201 **Final Exam** **Advanced OS**

Wednesday 10 May 2000

1. Client / server communication can be implemented using message-passing, but there are disadvantages to this implementation. How can client / server communication be improved when the items below are introduced?

a. Remote procedure calls (10 points)

RPC's free the programmer from having to be responsible for message formats, marshalling & unmarshalling parameters, data conversion, and most flow control and error handling.

b. Remote procedure calls and threads (10 points)

With RPC's and threads, a server can process requests from multiple clients concurrently, forking a new thread to service each client. Concurrency may also be improved on the client's side as well if the program can be multi-threaded in such as way as to be doing other work concurrently while talking to the server.

2. Write pseudo-code to "define" the semaphore P and V operations. (Note that this is a general question, not a Nachos question.) (15 points)

```
P(s):
s = s –1
if (s < 0)
        block the thread that called P
else continue into CS
```

```
V(s):
s = s + 1
if (s <= 0)
        wake up one of the waiting threads
```

or this alternate definition:

```
P(s):
if (s <= 0)
        block the thread that called P
s = s –1
continue into CS
```

```
V(s):
if (a thread is waiting)
        wake up one of the waiting threads
s = s + 1
```

3. Summarize Ricart and Agrawala's algorithm for mutual exclusion in a distributed environment. (20 points)

When a thread wants to enter the CS, it sends a timestamped request message to all threads in that CS's request set.

When a thread receives a request message, if it is neither requesting nor executing the CS, it returns a reply message. If it is requesting the CS, but the timestamp on the incoming request

is smaller than the timestamp on its own request, it returns a reply message. Otherwise, it defers answering the request.

A thread enters the CS when it has received a reply message from all other threads in the request set.

When a thread leaves the CS, it sends a reply message to all the deferred requests (thread with next earliest request will now received its last reply message and enter the CS).

4. Describe the general tradeoffs between deadlock detection and recovery, and deadlock prevention, in a distributed system. (15 points)

With deadlock detection, deadlock may occur, but when the deadlock detection runs later, some recovery mechanism will break the deadlock. With deadlock prevention, deadlock may never occur.

Deadlock detection can be applied in general, but the deadlock prevention mechanisms are usually very specific to particular circumstances or types of resources.

Many answers were acceptable here.

5. Explain how the private workspace and writeahead log are used in implementing atomic transactions. (15 points)

When a process begins a transaction, it gets a private workspace, containing copies of all files and objects that it needs. It then changes that private copy, and at the end of the transaction, a commit changes the originals while an abort leaves the originals untouched.

However, it is important to be able to survive a crash, so changes are recorded in a writeahead log (the transaction making the change, and the old and new values). For immediate update, operations are recorded in the log as described above, then the original data is immediately updated; if there is an abort the log is used to restore the original data. For deferred update, operations update the private data, and a commit updates the log and then the actual data as described above.

6. Under what circumstances, and why, can receiver-initiated load distribution algorithms become unstable? Are sender-initiated load distribution algorithms unstable under these same circumstances? Explain. (15 points)

Receiver-initiated algorithms really don't become unstable. At high system loads, there's a high probability that they will find a sender quickly, and at low systems loads, the polling may add some load to the system but it is highly unlikely that it would add enough to cause instability.

At high system loads, however, a sender-initiated algorithm can become unstable, as polling for a receiver is unlikely to be successful, which just adds to the system load, perhaps even enough to cause instability.

Name: _____

7. Describe write-through and write-back cache modification as they are used in distributed file systems, and the advantages and disadvantages of both approaches. (15 points)

With write-through the new value is immediately flushed through to the server. This keeps the server constantly up-to-date, but writing takes longer (losing the speed increase that caching usually provides).

With write-back the new value is flushed to the server after some delay. This keeps the speed increase, but means that if the server crashes before the flush some data may be lost.

8. Distinguish between symmetric and asymmetric cryptosystems. (15 points)

In symmetric cryptosystems the encryption and decryption keys are similar. Therefore the keys must be exchanged using a secure channel, but then encrypted messages can be exchanged using an insecure channel.

In asymmetric cryptosystems the encryption and decryption keys are different, and the encryption key need not be kept secret. Therefore both encryption keys and encrypted messages can be exchanged using an insecure channel.

9. Explain how a cluster is better suited for use as a web server than either a parallel machine or a big personal computer fully loaded with a lot of memory and disk space. (20 points)

Although a MIMD parallel machine might work as a web server, a cluster is generally cheaper than most parallel machines. Both would have the advantages of high reliability, load balancing, etc., but a cluster is much cheaper and may have specialized software dedicated for web-server-like tasks.

A distributed system may be scalable to more processors than a cluster (which is usually limited to 8, 16, or at most 32 processors), but the cluster probably has dedicated hardware and software for failover, giving a much higher reliability and fault tolerance, and thus making it a better choice.

A big PC isn't a very good solution at all. Its single processor severely limits its performance, and if it crashes, the whole site would go down.