

5 - Although both Amoeba and Sprite provide access to remote CPUs, only one provides process migration. Which one provides it, and what is the design philosophy that requires this?

6 - What are "finger successors" in Chord? How are they used?

7 - How does the LOCKSS system limit the **rate** of attacks that may be made upon a loyal peer?

8 - In what specific situation does the Time Warp Operating System send antimessages?

Long Answer (4 points each)

Answer **THREE** of the following **FOUR** questions.
If you answer all four, then only questions 1-3 will be graded.
A good answer can fit in the space provided.
However, use the back of the page if you need it.

(Go on to the next page.)

1 - The abstract Linda tuple space can be built in a variety of ways. The Linda paper that we read described two possible protocols for implementing Linda: the S/Net protocol and the Leichter protocol. How would both of these protocols implement the tuple space actions of the three processes described below? Give your answer in the form of a timeline listing all of the messages passed between each component.

Simplifying assumptions:

- Assume that processes A and B are started long before process C.
- Otherwise, you may assume any valid partial ordering of events.
- On the S/Net, use the first delete protocol.

Process A:

```
in(int x);  
g = G(x);  
out("G",g);
```

Process B:

```
in(int x);  
f = F(x);  
out("F",f);
```

Process C:

```
out(25);  
out(25);  
in("F", int f);  
in("G", int g);
```

2 - Both NFS and AFS strive to provide a filesystem with semantics very similar to those of Unix. However, they both fall short in subtle ways. Give ONE example of two processes interacting via the filesystem that yields different results in Unix, NFS, and AFS. Give the example in terms of Unix system calls such as open, read, write, and close. Explain how the results are different, and why the structure of Unix, NFS, and AFS makes this happen.

3 - Amoeba and xFS are similar in at least one respect: they both separate raw data storage and file system structures into distinct services. Because of this separation, they both have a garbage collection problem: how does the system identify what storage is no longer in use? Describe both the Amoeba and xFS solutions to this problem. Why did these systems take different approaches?

4 - The Timewarp paper mentioned briefly that it is necessary for the system to estimate a Global Virtual Time (GVT) in order to commit irreversible actions and collect garbage. However, it did not give an algorithm for doing so. Propose a **simple** but complete algorithm for estimating GVT and collecting garbage in a running Timewarp system. (Assume no hardware failures.) Discuss any tradeoffs between time, storage, and communication.