Qualifying Examination - Systems

Architecture

1.
   - A. Describe two different policies that are used to replace a line in a cache.
   - B. Describe some of the problems associated with caches in multi-processor designs. Are there solutions to these problems?
   - C. Calculate an estimate of the cache hit ratio R as a function of n and a cache size S (in words) and line length L (in words) in the program below. Assume an idealized cache that always makes the best line replacement choice. Assume that the size of a word is the size of a float.

```
float a[n];
register float x = 0;
while(1){
  for(register int j = 0; j < n; j++)
    x += a[j];
  }
```

2.
   - A. Describe the differences between CISC and RISC processor design.
   - B. Which is better for pipelined implementations? Why?
   - C. What does it mean when an instruction pipeline stalls? What are the techniques used to keep RISC pipelines from stalling?

Operating Systems

1. Explain the basic design of a paged virtual memory system. In particular, describe the steps the operating system must take when a page fault occurs. Also describe the hardware support, such as the TLB, that is often provided to assist paged virtual memory run faster.

2. Describe the FIFO page replacement algorithm. Describe the LRU algorithm. Given a machine with 8 pages and three page frames and a program with a page reference string of 0172327103, how many page faults will occur if the FIFO algorithm is used? (assume the pages are all empty at load time.) How many withe LRU?

3. Describe the differences between the access control list method of file system protection and the capability list method.

Parallel and Distributed Systems

1. What is the relationship between Data Parallelism and Single Instruction, Multiple Data (SIMD) architectures? What is Single Program Multiple Data programming and how does it relate to Data parallelism?

2. Two common ways for processes in distributed systems to interact are

- Remote Service Requests (e.g. "Active Messages", CORBA). The initiating process calls a function RSR which contains a processor and process identifier and a "pointer" to a service function to execute and a set of arguments to the function. The target process is interrupted when the message is received and the service function is executed with the supplied arguments. The result of the service function evaluation is returned to the "calling" process which has been suspended waiting for the reply.
- Typed Message Passing (e.g. MPI). The "sending" process issues a Send call which contains the processor and process identifier of both the sender and destination, the type (coded as an integer) of the message, the values and the size of the message. The receiving process issues a Recv call which provides the type of the message data to be received, the address of the buffer. The return value of the receive is the amount of data that was transfered to the buffer and the identity of the sender. If the Recv is executed before the Send, the receiving process will suspend. The Send will not return until the message has been delivered. There is also a function "Check" which returns the type value of the incoming message at the head of the queue, or 0 if there is no pending incoming messages.

Describe an implementation of message passing in terms of RSRs. You may assume the existence of semaphores to implement this.

Describe an implementation of RSRs in terms of message passing. (You may assume that there is a function ThreadCreate() which takes a pointer to a function and a pointer to an argument buffer and generates a new thread which will execute the function using the pointer to the argument buffer as an argument. You may also assume that client processes that are calling the Send and Recv functions are single threaded.)

Which implementation is the most efficient?