

Systems & Architecture Exam - Fall 2000

1 Systems

Problem 1.

- a. Any correct implementation of a critical section handler must satisfy three conditions: Mutual Exclusion, Progress and Bounded Waiting. Define these terms.
- b. Consider the following critical section implementation for two processes. (Assume one has process ID 1 and the other has process ID 0. Hence if you are process "i" the other process is process "j=1-i")

```
extern int turn; // turn is a shared variable
                // it is not cached.
int i = my_process_id(); // either 0 or 1
int j = i-1;
repeat
  // enter critical section handler
  while (turn != i ) do no-op;

  // enter critical section
  do_critical_section_work();

  // critical section exit
  turn = j;

  // outside of critical section
  do_outside_work();
until false
```

In which ways, if any, does this violate the three conditions above?

- c. How would you fix any problems you found in the the code above?

Problem 2. Consider a demand-paging virtual memory system with the following time-measured utilizations: the CPU is at 20% utilization, the paging disk is busy 97.7% of the time, other I/O devices are busy 5% of the time. Which, if any, of the following will (probably) improve CPU utilization?

- Install a faster CPU
- Install a bigger paging disk.
- Increase the degree of multiprogramming.
- Decrease the degree of multiprogramming.
- Install more main memory
- Install a faster hard disk, or multiple controllers with multiple disks

- Add prepagging to the page fetch algorithm.
- Increase the page size.

For each item that you feel is important explain why and rank order you list with the most likely to improve CPU utilization to those that are less likely. Do not list items that will, in your opinion, make the problem worse.

Problem 3. Describe two common methods used in file system implementations to keep track of free disk blocks. Each involve a data structure which may be kept in memory. What is involved in rebuilding a new version of this data structure in the event of a system crash? In other words, how does one build an accurate, new representation of free block information with a minimum loss of files and disk space?

2 Architecture

Problem 4. Analysis of 25 scientific programs shows that code in inner loops is executed 78% of the time. The loop code on average executes 200 machine instructions. Branches that terminate loops are predicted correctly 90% of the time.

As the pipeline designer of a new speculative architecture you have enough area on the VLSI CPU to put in 20 "blocks". Each block may hold one pipeline stage, or one logic unit to select one of two speculative computations, or one branch prediction unit. All blocks are designed to finish their work in one clock cycle that is 15 times faster than the clock cycle of an unpipelined machine.

- a. 10% Compute the performance improvement over an unpipelined machine if all 20 blocks are allocated to a single pipeline.
- b. 20% Compute the performance improvement over an unpipelined machine if 18 blocks are allocated to two nine-stage pipelines, one of which always executes the code following a branch, whether it is taken or not.
- c. 10% State any assumptions you need so as to allocate the 20 "blocks" in an optimal, or as close to optimal, CPU design to support the 25 scientific programs analyzed. Explain the limitations that this "optimal" design will have for other non-scientific programs such as word processing.

Problem 5. A computer has a small direct-mapped cache between main memory and the CPU. The cache has 64 16-bit words. Assume that only instructions are stored in the cache. Data operands are fetched directly from the main memory and not copied into the cache. Under what circumstances does this choice lead to faster execution than when both instructions and data are written into the cache? Explain.

Problem 6. RISC architectures have fixed length instructions, LOAD/STORE memory access instructions, and arithmetic and logic instructions that operate on registers only. List and explain five reasons why this architecture is superior to stack, memory-memory, and other CISC instruction sets.