Final Exam
CSE 451
Fall 2006
Version A

Name:

Short Answer (worth 4 points each)

1) Suppose (for some strange reason) you were asked to exhaust all the inodes on a system. How might you do this?

2) What is an advantage of a service-oriented architecture for distributed systems?

3) Larger page sizes tend to help TLB hit rates because the TLB maps a larger range of memory. However, this is not always the case. Can you describe a workload for which larger page sizes would decrease the TLB hit rate?

4) Why is a TLB necessary, given that all page translation information is stored in the page tables? Be sure to describe what would happen if the system did *not* use a TLB.

5) Round-robin is a commonly used strategy for CPU scheduling across a set of competing processes (or threads). Is round-robin a good strategy for scheduling the disks requests of a set of competing processes? Why or why not?

6) In the worst case, how many disk accesses are required to find the inode number for the following file:

```
/homes/andrew/super-secret/final-solution.pdf
```

You should make the following assumptions. First, all caches are initially empty. Second, once an item is referenced, it is cached forever. Third, all directories are small. Finally, the file `final-solution.pdf` is a hard link to the following file:

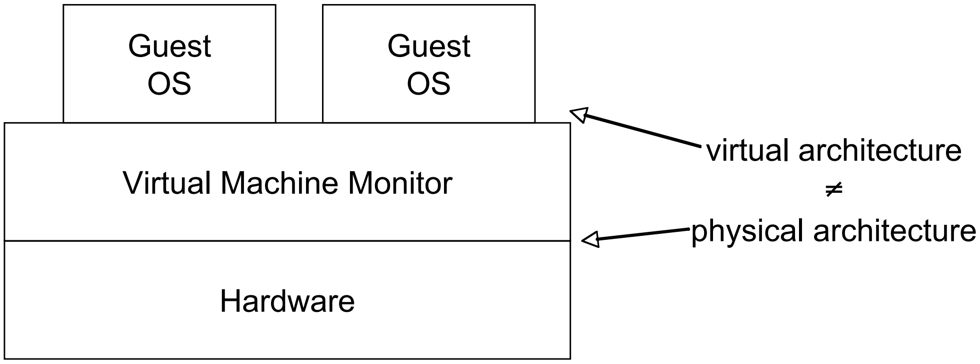
```
/homes/ak/fs.pdf
```

(To get partial credit, you should describe *what* disk accesses are required, not simply the total number).

7) Repeat problem 5, except that the link is symbolic instead of hard.

8) Suppose we have two identical computers, each running the Clock algorithm for page frame replacement. For computer A, the algorithm inspects an average of 25 page frames. For computer B, the algorithm inspects an average of 50 page frames. Which computer is closer to thrashing?

9) In class, we discussed how a virtual machine monitor supports multiple “guest” operating systems by efficiently cloning the underlying hardware architecture. In some cases, a VMM can (and does) expose a different virtual architecture than the underlying physical architecture:

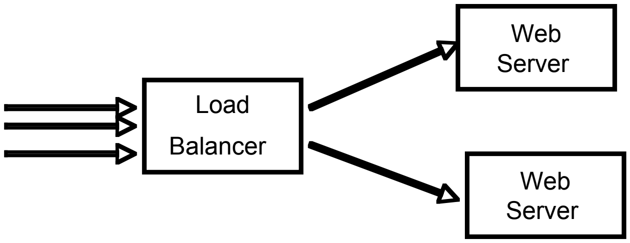


9a) [5 points] Is it possible for the VMM to expose a different Ethernet device in the virtual architecture? For example, could the VMM expose an Ethernet card made by 3Com when the physical machine’s Ethernet card is made by Intel? Explain your answer

9b) [5 points] Is it possible for the VMM to expose a virtual instruction set architecture that contains more registers than the underlying physical architecture? Explain your answer.

10) Consider a single web server that serves a large number of equally popular web pages. After benchmarking the web server, we learn that it can serve 100 req/sec.

Now suppose we add a second identical web server. A load balancer distributes traffic across these two servers.



10a) [6 points] After benchmarking the two web server configuration, we learn that it has a throughput of 10,000 requests per second for the same set of web pages. This is a 100X speedup over the single machine configuration. Can you explain how this is possible? Be sure to describe the behavior of the load balancer.

10b) [4 points] Suppose we further increased the server farm to include four web servers. Would you expect another 100X speedup? Why or why not?

11) [8 points] The C function `bzero` fills a memory buffer with zeroes:

```
// fill this memory buffer with zeroes
void bzero(void* buffer,long bufferSize);
```

Suppose you are asked to write an efficient `bzero` implementation. The programmer wants the ability to `bzero` large extents of memory. However, she does not know up front how much of the memory buffer will actually be used. In some cases, very little of the buffer will be used. How would you design such a `bzero` facility? You should provide enough detail so that a student outside this class could implement your design. You can use pseudocode, a detailed description, or a combination.

12) [10 points] In class, we looked at a `BoundedBuffer` class, which implements a fixed-size queue of Objects. When the queue is full, any attempt to add an element will block. Likewise, when the queue is empty, any attempt to take an element will block.

In the original implementation (shown below), the blocking operations are implemented with a condition variable. Your task is to convert this code to use one or more Semaphores instead of a condition variable. In other words, your solution should not use the `wait`, `notify`, or `notifyAll` methods.

```
public class BoundedBuffer  {
   private final Object [] buf;
   private int tail;
   private int head;
   private int count;

   public BoundedBuffer (int capacity) {
      this.buf =  new Object[capacity];
   }

   // Add an element to the queue, or block if the queue is full
   public synchronized void put(Object v) throws InterruptedException {
      while (isFull())
         wait();

      buf[tail] = v;
      if (++tail == buf.length)
         tail = 0;
      ++count;

      notifyAll();
   }

   // remove an element from the queue, or block if empty
   public synchronized Object take() throws InterruptedException {
      while (isEmpty())
         wait();

      Object v = buf[head];
      buf[head] = null;
      if (++head == buf.length)
         head = 0;
      --count;

      notifyAll();
      return v;
   }

   public synchronized boolean isFull() {
      return count == buf.length;
   }
   public synchronized boolean isEmpty() {
      return count == 0;
    }
}
```