Indiana University
Department of Computer Science
2005 Systems Qualifier

1. *Distributed file systems.* Network file systems (*e.g.* NFS v4.0) and the World Wide Web both provide users with access to remote files, yet the two systems have very different user interfaces. In network file systems, the user need only mount the remote file system onto the local machine. Then he or she can access remote files just as if they were local. In the WWW, a browser sends an address (i.e., a URL) of the file to a web server and displays the result to the user.

   a) List the major differences between the two systems' services. In particular, point out their differences with respect to:
      a. granularity of file access,
      b. semantics of the caching of remote files on local disks (including consistency),
      c. handling concurrent reads and writes of a file, and
      d. authorization and authentication.

   b) If NFS were universally available, how would that simplify the task of implementing the WWW?

2. *Threads and processes*. Suppose the following semantics of the "fork" call: the calling program invokes the "fork" command and passes it the name of a new executable.

    a. Explain, by means of a series of basic steps, the actions that take place, and the actors, between the execution of the 'Fork' command until the time the Wait(pid) returns. Give specific attention to the operations that the kernel must execute, the transitions between kernel space and user space, scheduling, and any synchronization operations that must take place.

    b. In some operating systems, user-invoked processes are actually implemented by kernel threads. Explain how the Fork command relates to kernel threads.

```
ProcessID pid = Fork("new_executable_path_name");
Wait(pid);
```

3. *Synchronization.* A doctor's office consists of a waiting room with *n* chairs and the examination room containing the examination table. If there are no customers to be served, the doctor goes to sleep. If a patient enters the doctor's office and all waiting room chairs are occupied, then the patient leaves the office. If the doctor is busy but chairs are available, then the patient sits in one of the free chairs. If the doctor is asleep, the patient wakes up the doctor. Write an algorithm that uses either POSIX threads or semaphores to coordinate the doctor and the patients.

4. *Mass Store.* In this question you will compare the cost and performance of two terabyte storage systems. The first, D, is made entirely from disks and the second, T, incorporates tertiary storage. Suppose that magnetic disks each hold 500 GB, cost $1,000, transfer 50 MB per second, and have an average access latency of 15 milliseconds. Suppose that a tape library costs $1 per gigabyte, transfers 100 MB per second, and has an average access latency of 20 seconds. In the questions below, you can use estimates in your comparison.

   a) Compute the total cost, the maximum total data rate, and the average waiting time for solution D. If you make any assumptions about the workload, describe and justify them.

   b) Now suppose that 5% of the data are frequently used, so they must reside on disk, but the other 95% are archived in the tape library. Further suppose that the disk system handles 95% of the requests, and the library handles the other 5%. What are the total cost, the maximum total data rate, and the average waiting time for this hierarchical storage system?

5. *Virtual Memory.* Physical memory is broken into fixed-size blocks called *frames*. Logical memory is broken into blocks of the same size called *pages*. Every address generated by the CPU has at least two parts: a page number (p) and a page offset (d). For question parts a) and b) consider a page size of 4 Kbytes, and a page table entry that takes 4 bytes. Assume virtual addresses are the same number of bits as the physical addresses.

   a) Assume a 32-bit address space. How much memory would be needed to hold the page table if a single-level page table was used?

   b) How many levels of page tables would be required to map a 64-bit address space, if the top level page table fits into a single page? What are the problems with this solution?

   c) For a 64-bit architecture, hierarchical page tables are generally considered inappropriate. The inverted page table is one solution, and is used in the 64-bit UltraSPARC and PowerPC. Sketch the operation of an inverted page table assuming each virtual address in the system consists of a triple <process-id, page-number, offset>. Discuss solutions (hardware or software) to reduce the additional costs inherent in an inverted page table solution.

6. *Security.* a) Explain how public key encryption and symmetric key encryption schemes work. What are the main differences between the two? Give an example each of the scenarios where one scheme would be preferred over the other. b) How do Unix and Linux store user passwords? What is the biggest drawback of this method?

7. *Scheduling*. You have a computer with a single processor, standard I/O subsystem, and hierarchical memory (e.g., a Pentium PC), running a modern preemptive multitasking operating system with a paged virtual memory subsystem (e.g., Linux). You have two programs to run on this computer, Batman and Robin. The elapsed real time (wall-clock time) for running either of these programs alone on your computer is exactly one minute. That is, when Batman runs alone, it takes one minute to complete and when Robin runs alone, it takes one minute to complete. Now, suppose Batman and Robin run on the computer at the same time, *e.g.*, they are launched simultaneously from two separate shells. We are concerned with the total real (wall clock) time it takes for both jobs to finish. Under what conditions (on the jobs, the OS, the computer), might the following occur:

   a) Batman and Robin are launched simultaneously and the last job to finish finishes after two minutes have elapsed (*i.e.*, at two minutes)

   b) Batman and Robin are launched simultaneously and the last job to finish finishes in less than two minutes (*e.g.*, at one minute 30 seconds)

   c) Batman and Robin are launched simultaneously and the last job to finish finishes in much more than two minutes (*e.g.*, four and a half minutes)

   d) Batman and Robin are launched simultaneously and the last job to finish finishes in one minute (*i.e.,* at one minute)