1. (5%) Describe the three-way handshake procedure for setting up a TCP connection.

2. (10%) Describe how the *Traceroute* program works.

3. (10%) Describe how network address translation (NAT) works.

4. (10%) Describe the operations of the address resolution protocol (ARP).

5. (6%) What are the requirements a solution to the critical-section problem must satisfy?

6. (6%) Consider a logical address space of eight pages of 1024 words each, mapped onto a physical memory of 32 frames.

    a) How many bits are there in the logical address?

    b) How many bits are there in the physical address?

7. (6%) What are the differences between a trap and an interrupt? What is the use of each function?

8. (5%) What are two differences between user-level threads and kernel-level threads? Under what circumstances is one type better than the other?

9. (6%) Researchers have suggested that, instead of having an access list associated with each file (specifying which users can access the file, and how), we should have a user control list associated with each user (specifying which files a user can access, and how). Discuss the relative merits of these two schemes.

10. (6%) What is the cause of thrashing? How does the system detect thrashing? Once it detects thrashing, what can the system do to eliminate this problem?

11. Assume that X consists of 3 bits, x2, x1, x0, and Y consists of 3 bits, y2, y1, y0. Write logic functions that are true if and only if

    a) (5%) X when interpreted as a signed (two's complement) number is less than -1

    b) (5%)X<Y, where X and Y are thought of as signed (two's complement) binary numbers

12. (10%) Here is a series of address reference given as word addresses: 1, 4, 8, 5, 20, 17, 19, 56, 9, 11, 4, 43, 5, 6, 9, 17.   Show the hits and misses and final cache contents for a direct-mapped cache with four-word blocks and a total size of 16 words.

13. (10%) The following program tries to copy words from the address in register $a0 to the address in register $a1, counting the number of words copied in register $v0. The program stops copying when it finds a word equals to zero. There is no need to preserve the contents of registers $v1, $a0. and $a1. The terminating word should be copied but not counted.

```
loop: lw    $v1, 0($a0)      # read next word from source
      addi  $v0, $v0, 1      # increment count words copied
      sw    $v1, 0($a0)      # write to destination
      addi  $a0, $a0, 4      # advance pointer to the next source
      addi  $a1, $a1, 4      # advance pointer to the next destination
      bne   $v1,$zero,loop   #loop if word copied is not zero
```

There are many bugs in this program. Fix them and turn in a bug-free version.