CS 502 Operating Systems WPI, Fall 2001
Craig E. Wills Final Exam (100 pts)
Given: Wednesday, December 12, 2001

NAME:___________________________________

This is a closed book (and notes) examination. Answer all questions on the exam itself. Take the number of points assigned to each problem and the amount of space provided for your answer as a measure of the length and difficulty of the expected solution. The exam totals 100 points.

SCORE:____________________________________

1. (7 points) Consider the following set of files with the first nine characters on each line showing read, write and execute permissions for the file owner, a group member and all other users. The permissions are followed by the file name, its owner and its group. The user `bob` is the only member of `bobgroup`. The user `grover` is the only member of `grovergroup`. Users `bob` and `jane` are members of `fungroup`.

```
rwxrwx---   foo         bob    fungroup
rw-rw----   foo.c       bob    fungroup
rwxrwxr-x   bar         bob    fungroup
rw-rw-r--   bar.c       bob    fungroup
rw-------   proj.doc    bob    bobgroup
rwxr-xr-x   func        jane   fungroup
rw-r--r--   func.c      jane   fungroup
rw-------   report.doc  grover grovergroup
```

Circle the correct answer for each of the following:

(a) Is `bob` allowed to read the file `proj.doc`? YES NO

(b) Is `jane` allowed to write the file `proj.doc`? YES NO

(c) Is `grover` allowed to execute the file `bar`? YES NO

(d) Is `jane` allowed to read the file `report.doc`? YES NO

(e) Is `bob` allowed to write the file `func.c`? YES NO

If `grover` wanted to give read and write access to `bob` for the file `report.doc`, what could `grover` do in the Unix file system? What could `grover` do in the Windows NT file system?

2. (13 points) The dining philosophers problem is a classic synchronization problem where each of five philosophers have a bowl of rice in front of them and a chopstick between each of them. A philosopher must pick up the chopstick to their left and right to eat.

   (a) Describe a resource allocation policy for chopstick management that can lead to deadlock.

   (b) Describe a relatively liberal resource allocation policy for chopstick management that does not lead to deadlock.

   (c) Rather than placing chopsticks between each philosopher, consider a new problem where the five chopsticks are placed in the center of the table. These chopsticks are shared amongst all philosophers with two chopsticks still needed by a philosopher to eat. Describe the most liberal resource allocation policy for chopstick management that does not lead to deadlock.

3. (18 points)

(a) A directory is a special file with entries for each file contained within the directory. Listing out the files in a directory is a common file system operation. In addition to listing each file, another operation is to also list the size and time stamp of each file. Where is this information stored for each file in each of the following file systems?

i. FAT

ii. NTFS

iii. Unix

(b) It has been observed that the contents of some files can be large. What must the file system software do to access the last byte of a large file in each of the following file systems?

i. FAT

ii. NTFS

iii. Unix

4. (7 points) Windows NT uses a working set manager for virtual memory management. What policies does this manager follow?

5. (7 points) What is watermark processing in terms of output handling and why is it used?

6. (8 points) What are cylinder groups and why were they added to the Berkeley Unix Fast File System? Give an example of how they are used.

7. (8 points) An inode in the Unix file system contains 15 block pointers with 12 as direct block pointers and the remaining three for single, double and triple indirection. Consider an alternate design for an inode in the Unix file system where it only contains 4 block pointers—one as a direct block pointer and the remaining three for single, double and triple indirection. Discuss any advantages and disadvantages of this alternate design in comparison to the existing approach.

8. (8 points) What are the three time factors involved in positioning the disk head and reading/writing a disk block? What approaches are used to reduce (if possible) each of these factors for given a set of disk requests.

9. (6 points) The ACME computer company is marketing a new disk driver that uses the elevator algorithm and also queues multiple requests within a cylinder in sector order. Your company buys one and you put another of your ace employees, Harry Hacker, on the job of evaluating it. He writes a program to randomly read 10,000 blocks spread across the disk. To his amazement, the performance that he measured is identical to what would be expected from first-come, first-served. Who is right, ACME or Harry? Why?

10. (9 points) In Project 5 you used sockets for communication between a client and server.

(a) In Project 3 you implemented mailboxes for communication amongst threads. How are these two communication mechanisms the same in terms of how you used them? How are they different?

(b) What are the advantages of a single-process, single-threaded server?

(c) What are the advantages of a multi-process, single-threaded server?

(d) What are the advantages of a single-process, multi-threaded server?

11. (9 points) In Project 4 you implemented the clock policy to handle replacement of cached files. Consider a new policy, called *clock frequency*, which also considers frequency of use. In this policy the reference "bit" is replaced by a reference count. The reference count is incremented on each access and decremented on each sweep of the clock. Just as with the clock policy, a file is replaced when its count reaches zero.

    Explain whether the clock frequency policy will provide *better*, *worse* or *about the same* cache hit rates when compared to the clock policy for each of the following three cache sizes. For a given set of files, the maximum cache size needed is the total amount of cache space needed to cache all unique files.

    (a) A small cache permitting only a small number of files to be cached at a time.

    (b) A medium size cache with about half of the maximum cache size needed.

    (c) A huge cache with almost as much space available as the maximum cache size needed.

THIS PAGE INTENTIONALLY LEFT BLANK.