CSE 121: Operating Systems - Architecture and Implementation
Homework 1
Fall 2003

Due: Tuesday, October 14

1 Processes and Virtual Memory

a) An operating system maintains state for each process in the form of a process control block. If the operating system also supports kernel threads, it must maintain per-thread state as well. Identify which of the following must be maintained per process and which must be maintained per thread:

- UID:
- Stack Pointer:
- Block Condition:
- Memory Map:
- File Descriptors:
- Execution State/Registers:

b) If we didn't have a virtual memory system, describe how an operating system would be able to support multiprogramming (where multiple programs would be resident in memory at a time). Specifically, explain any additional difficulties in allowing programs to be loaded into memory and subequently paged or swapped during execution. If you do not think this is possible, explain why.

c) What is the supervisor bit? Why does it need to be implemented in hardware?

2 File Systems

One defining characteristic of a file system is the data allocation method used. The FAT file system (used in MS-DOS) was presented in lecture as an example of a file system using *linked* allocation. FFS (used in Unix) is an example of a file system using *indexed* allocation. In the context of performance (file system operations), robustness (what happens if things crash), and space efficiency (how much of the disk is used for data vs meta-data) give specific examples showing how each scheme has advantages over the other.

3 Fast File System

a) Give two reasons why doubling the block size more than doubled the system throughput. (*Hint*: what type of workload was used to measure the system throughput?)

b) Why does FFS need to maintain a free space reserve?

c) How many blocks and fragments are allocated to a 31,200-byte file on a FFS with 4096-byte blocks and 1024-byte fragments? How many blocks and fragments are allocated to this file on a FFS with 4096-byte blocks and 512-byte fragments? Also answer these two questions assuming that an inode had only six direct block pointers, instead of 12.

d) Why are FFS inode allocation policies different from those for data blocks?

4 Log-Structured File System

a) LFS was designed around particular assumptions about relavent disk/cache technology and file access patterns. These tradeoffs may or may not be valid today. Describe the tradeoffs in using LFS versus FFS, an example of usage scenarios in which LFS would outperform FFS, and a scenario in which FFS would outperform LFS will suffice. (Do not use the LFS cleaning mechanism as one of your examples.)

b) Construct an example of an LFS segment where cleaning would lose, rather than gain free blocks.