Examination 3

COP 4600 Operating Systems
April 17, 1997

Instructions

1. **Read and follow all instructions.** Failure to do so will result in penalty.
2. This is a closed-book examination.
3. You are permitted one 8.5 by 11 inch sheet of notes, both sides, that you have prepared.
4. **Answer any three (3) questions**. All questions are of equal value. Points for each part of each question are given in parentheses.
5. **Leave sufficient room in the upper lefthand corner for the staple.**
6. **Use no more than two pages** of paper (both sides OK) to hold the answer to each question.
7. **Write legibly.**
8. **Start** the answer to each question on a **new page** (i.e., do **not** put the answer to more than one question on the same page).
9. Assemble your answers in **numerical order** of the questions when you submit them.
10. **Read and sign the following statement.** You may write this on your exam and sign it there if you wish to take the exam questions home with you today. Do NOT discuss this exam with anyone until after they have taken it - ask to make sure.

On my honor, I have neither given nor received unauthorized aid on this examination.
Signed:

1. (a) (9) What is binding, relative to main memory management in OS? At what times may binding occur, and how does it occur at each of these times?
   (b) (8) What is dynamic loading? Why is it used? Are there any disadvantages to its use? If so, what are they?
   (c) (8) What is dynamic linking? Why is it used? Compare to static linking.

2. Consider the following sequence of requests and releases of main memory in a system with 1024 KB of storage (all requests are in KB): R1: 200, R2: 115, R3: 130, R4: 280, R5: 60, R6: 98, Release R2, Release R5, Release R3, R7: 100, Release R1. State what actions are taken (if the request makes sense!) and the state of memory after each request is handled for each of the following methods. Include any queues of pending requests that were not satisfied and any structures used for free space management.
   (a) (6) Contiguous allocation with minimum allocation unit of 10 KB. Assume first fit allocation.
   (b) (6) MFP with partitions of 100K, 100K, 200K, 200K, and 424K (in that order in memory). Assume that a request may only be allocated the smallest partition size in which it will fit.
   (c) (7) Buddy system.
   (d) (6) Where does (and what types of) fragmentation exist in each of the systems above? Be specific with representative examples from each system.

3. Assume a pure demand paging environment with local page replacement. Show the final state of memory after the reference string has been processed, and give the number of page faults for each of the following replacement algorithms and allocations for the following reference string: 1, 2, 5, 2, 5, 2, 6, 2, 6, 2, 7, 2, 3, 7, 2, 5, 2, 3, 7, 2, 3.
   (a) (7) FIFO with 3 frames; FIFO with 4 frames.
   (b) (7) LRU with 3 frames; LRU with 4 frames.
   (c) (7) CLOCK with 3 frames; CLOCK with 4 frames.
   (d) (4) Give the optimum page replacement sequence for 3 frames along with the number of page faults it produces.

4. (a) (9) In a system that permits multiple links to the same file, what are three distinct ways of handling the problem of file deletion without leaving dangling pointers? Describe and compare these methods.
   (b) (6) What is the difference between a relative path name and an absolute path name? Describe how a path name is resolved to the file control block desired.
   (c) (5) What is the difference between symbolic links and hard links? What restrictions are there on their use in Unix?
   (d) (5) What does the mount command do, and how does it do it? Be specific. What does the operating system do when it resolves a link to a mounted file system?

5. For the first three subparts of this question, consider a file of size 400 KB stored on a sytem that uses a block size of 1 KB. A process has opened the file (so its FCB is in RAM), but none of the actual file blocks are in RAM. If the process issues a read of byte 204,867, give the computations showing what logical block is accessed, what the offset is within that block, and how many disk operations must performed in order to read the desired byte for each of the following file allocation methods. Assume 24 bits per physical block number. Show your work.

   (a) (4) Contiguous allocation.

   (b) (6) Linked allocation.

   (c) (6) Two-level indexed allocation. Also, what is the maximum file size in this sytem?

   (d) (4) How is the System File Table used in Unix? What are the fields of its entries and how are they set and used?

   (e) (5) What is the largest size disk that Minix can use if zones are of size 4KB each? What is the largest file size that Minix can have in this system? Show your work.