CSE 380: Introduction to Operating Systems

Final Exam

17 December 2003

There are six problems and each problem is worth 20 points. Although some problems are easier to answer than others, they are all equally weighted. Do any FIVE of them, but no more than five. Please indicate which problems that you want us to grade if you answer more than five problems. If not specified, we will pick any five of them randomly.

Please write your answers **legibly** so that we can understand your answers. If a problem seems ambiguous or impossible, please feel free to state your assumptions explicitly and solve the problem. Obviously, your assumptions should be reasonable and should not trivialize the problem. Please hand in only the exam answerbook. We will not grade any answers written on the copy of the exam itself.

Please sign the **pledge** at the back of the answerbook. Good luck!

1 (20 points)

1. What is the difference between external fragmentation and internal fragmentation?

   Answer:
   When areas of memory are allocated, space can be wasted in two general ways. Wasted space inside of an allocated area of memory is referred to as internal fragmentation. External fragmentation refers to the wasted space outside of the allocated areas of memory.

2. What are the pros and cons of choosing a small page size? What are the pros and cons of selecting a large page size?

   Answer:
   large pages:
   pro - smaller page table, less page faults, less overhead in reading/writing of pages
   con - more internal fragmentation, worse locality of reference

   smaller pages:
   pro - reduces internal fragmentation, better with locality ofreference
   con - bigger page table, more page faults, overhead in reading/writing of pages

3. Describe the three possible states of a process?

   Answer:
   Running: actually using the CPU at that time
   Ready: runnable; temporarily stopped to let another process run
   Blocked: unable to run until some external event happens (e.g. I/O done)

4. In modern operating systems, a process consists of multiple threads. What are the differences between process and thread? What does a thread have exclusively ?

   Answer:
   Multitasking OS can do more than one thing concurrently by running more than a single process. A process can do several things concurrently by running more than a single thread. Each thread is a different stream of control that can execute its instructions independently. A thread has its own stack and program counter (PC).

5. Describe execution steps in RPC (Remote Procedure Call). Be sure to explain the need for parameter marshalling.

   Answer:
   - Steps in RPC
   1. Client invokes a library routine called client stub, possibly with parameters.
   2. Client stub generates a message to be sent: parameter marshalling.
   3. Kernels (RPC runtime) on client and server handle communication.
   4. Receiver kernel calls server stub.
   5. Sever stub unpacks parameters and invokes server routine.

   Parameter marshalling is needed because representataion needs to deal with byte ordering issues as following.

- different data representation (ASCII, UNICODE, EBCDIC)
- big-endian versus little-endian
- strings (some CPUs require padding)
- alignment, etc.

2 (20 points)

1. What is the fundamental limitation of secret-key cryptography? How is this limitation overcome with public-key cryptography?

   answer: Secret-key crypto requires that keys be shared by communicating parties. If two parties do not already share a key or a secure channel, they must physically transfer the keys to each other. Public-key crypto allows a public key to be viewed by anyone when it is distributed without affecting the integrity of the holder of the private key.

2. Why would someone want to both sign and encrypt data sent to someone? answer: An encrypted message can be sent to the holder of the private key by anyone who has the public key. The recipient knows that the message is encrypted but cannot verify who actually sent the message. The signature provides assurance to the recipient that the data has been encrypted and that the sender actually sent the message.

3. How are hash functions used in the context of password storage and what is the purpose of the hash function? Explain the vulnerability of using a poor hash function for hashing passwords.

   answer: Hash functions allow passwords to be stored on disk without having to store the plaintext of the password. A user logs in with their password, the system applies the hash function, and compares the output with the known hash output. If a poor hash function is chosen, an attacker would have an easier time guessing another user's password, because the probability that different passwords hash to the same hash value is greater.

4. Two different protection mechanisms are capabilities and access control lists. Explain each of these two protection mechanisms. Also, provide two situtations, one that is better for capability based system and another that is better for access-control list based system.

   Answer: Problem 35 in Chap 9

3 (20 points)

1. Explain how a buffer overflow occurs and can be exploited by an attacker. What can a programmer do to prevent a buffer overflow?

   answer: A buffer overflow occurs when data is written into memory that is larger than the programmer intended. The goal of the overflow is to overwrite the return address of the function to a new value and have the process execute code contained in the

large buffer. To protect against a buffer overflow, programmers can use variations of functions that limit the size of the operation such as strncpy instead of strcpy

2. Prove that if any message can be lost, it is not possible for two processes to agree on non-trivial outcome using only messages for communication.

   Answer: see the lecture note on distributed systems.

3. For the following questions on user-level threads vesus kernel-level threads, assume that process A has one thread and process B has 10 threads, and the scheduler allocates the time slices equally. Answer the following questions.

   (a) If threads are in user level, which thread between a thread of A and a thread of B is scheduled longer by how much, or the same? Justify your answer.
   Answer: A thread in process A runs 10 times as fast as a thread in process B.

   (b) If threads are in kernel level, which process between A and B is scheduled longer by how much, or the same?
   Answer: Process B receives 10 times the CPU time than process A.

   (c) When a thread in a process invokes a system call, are the other threads in the process also be blocked, or not? Answer for both processes A and B.
   Answer: In the user-level threads, all threads in process B are blocked, but in the kernel-level threads, they are not blocked.

   (d) Which case is more time-consuming to switch thread in process A and process B? Justify your answer.
   Answer: Kernel-level thread because context switching is needed.

4 (20 points)

1. What are four necessary conditions of deadlock? For each of the conditions, describe a scheme to prevent that condition from being true (to prevent deadlock).

   Answer:
   - Mutual Exclusion - spooling
   - Hold and Wait - a process requests all necessary resources at once and then get a grant if all are available; otherwise, deny.
   - No Preemption - preemption is allowed
   - Circular Wait - hierarchical allocation

2. Describe a deadlock prevention scheme for a distributed system using time-stamp ordering. Prove/explain how the scheme prevents deadlock.

   Answer: Wait-Die or Wound-Wait.
   Wait-Die is non-preemptive. When P requests a resource currently held by Q, P is allowed to wait only if it is older than Q. Otherwise, P is rolled back (i.e., dies).

   Wound-Wait is preemptive. When P requests a resource currently held by Q, P is allowed to wait only if it is younger than Q. Otherwise, Q is rolled back (releasing its resource). That is, P wounds Q.

3. Consider a virtual memory system with 34-bit addresses. The first 23 bits are used as a page number, and the last 11 bits is the offset. (Note that 2^{10} = 1K, 2^{20} = 1M.)

   (a) How many Kilobytes (KB) is a single page frame?
   Answer: 2^{11} = 2KB

   (b) Assuming the single-level paging and each page-table entry to be 4 bytes, how many Megabytes of memory is needed to store the page table?
   Answer: 2^{23} entries * 4 bytes = 32 MB

   (c) Suppose the system using two-level paging, with first n bits ($n > 11$) of the address used as an index into the first-level page table. Assume that a typical program uses 8MB memory. White an expression in terms of n that gives the number of second-level page-tables used by the typical program.
   Answer: One second level page-table covers $2^{(34-n-11)} * 2^{11} = 2^{(34-n)}$ bytes. So the number of second-level page-tables for the program is 8MB/ $2^{(34-n)}$ bytes = $2^{(n-11)}$.

   (d) How many entries does an inverted page-table need on a machine with 128 MB physical memory?
   Answer: The number of entries in inverted page table is same as the number of physical memory page frames. So, 128MB / 2KB = 64K

4. Suppose you search for an element x in a sorted array of N elements. A linear search would require $N/2$ memory accesses on average, while a binary search would require ($\log N$) memory accesses. Assume that the array spans over multiple pages (e.g. a page holds p array elements but $N > p$). How will the performance of the two algorithms be affected due to paging. Explain it with page faults. (Hint: the cost of page fault >> the cost of memory access)

   Answer:
   For the linear search, a page fault will be generated every p accesses, so overall about $N/2p$ page faults. For the binary search, every access will generate a page fault until the search range reduces to about p. So number of page faults will be $\log(N/p)$. Finally, the binary search is still better than the linear search.

5 (20 points)

1. For file sharing, explain a hard link and a soft link. Explain one drawback for each method.

   Answer:

   Hard link: Pointer to i-node for the shared file is added to the directory entry. Whenever the file is shared by other process, link counter is incremented. I-node can be deleted only when the link counter goes down to 0. The original owner can not free disk quota unless all hard links are deleted.

   Soft link: The system creates a new file, of type LINK, and enters that file in the directory. The new file contains just the path name of the file to which it is linked.

Extra overhead is required. The file containing the path must be read, then the path must be parsed and followed until the i-node is reached. All of this activity may require a considerable number of extra disk accesses. (IL: also staled/dangling referece problem)

2. In the Bully algorithm, the goal is to elect a new leader when the current leader is crashed in distributed systems. The leader is the active process with the maximum ID. Assume that there are 8 processes (ID:0, ..., 7) and 7, 5 and 3 processes are currently failed, and process 2 detects a failure of current leader. Show how they elect a new leader with maximum ID. You may find figures helpful.

   Asnwer: 0, 1, 2, 3(x), 4, 5(x), 6, 7(x) Initially, process 2 sends an "election" message to 3, 4, 5, 6 and 7. Then, only 4 and 6 respond, telling 2 to stop. Now, 4 and 6 send an "election" to 5, 6 and 7, and to 7, respectively. Then, 6 sends "STOP" to 4, but 6 does not get "STOP" from 7 because 7 is crashed. Finally, from time interval T, process 6 sends "Coordinator" message to all processes with lower ID. Processes which are received "Coordinator" message from 6 record the ID(6) as a new leader.

3. What is "false sharing" in DSM (Distributed Shared Memory)? How can this problem be eliminated?

   Answer: No variables actually shared, but they may reside on the same page

4. What is the "happen-before" relation in distributed systems? Show that it is a partial order not a total order.

6 (20 points)

1. What is a multiprocessor system?
2. Why is caching necessary for a multiprocessor system?
3. Provide an example in which caches in a multiprocessor system can be inconsistent.
4. Describe the difference between invalidate protocol and update protocol?
5. Identify five types of failures in RPC.

   Answer:

   1. cannot locate the server
   2. request msg is lost
   3. reply msg is lost
   4. server crashes after receiving a request
   5. client crahses after sending a request