Stanford University
Computer Science Department
CS 140 Final Exam
Dawson Engler
Winter 1999

Name: ____________________________________

Please initial the bottom left corner of each page.

This is an open-book exam. You have 120 minutes to answer as many questions as possible. The number in parenthesis at the beginning of each question indicates the number of points given to the question. Write all of your answers directly on the paper. Make your answers as concise as possible. Sentence fragments ok.

Stanford University Honor Code:
In accordance with both the letter and the spirit of the Honor Code, I did not cheat on this exam.

Signature: _______________________________________

Problem	**Points**	**Score**
1-4	16	
5-7	22	
8-9	25	
10	20	
11	27	
12-13	38	
Total	148	

True False Questions

Each question is worth 4 points.

Please answer each question as true or false and concisely state the intuition behind your answer.

1) Without “free” (deallocate) it is easy to write a “malloc” implementation that never fragments memory. (True/false)

2) You run a workload using a fixed-size 1 MB (megabyte) buffer cache and notice that there are many disk references. If you repartition main memory to increase the buffer cache size to 2 MB, the workload may actually run significantly slower. (True/false)

3) You profile your file system and notice that over the course of the day the system used 6200 files, whose total size in bytes was 100 times larger than the size of the file system’s buffer cache. A cache cannot provide much benefit for this type of workload. (True/false)

4) You install new network hardware that you’ve purchased from a “MORE BITS NOW!!!” infomercial. Unfortunately, you notice that packets arrive out of order much more frequently than on it than on the old network. The new network is more likely to be connection-based rather than connection-less. (True/false)

Short Answer questions

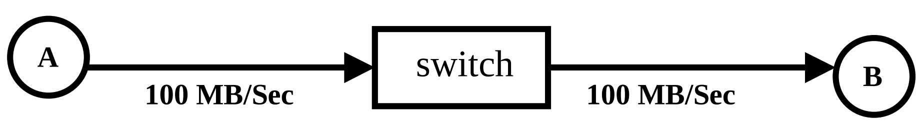
5) [6 points] You conceive of a get-rich scheme of retrofitting virtual memory onto old government computers. To save time, you base your virtual memory system on the MIPS architecture, which when a TLB fault occurs, does the following three actions:

   1). Writes the program counter of the faulting instruction into the special register "epc."
   2). Writes the faulting virtual address into the special register "badva."
   3). Changes to privileged mode and transfers control to a kernel TLB handler.

If a translation for badva exists, the OS installs it and jumps to the program counter value held in epc. If no translation exists, the OS signals an exception. What problem do actions (1) and (2) cause for "mapped" page tables? (I.e., page tables that are mapped using virtual memory.)

6) [6 points] Recall that we can enforce security policies using either technical mechanisms (e.g., locks on doors) or social mechanisms (e.g., laws against stealing). The Stanford Honor code is an example of a social mechanism used to enforce a security policy. State two assumptions that the Honor Code makes that a technical approach would not need. (Hint, consider why leland requires passwords.) In situations where these assumptions are valid state an attack it prevents that a more technical approach would have difficulty with.)

7) [10 points] Consider the following switch attached to a single 100MB/s incoming link and a single 100MB/s outgoing link.



a) [5 points] If we view incoming packets as jobs, and the switch as a processor, what CPU algorithm does the above system correspond to?

b) [5 points] What weakness of that scheduling algorithm do you expect to see replicated here? What packet feature can you control to minimize this problem?

8) [15 points] Threads on a new multithreaded WWW browser periodically query a nearby WWW server to retrieve documents. On average, a browser's thread performs a query every N instructions. Each request to the server incurs an average latency of T milliseconds before the answer is received.

a) [5 points] For N = 2,000 instructions and T = 1 millisecond, what is the smallest number of such threads that would be required to keep a single 100 MIPS processor (i.e., executing 100 million instructions per second) 100% busy? Assume the context switch time is instantaneous and that the scheduler is optimal

b) [5 points] Unfortunately, context switches are not instantaneous. Assume that a context switch takes C instructions to perform. Recalculate your answer to part (a) for C = 500 instructions.

c) [5 points] Excited by your calculations, you add threads to your system. However, system throughput improves by a much smaller amount than part (b) would suggest. Give two features of a computer system that could be the reason for this dismal outcome.

9) [10 points] You have written a mark and sweep garbage collector for use on a single processor machine. The collector is woken up once a second by a timer interrupt and:

1) Disables all interrupts.
2) Finds all active memory.
3) Compacts memory into a two contiguous address ranges, where one range holds all free memory, the other all allocated memory
4) Sets malloc's global heap pointer to point to the new range of free memory.
5) Reenables interrupts.

a) [5 points] Assume only one garbage collection thread can run at a time and the application is single threaded. Does the locking discipline in the following code guarantee mutual exclusion for the global heap pointer? (Why or why not?)

```
int lock;
void *heap;

void *malloc(unsigned nbytes) {

    /* …  */
    lock = 1;
    heap = (char *)heap + nbytes;
    lock = 0;
    /* … */
}

void initiate_gc(void) {
        if(lock)
                return;
        /* find free memory */
        heap = new_heap;   /* reset heap */
}
```

b) [5 points] A program that formerly paged heavily runs significantly faster when your garbage collector is used. What is a likely reason for this? (Assume that you have determined that the program does not leak memory.)

10) [20 points] Suppose the longest packet you can transmit on the Internet can contain 480 bytes of useful data, you are using a lock-step end-to-end protocol, and you are sending data from California to your geeky cousin at MIT (go team). You have measured the round-trip delay and found that it is about 100 milliseconds. (A lock-step protocol is one where you must wait for the recipient to send an acknowledgement of your previous packet before you can send the next packet.)

a) [5 points] If there are no lost packets, estimate the maximum data rate (expressed in megabytes per second) that you can achieve.

b) [5 points] Unfortunately, 1% of the packets are getting lost. So you install a "resend" timer, set to 1000 milliseconds. Estimate the data rate you now expect to achieve. (For simplicity, assume no retransmitted packets are lost.)

c) [5 points] On Tuesdays the phone company routes some eastward-bound packets via satellite link, and you notice that 50% of the round trips now take exactly 100 extra milliseconds. What effect does this delay have on the overall data rate when the resend time is not in use? (Assume the network does not lose any packets.)

d) [5 points] An unscrupulous startup steals your code as a core technology. However, they haven't heard about satellite delays and set the resend timer to 150 milliseconds. What is the data rate now on Tuesdays? (Assume the network does not lose any packets.)

11) [24 points] Recall that a simple definition of fragmentation is "the inability to use memory that is free." Of course, whenever we do any sort of allocation --- of CPU time quanta, of disk blocks, of cache space --- fragmentation is an issue. For example, if we allocate fixed sized units, internal fragmentation is a problem. A more general statement would be "the inability to use X that is free."

a) [4 points] Give two examples where an OS intentionally caused internal fragmentation.

b) [2 points] How does malloc minimize internal fragmentation?

c) [4 points] What corresponds to internal fragmentation in the IP address space? How does the agency that gives out addresses minimize it?

d) [6 points] What corresponds to internal fragmentation of a process's time quanta? How does the CPU scheduler minimize it?

e) [4 points] How does switching from a connection-based to connectionless network eliminate fragmentation of router memory? What does it give up to do this?

f) [4 points] One effect of external fragmentation is that we may not be able to satisfy an allocation request, even though there is enough free memory to do so. Why is external fragmentation not a problem with money?

12) [15 points] Routes R. Russ decides he's sick of having his router rely on buggy routing protocols to dynamically figure out which outgoing port to send arriving IP packets out on. Instead, he decides to create a huge file, "routes.txt" that, once and for all (dammit), records the mapping of 32-bit IP address to outgoing port for every host on the Internet. His idea is that once he has created this file, he'll never have to talk to Crisco boxes again.

Since his router only has 254 outgoing links, he decides to treat each byte in the file as a port number, and the offset of this byte as the IP address associated with this port. Any IP address whose port is unknown is flagged with the special port 0. Thus, the port number for IP address 0.0.0.0 is at offset 0, for 0.0.0.1 is at offset 1, and so forth. Given this organization, the following routine returns the port associated with the given IP address:

```
/* route_fd is a file handle for the routes.txt file. */
unsigned char ip_to_port(int route_fd, unsigned ip_addr) {
        unsigned char port;

        lseek(route_fd, ip, SEEK_SET);    /* move the file pointer to the right offset */
        read(route_fd, &port, 1);         /* read the port associated with this route. */
        if(port == 0) {                   /* Sigh. I thought we were done */
                port = determine_port(ip);   /* talk to other routers to figure out the port */
                lseek(route_fd, -1, SEEK_CUR);  /* back up one byte */
                write(route_fd, &port, 1);      /* write out the port */
        }
        return port;
}
```

Recall that the IP address space is divided into three classes (A, B, and C):

0	net id	host id
1 bit	7 bits	24 bits

10	net id	host id
2bits	14 bits	16 bits

110	net id	host id
3 bits	21 bits	8 bits

a) [5 points] Assume Russ is using a "classic" Unix file system with the following components:

1) Disk blocks are 4096 bytes. Sectors are 512 bytes long.
2) All metadata pointers are 32-bits long.
3) An inode has 12 direct block pointers, one indirect block pointer and one double-indirect block pointer. The total inode size is 256 bytes.
4) Both indirect and double indirect blocks take up an entire disk block.

How much disk space, including metadata and data blocks, is needed to store routes.txt? Please show your calculation.

b) [5 points] Assuming Russ does not use a buffer cache. How many disk reads will be required to read the last byte in this file? To write it? (Please show your calculation.)

c) [5 points] Mr. Russ adds a buffer cache. It speeds everything up a lot. He gets really excited. He then notices that if you send a packet to one host on a network you'll likely send packets to other hosts as well. He decides to restructure his table to exploit this. Allysa P. Hacker says that because of the way the IP addresses are organized that the current system implicitly does this. Who is right and why?

13) [22 points] Mr. Russ's router is so successful that he hires Allyssa P. Hacker to make his router reliable across crashes.

a) [8 points] As her first official act, she points out that a normal Unix file system may write newly allocated disk blocks back to disk after the metadata that points to them has been written out. What problem does a system crash cause in this situation? Suggest a simple, reasonable fix for this problem. You may suggest file system alterations if necessary. (Recall that a disk sector can be written atomically, and a block on this system is made of 8 sectors.).

b) [6 points] Spastic with excitement after selling a copy of his router to his mother, Mr. Russ decides to add another file, “arp.txt,” that records the binding of IP address to Ethernet address for every node on his local network. Since Ethernet addresses are 6 bytes long, what problem can a crash cause if Russ is in the middle of writing an entry to arp.txt?

c) [8 points] Assume the router hardware provides 10 bytes of non-volatile RAM (i.e., memory that persists across system crashes). Sketch how this memory can be used to fix the problem in part (b). In particular, indicate what must be done on writes to arp.txt and what must be done after a crash.