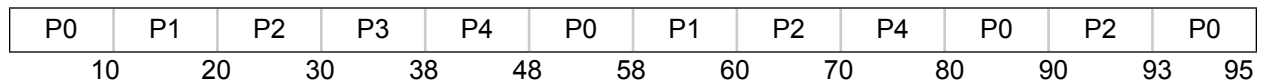


405 Operating Systems. Final exam. 2010.

Total XP: 250

HOW TO SUBMIT: Because I am old and have a hard time reading the script of the young, please submit your answers typed (PDF preferred) to submit.o.bot@gmail.com.

1. **CPU Scheduling (15XP)** Suppose 5 jobs arrive nearly simultaneously into an empty ready queue in a system with a single CPU core (The order is P0, P1, P2, P3, P4). The following Gantt chart shows the result from RR scheduling.



The times indicated are associated with the vertical lines on the Gantt chart. (The timeline is not to scale). The last job completes at time 95.

a) (3 points) What is the quantum size (q) for this example? _____

b) (4 points) Calculate the job sizes in this example

P0:

P1:

P2:

P3:

P4:

c) (4 points) Draw the Gantt chart for RR with $q = 20$.

d) (4 points) Draw the Gantt chart for FCFS

2. **Semaphores (35XP)**. There are three types of threads that access a shared linked list: retrievers, adders, and removers. Retrievers walk through the linked list to find a particular item. Since they merely examine the list, there can be multiple retrievers accessing the list concurrently. Adders add new items to the end of the list. Adding an element must be mutually exclusive meaning multiple adders cannot access the list concurrently. However, an add can proceed in parallel with any number or retrievals. Finally, removers remove items from anywhere in the list. At most one remover can

access the list at a time and removing an item must be mutually exclusive with retrievals and additions.

Puzzler: Write code for retrievers, adders, and removers, that enforce this kind of three-way categorical mutual exclusion.

3. System Calls (25XP).

a) (10XP) What is the purpose of a system call in an operating system. (in your own words, be clear, and aim the answer at an undergrad cs major who has not taken OS)

b) (15XP) Describe the steps from executing an API call like printf in a userland program to the execution of the associated system call in the OS.

4. General Question (25XP). This is one you may want to mull over for awhile.

This semester I seem to be flying frequently between my home in Las Cruces and here. I am tired of lugging around my heavy MacBook Pro. I have a desktop here and in Cruces. However, I would like to have the exact same operating system experience regardless of where I am. For example, if I download an episode of Chuck to my desktop in Cruces, I want it to be on my desktop in Fredericksburg. If I download and install an app like VLC in Fredericksburg, I want it to be installed in Cruces. If I set my background picture to some old-skool 8 bit pattern, it should be my background everywhere. Finally, in addition to the two desktops, I want the same experience on my laptop.

I have the following resources: an Altoids tin, a 20GB Dropbox account, nail clippers (sadly I can't get my pocket knife through TSA security), a 32GB USB memory stick. and some old ukulele strings. From the above description you can see that I can install what I want on the physical computers.

How would you accomplish this? Give **specific instructions** that I can follow.

[clarification to avoid confusion: This is a thought experiment. I am not soliciting advice on how to solve a real problem I have]

5. Memory (20 XP)

Consider a 32-bit virtual address divided up as follows:

4 bit outer page number	12 bit inner page number	16 bit offset
-------------------------	--------------------------	---------------

Here are the relevant tables (values in hexadecimal):

Outer Page Table	
0	Page Table A
1	Page Table B
x	(rest invalid)

Page Table A	
0	CAFE
1	DEAD
2	BEEF
3	BA11
x	rest invalid

Page Table B	
0	F000
1	D8BF
x	(rest invalid)

Find the physical address corresponding to each of the following virtual addresses (answer 'bad address' if the virtual address is invalid)

1. 0x00000000
2. 0x20022002
3. 0x10015555
4. 0x10030099

6. Demand Paging (25 XP)

Consider the following experiment and explain the observations.

A program is run by itself on a paging machine. The program begins execution with its first procedure page. As it runs, the pages it needs are demand paged into available

page frames. There is a dial external to the computer that allows a person to set the maximum number of page frames the program may use. In addition there is an LED display that indicates the total number of page faults that occurred during program execution. Initially, the dial is set to 2 frames and the program is run to completion. The dial is then set at 3 frames and again the program is run to completion. This process is continued until the dial is eventually set to the number of available pages in real storage, and the program is run for the last time. For each run, the execution time and number of page faults is recorded

Observation 1:

As the dial is changed from 2, to 3, to 4, the number of page faults declines. As the dial is changed to 5 the number of page faults rises. From 5 to 6 and onward, the number of page faults again declines.

Observation 2 (a completely separate study):

For a different program we follow the same procedure. As the dial is changed from 2, to 3, to 4, the execution time improves dramatically. From 4, 5, to 6, the execution time still improved each time, but less dramatically. With the setting of 7 and higher the execution time is essentially constant.

7. TLB (25 XP)

1. A computer system has a page size of 1,024 bytes and maintains the page table for each process in main memory. The overhead required for doing a lookup in the page table is 500 ns. To reduce this overhead, the computer has a TLB that caches 32 virtual pages to physical frame mappings. A TLB lookup requires 100ns. What TLB hit-rate is required to ensure an average virtual address translation time of 200ns?

2. Discuss the issues involved in picking a page size for a virtual memory system.
- a) Name one issue that argues for small page sizes? Name two that argue for large page sizes?
 - b) How do the characteristics of disks influence the selection of a page size?

8. Memory (15XP)

Given the following program that uses three memory segments in an address space as described in class (code segment, data segment, and stack segment):

```
char a[100];
main(int argc, char ** argv)
{
    int d;
    static double b;
    char *s = "boo", * p;

    p = malloc(300);
    return 0;
}
```

Identify the segment in which each variable resides and indicate if the variable is private to the thread or is shared among threads.

Be careful.

9. Ada Codeslinger (25XP)

One day famed University of Mary Washington student Ada Codeslinger had an inspiration while stuck in traffic. She observes that most programs have most of their data at the beginning of the address space. For her homegrown Eagle OS, she decides that she is going to implement her page tables similar to the way Unix implements inodes. She calls this page table design Inode Page Tables. Inode Page Tables are essentially two-level page tables with the following twist: The first half of the page table entries in the master page table directly map physical pages, and the second half of the entries map to secondary page tables as normal. Call the first half the entries fast, and the second half normal.

For the following questions, assume that addresses are 32 bits, the page size is 4 KB, and that the master and secondary page tables fit into a single page.

- 1 How many virtual pages are fast pages?
- 2 How many virtual pages are normal pages?
- 3 What is the maximum size of an address space in bytes?
- 4 Inode Page Tables reduce the lookup time for fast pages by one memory read operation. Do you think that this is an effective optimization? Briefly explain.

10. iNode (25XP)

Consider an indexed file allocation using index nodes (inodes). An inode contains among other things, 7 indexes, one indirect index, one double index, and one triple index.

- What usually is stored in the inode in addition to the indexes?
- What is the disadvantage of storing the file name in the inode? Where should the file name be stored?
- If the disk sector is 512 bytes, what is the maximum file size in this allocation scheme?
- Suppose we would like to enhance the file system by supporting versioning. That is, when a file is updated, the system creates new version leaving the previous one intact. How would you modify the inode allocation scheme to support versioning? Your answer should consider how a new version is created and deleted.

11. Backup (25XP)

The Land of Enchantment backup system for Unix works as follows. At the beginning of each week, the backup system traverses the file system tree structure and saves all directories and files. This stage is called full backup. Then, on a daily basis, it performs incremental backups. It does so by traversing the directory tree, and saving only the files whose time stamps show that they have been modified since the previous incremental backup. Note that the Land of Enchantment system does not follow the symbolic links when saving files, instead storing them as symbolic links. This preserves the integrity of the symbolic link.

1) What happens if the backup system were to follow the symbolic links and save the files the link points to?

2) Identify problems with this backup scheme.