University of California, Berkeley
College of Engineering
Computer Science Division – EECS

Spring 2008 Anthony D. Joseph

Midterm Exam #2

April 16, 2008
CS162 Operating Systems

Your Name:	
SID AND 162 Login:	
TA Name:	
Discussion Section Time:	

General Information:
This is a **closed book and notes** examination. You have 90 minutes to answer as many questions as possible. The number in parentheses at the beginning of each question indicates the number of points given to the question; there are 100 points in all. You should read **all** of the questions before starting the exam, as some of the questions are substantially more time consuming.

Write all of your answers directly on this paper. ***Make your answers as concise as possible.*** If there is something in a question that you believe is open to interpretation, then please ask us about it!

Good Luck!!

Problem	Possible	Score
1	30	
2	22	
3	23	
4	25	
Total	**100**	

1. (30 points total) Short answer questions:
   a. (12 points) True/False and Why?
      i) Small time slices always improve the average completion time of a system.

TRUE FALSE

Why?

ii) Shortest Job First (SJF) or Shortest Completion Time First (SCTF) scheduling is difficult to build on a real operating system.

TRUE FALSE

Why?

iii) You run a workload using a fixed-size 100 MB (megabyte) buffer cache and notice that there are many disk references. If you repartition main memory to increase the buffer cache size to 200 MB, the workload may actually run significantly slower.

TRUE FALSE

Why?

iv) You profile your file system and notice that over the course of the day the system used 6,200 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

Why?

b. (5 points) Roughly order stack, code, and heap segments as to how well you would expect them to perform under a paged VM system that uses LRU page replacement. *State the intuitions for your ordering.*

c. (5 points) You're hired to develop the next-generation of high performance Web 2.0 servers. One of the developers on your team proposes the idea of using disk compression to store data (storing data in compressed disk blocks), but your boss disagrees saying that disk storage is cheap and there's no reason to use compression. Do you agree with your teammate or your boss? What are the arguments (tradeoffs, benefits, and disadvantages) in favor of or against disk compression?

d. (8 points) Since high-performance scientific computing applications run on very expensive supercomputers, it is very important to optimize their performance. Consider the following scientific computing function:

```
MatrixAdd(int a[][], b[][], int n) {
      for (i = 0; i < n; i++)
         for (j = 0; j < n; j++)
           a[j][i] += b[j][i];
}
```

The matricies a and b are allocated using the following code:

```
MatrixAllocate(int n) {
      a = malloc (n * size of *a);
      for (i = 0; i < n; i++)
         a[i] = malloc(n * sizeof a[i]);
      return a;
}
```

You observe that the program runs more slowly than expected when calling `MatrixAdd`, so you turn on profiling and notice that the function is causing large numbers of TLB flush operations – much more than expected. Explain the likely cause and implement a simple fix. Also, explain what you generally expect to happen to paging performance when your fix is implemented?

2. (22 points total) Multi-level Address Translation.

a. (10 points) You're hired by the HAL Corporation to develop their next generation machine, the HAL 2010. It will have a virtual memory architecture with the following parameters:

- Virtual addresses are 48 bits.
- The page size is 32K byte.
- The machine has 128 Terabytes (TB) of real memory (RAM).
- The first- and second-level page tables are stored in real memory.
- All page tables can start only on a page boundary.
- The maximum size of each second-level page table is limited such that it fits in a *single* page frame
- There are no permission or other extra bits.

Your job is to draw and label a figure showing how a virtual address gets mapped into a real address. You should list how the various fields of each address are interpreted, including the size in bits of each field, the maximum possible number of entries each table holds, and the maximum possible size in bytes for each table (in bytes). Also, your answer should indicate where checks are made for faults (e.g., invalid addresses).

b. (6 points) To improve the HAL 2010's performance, you decide to add a TLB to the HAL 2010's virtual memory architecture. The TLB will be 4-way set associative and have 1024 rows. Draw and label a diagram of the TLB, showing the size of each field in the TLB. Indicate how bits of the virtual address are used as input to the TLB, and describe the outputs from the TLB. Include a read enable bit and write enable bit for each page in your TLB.

c. (3 points) If your Page Table Entries contain R/W enable bits, why do they also need to be included in the TLB?

d. (3 points) Your manager is concerned that the page tables for the HAL 2010 memory system might become very large, especially if they are sparsely populated. Describe an alternate technique for the virtual memory system to more efficiently store virtual to physical address mappings, and explain how this technique would improve the storage efficiency of the page tables.

No Credit – **Problem X** (000000000000 points)

Degrees of Matriculation

(April 1, 2008, adapted from NY Times Op-Ed Contributor ANDY BOROWITZ)

FROM: Office of Admissions, Stanford University

To: Double Legacy Applicant, Class of 2012

Here at Stanford's Office of Admissions, we have some very exciting news for you. While your SAT scores and grade point average fall below the threshold for acceptance to Stanford's class of 2012, your Stanford parents' dogged participation in our annual fund-raising appeals — including their generous contributions to Stanford's recombinant DNA lab and IMAX theater — have gained you admission to a unique new program called LegacyPlus™.

With LegacyPlus™, you, the Stanford double legacy, will enjoy all of the perks of students who actually got into Stanford — except for the education part.

As a LegacyPlus™ attendee, you will live on the Stanford campus for the next four years in special LegacyPlus™ housing in historic East Palo Alto. You will walk, talk and — on four special "Common Nights" a year — dine with real Stanford undergraduates.

What will you and your fellow LegacyPlus™ enrollees be doing while the actual Stanford Class of 2012 is going to lectures, choosing majors and taking exams? A better question would be, "What won't you be doing?" With a jam-packed schedule of specially organized LegacyPlus™ pub crawls, you won't have time to think about all of those classes you won't be taking.

And the benefits of LegacyPlus™ don't end in June 2012, when, in a ceremony scheduled to coincide with Stanford commencement, you will receive an authentic-looking Stanford College Certificate of Participation™. This diploma-like document, produced by an expensive laser printer, allows you to tap into Stanford's "old boy network" of jobs, fellowships and government positions that have been turned down by actual Stanford graduates.

Some of the additional perks of LegacyPlus™ include:

- Official Stanford TuitionBill™ identical to those paid by parents of the Stanford Class of 2012.
- Permission to root for the Stanford football, hockey and lacrosse teams; complimentary tickets to swim meets.
- 15 percent discount on LegacyPlus™ Stanford insignia items, including a car window sticker reading, "My Child Is at Stanford."
- Special LegacyPlus™ "Don't ask, don't tell" policy regarding your use of the words "Stanford College" on your résumé.
- Invitations to LegacyPlus™ class reunions, including such events as the "Big Game" attended by the actual Stanford Class of 2012, which you will view from your hotel via closed-circuit TV.
- And finally, the greatest privilege of all: a lifelong relationship with Stanford's dedicated network of fund-raising cold-callers.

So join us, won't you? Take advantage of this rare opportunity by wiring funds today. (Prices double after May 1.)

LegacyPlus™: The Class, Without the Classes.

3. (23 points total) Filesystems.

a. (12 points) For each of the following file descriptor data structures, describe: how efficiently this structure handles sequential access of large files, and why.

i) Contiguous Allocation:

ii) Linked Allocation:

iii) UNIX 4.2BSD inodes:

b. (11 points) UNIX filesystems.

Consider a UNIX filesystem with the following components:

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

i) (6 points) How much disk space, including metadata and data blocks, is needed to store a 4 GB DVD image file? Show your calculation for partial credit.

ii) (5 points) Assuming that there is not a buffer cache, how many disk reads will be required to read *only* the last byte in this file? To write it? Show your calculations for partial credit.

4. (25 points total) CPU Scheduling. Here is a table of processes and their associated arrival and running times.

Process ID	**Arrival Time**	**Expected CPU Running Time**
Process 1	0	4
Process 2	2	3
Process 3	4	5
Process 4	6	2

a. (9 points) Show the scheduling order for these processes under First-In-First-Out (FIFO), Shortest-Job First (SJF), and Round-Robin (RR) with a quantum = 1 time unit. *Assume that the context switch overhead is 0 and new processes are added to the* ***head*** *of the queue except for FIFO.*

Time	**FIFO**	**SJF**	**RR**
0			
1			
2			
3			
4			
5			
6			
7			
8			
9			
10			
11			
12			
13			
14			

b. (12 points) For each process in each schedule above, indicate the queue wait time and turnaround time (TRT).

Scheduler	Process 1	Process 2	Process 3	Process 4
FIFO queue wait				
FIFO TRT				
SJF queue wait				
SJF TRT				
RR queue wait				
RR TRT				

The queue wait time is the *total* time a thread spends in the wait queue. The turnaround time is defined as the time a process takes to complete after it arrives.

c. (4 points) A FIFO page replacement algorithm replaces the page that was first referenced the longest time ago. Under a workload with temporal reference locality, choosing to replace a page accessed a long time ago seems like a good idea, since it is not likely to be within the current locality being accessed. Does this mean that a FIFO replacement algorithm should approximate a LRU algorithm for workloads with strong locality? Justify your answer with an example.

This page intentionally left blank