Version D

Exam #2

CSE 3320.002

Spring 2015

Name: _____

UTA ID:_____

"I certify that the following work is my work alone and I will follow the highest standards of integrity and uphold the spirit of the Honor Code"

Signature:_____

Directions: This is a closed book, closed notes exam. You may use a hand written 8.5 x 11 sheet of paper with notes. Please answer the questions briefly. Complete sentences are not necessary. Write your answers legibly. Unreadable answers will be counted wrong. You may write on back if needed. A table of the powers of 2 is available on the last page of the test.

1. (10pts.) Describe memory paging. Why do we use it? What are its benefits?

2. (10pts) Explain the three different disk allocation schemes we discussed. Make sure to explain if they suffer from external fragmentation and if they support random access.

3. (10pt) Given a translation lookaside buffer lookup time of 5 nanoseconds. Assuming a memory cycle time of 100 nanoseconds. The TLB hit ratio is 90%. What is the effective access time?

4. (10pts.)In a virtual memory environment with 4 GB addressable space, where pages are 1KB bytes in size:
   1. How many entries are in the page table (maximum)?
   2. How would 32-bit addresses be used (how many page bits, how many offset bits)?

5. (10pts.) Give a page table of 65,536 entries and each table being 1024 bytes in size, what is the maximum addressable memory of this table?

6. (10pts.) Given a file system that uses inodes to represent files. Disk blocks are 4096 KB in size, and a pointer to a disk block requires 4 bytes. This file system's index nodes have 12 direct disk blocks, as well as a single second level indirect disk block. What is the largest file that can be held using this in inode layout?

7. (10pts.) Given a page request reference string of D C A B C D E F A B D and a page table size of three, calculate how many page faults will occur with the optimal page replacement algorithm. If all pages are equally replaceable pick the first available.

8. (10pts.) Given a page request reference string of D C A B C D E F A B D and a page table size of three, calculate how many page faults will occur with the FIFO page replacement algorithm. If all pages are equally replaceable pick the first available.

9. (10pts.) Describe internal fragmentation and external fragmentation.

10. (10pts) What is a race condition? How do you guard against it?

Extra Credit

1. (5 pts) In the inode structure of the ext2 file system there are pointers to the first 12 direct blocks. There is a pointer to an indirect block, a pointer to a doubly indirect block and a pointer to a trebly indirect block. Given a block size of 2048 bytes, what is the maximum file size allowed by the file system.

Extra Credit

2. (5 pts) The following code suffers from performance degradation due to excessive page faults. How would you restructure it to reduce the number of page faults? Assume a page table entry is 4096 bytes and integers are 32 bit.

```
#define MAX_GRADES 1024
struct {
  int  id;
  int  homework_average;
  int  quiz_average;
  int  exam_average;
  int  grades[MAX_GRADES];
} gradebook;

int main()
{
  gradebook 3320_gradebook[1024];

  int class_quiz_average;
  int class_exam_average;
  int class_homework_average;

  for( int i = 0; i < 1024; ++i )
  {
    class_quiz_average     += 3320_gradebook[i].quiz_average;
    class_homework_average += 3320_gradebook[i].homework_average;
    class_exam_average     += 3320_gradebook[i].exam_average;
  }

  class_quiz_average     = class_quiz_average / 1024;
  class_homework_average = class_homework_average / 1024;
  class_exam_average     = class_exam_average / 1024;

  return 0;
};
```

Extra Credit 2 workspace if needed.

n	2^n	n	2^n	n	2^n
0	1	11	2,048	22	4,194,304
1	2	12	4,096	23	8,388,608
2	4	13	8,192	24	16,777,216
3	8	14	16,384	25	33,554,432
4	16	15	32,768	26	67,108,864
5	32	16	65,536	27	134,217,728
6	64	17	131,072	28	268,435,456
7	128	18	262,144	29	536,870,912
8	256	19	524,288	30	1,073,741,824
9	512	20	1,048,576	31	2,147,483,648
10	1,024	21	2,097,152	32	4,254,967,296