Name: KEY

Operating Systems
Fall 2014
Test 1
October 1, 2014

Closed books, notes, cell phones, PDAs, iPods, laptops, etc. No headphones, please. No calculator needed.

You have 75 minutes to solve 7 problems. You get 10 points for writing your name on the top of this page. As with any exam, you should read through the questions first and start with those that you are most comfortable with. If you believe that you cannot answer a question without making some assumptions, state those assumptions in your answer.

Partial credit will be offered only for meaningful progress towards solving the problems.

Please read and sign below if you agree with the following statement:

In recognition of and in the spirit of the academic code of honor, I certify that I will neither give nor receive unpermitted aid on this exam.

Signature:________________________________

0	/10
1	/5
2	/15
3	/15
4	/15
5	/10
6	/15
7	/15
Total	**/100**

1. **[5 points, 1 each] True or False?**

   a. In a multiprogramming system the main memory is not generally occupied simultaneously by multiple processes.

   False

   b. A hardware mechanism is needed for translating virtual addresses to physical main memory addresses at the time of execution of the instruction that contains the reference.

   True

   c. A physical address is the location of a word relative to the beginning of the program and the processor translates that into a virtual address.

   False

   d. All segments of all processes must be of the same length.

   False

   e. It is impossible to have both paging and segmentation in the same system.

   False

2. **[15 points, 3 each] Short attention span:** Answer the following questions:

   a. What is the difference between a virtual address and a physical address?

VA is an address in a logical space (an address in an illusion) while the physical address is an address in the actual physical memory.

   b. Which of the following instructions should be privileged?
      i. Read the clock
      ii. Set value of timer
      iii. Turn off interrupts
      iv. Switch from user to kernel mode
      v. Clear memory

   c. Give an example of a preemptive scheduling algorithm.

Round Robin

   d. True or False: A memory system that uses paging is vulnerable to external fragmentation. Why or why not?

False. There is no external fragmentation in a paging system because there is no space left outside a page/frame.

   e. Of the following items, which are stored in the process control block?
      i. Page table pointer
      ii. ~~Page table~~
      iii. Stack pointer
      iv. ~~Segment table~~
      v. ~~List of processes in the ready state~~
      vi. The content of CPU registers
      vii. Program counter

3. **[15 points]** Consider the following program:

```
main (int argc, char ** argv){
     int child = fork();
     int x = 5;

     if (child == 0) {
          x+=5;
     }
     else {
          child = fork();
          x+=10;
          if (child){
               x+=5;
          }
     }
}
```

How many different copies of the variable x are there? What are their values when their process finishes?

3 copies of values 10, 15 and 20.

4. **[15 points]** For each of the following processor scheduling policies, describe the set of workloads under which that policy is optimal in terms of minimizing average response time and the set of workloads under which the policy performs the worst. If there are no workloads under which a policy is optimal or worst, indicate that.
    a. FIFO
    b. Round Robin
    c. Multilevel Feedback Queue

a) FIFO:
    a. optimal: shortest jobs first or short few jobs
    b. pessimal: longest job first or many long jobs

b) Round Robin:
    a. optimal: few jobs in the workload;
    b. pessimal: very many jobs arriving at once;

c) MLFQ:
    a. optimal: few jobs in the highest queue, with new jobs arriving not very many at the same time;
    b. pessimal: the highest priority queue already contains very many jobs (for example, I/O bound) and many new jobs (of any type) arrive at once – they will queue at the end of the highest priority queue and give a long average response time.

5. **[10 points]** A computer whose processes have 1024 pages in their address space keeps its page tables in memory. The overhead required for reading a word from the page table is 500nsec. To reduce this overhead, the computer has TLB, which holds 32 (virtual page, physical page frame) pairs, and can do a lookup in 100nsec. What hit rate is needed to reduce the mean overhead to 200 nsec?

Mean Overhead = HR*100ns + (1-HR)(100ns+500ns) = 600ns – HR*500ns = 200ns =>
HR = 400/500 = 0.8

6. **[15 points]** Consider a simple paging system with the following parameters: 2^{32} bytes of physical memory; page size of 2^{10} bytes; 2^{16} pages of virtual address space.

   a. How many bits are in a virtual address? [2 points]
   b. How many bytes in a frame? [2 points]
   c. How many bits in the physical address specify the frame? [2 points]
   d. How many entries in the page table? [2 points]
   e. How many bits in each page table entry? Assume each page entry includes a valid/invalid bit and padding bits to make its size a power of 2. [2 points]
   f. What is the effect on the page table if the physical memory space is reduced by half? [5 points]

a) 26 (10+16)
b) 2^10
c) 32-10=22
d) 2^16 entries
e) 22 bits for specifying the frame + 1 = 23; 32 bits = 4 bytes (but 3 bytes should work, too, as it needs to be an integer number of bytes rather than a power of 2...)
f) Physical memory reduced by half => 31 bits to represent a physical address => 31-10=21 bits to represent the frame number. In theory, a page table entry will need one less bit. In practice, there might be no change, since the size of the page table entry will be an integer number of bytes. The number of entries in the page table does not change.

7. **[15 points]** Write the binary translation of the logical address 0001010010111010 under the following hypothetical memory management schemes, and explain your answer.
   a. A paging system with a 256-address page size, using a page table in which the frame number happens to be four times smaller than the page number.
   b. A segmentation system with a 1K-address maximum segment size, using a segment table in which bases happen to be regularly placed at real addresses: 22 + 4,096 x segment number.

a)
0001010010111010
256 page size => offset represented on 8 bits => offset is 10111010, page # is 00010100
Frame number is 4 times smaller than the page number => frame # is 101 (we don't know how many 0s at the beginning, but it doesn't matter).

Thus, the physical address is **10110111010 (in decimal: 1466)**

b)
1K-address maximum segment size => offset needs 10 bits => offset is 0010111010
Segment is 000101, which is 5
Base address for segment 5 is 22 + 4096x5 = 20502
Offset in decimal is: 186

Thus, the physical address is 20502 + 186 = **20688 (**in decimal**)** or **101000011010000** in binary