Operating System Concepts, Midterm

November 18, 2013

1. (12%) Briefly explain the definition of following terminologies, and do a simple comparison in terms of their strength and weakness: (4 pt. each)
   (a) Layered OS structure vs. Microkernel
   (b) Message passing communication vs. shared memory communication
   (c) Compile-time address binding vs. Runtime address binding

2. (5%) Use a simple diagram to illustrate the key steps for handling an interrupt.

3. (4%) Briefly explain why context switch will cause system performance degradation in terms of **program execution time** and **memory access time**.

4. (5%) How many processes are created in the following program? (You must plot the process tree with process ID to explain your answer. You can assume the process ID is assigned in increasing order from 0.)

```
int main() {
  int pid = fork();
  if (pid==0) {
       for (int i=0; i<2; i++) {
             pid = fork();
       }
    execlp("\bin\ls");
  } else {
       pid = fork();
  }
    if (pid == 0) fork();
}
```

5. (7%) a. (3%) Explain what is many-to-one multithreading model? b. (4%) Give two examples in which multithreading does NOT provide better performance than a single-threaded solution.

6. (5%) Briefly explain why the modern general purpose OS uses both segmentation and paging to manage physical memory?

7. (5%) a. (3%) Explain what is a TLB (Translation Lookaside Buffer)? b. (2%) Why TLB is often implemented by associative memory?

8. (10%) a. (4%) Use an example to explain inverted page table mechanism. b. (6%) Discuss the difference between inverted page table and the traditional single-level page table in terms of their memory access time, page table size, and page sharing mechanism.

9. (8%) a.(3%) What is a page fault? b. (5%) Describe the steps to handle a page fault.

10. (12%) Consider a computer with 3 memory frames to handle a reference string "2, 5, 3, 4, 2, 3, 1, 3, 4, 1". Show the step-by-step reference result and indicate the which reference will cause page faults using the following page replacement algorithm: (4 pt. each)
    (a) FIFO (b) Optimal (c) LRU

11. (10%) a. (5%) Describe the steps that would cause trashing. b. (5%) Explain how to use working-set solution to solve the problem.

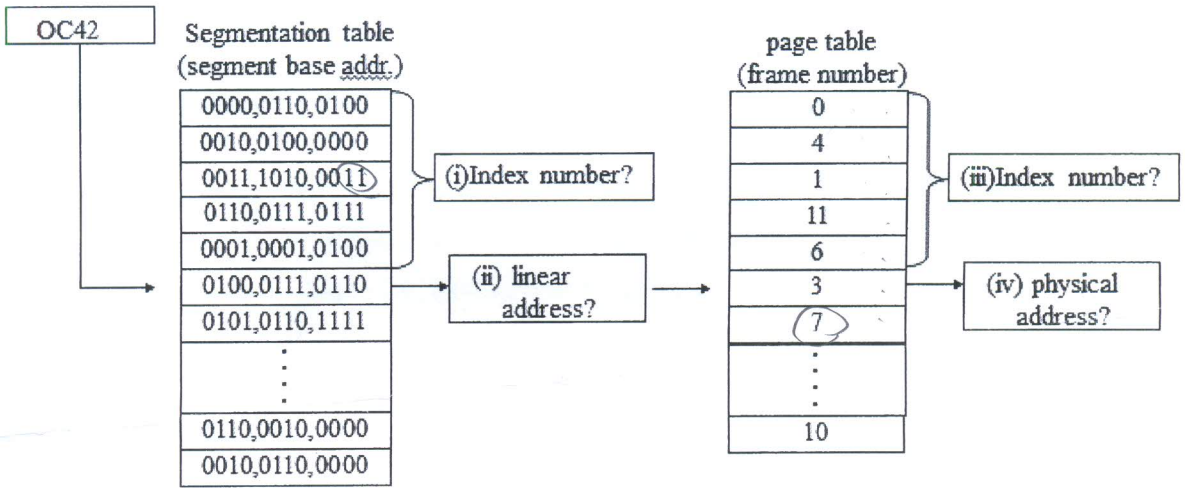
12. (5%) Consider the 2-dim array A: **int A[][] = new int [50][50]**; where **A[0][0] is stored at logical address location 400**, in a paged memory system with **pages of size 200 bytes**. Each integer takes 2 bytes and the array is allocated in row-major order "addr(A[i][j]) = addr(A[0][0]) + (i*50+j) * 2". A small process resides in **page 0** (locations 0 to 199) for manipulating the A matrix. Assume the system only has **3 available frames**. If using **pure demanding paging**. How many page faults are generated by the following code, using LRU replacement? **You must explain which memory reference or operation will cause each of the page faults.**

```
for (int i = 0; i < 50; i++)
    for (int j = 0; j < 50; j++)
        A[i][j] = 1;
```

13. (12%) Consider a byte-addressable computer system with a 16-bit virtual address, total physical memory size 4KB, page size is 256 Bytes, the maximum size of a segment is 1KB. Given the following segment table, page table and a 16 bits hexadecimal logical address "0C42", complete the address translation diagram below.

    (i) (2%) segment table index; (ii) (4%) linear address (i.e. in binary form);

    (iii) (2%) page table index; (iv) (4%) physical address (i.e. in binary form);



1110

1101