

# Homework 2 – System Programming

## CS241, Spring 2007

**Deadline: Wednesday 05/02/2007, 4pm, 3120 SC**

**Submission Steps:**

- (1) YOUR NETID MUST BE AT THE TOP OF EVERY PAGE;**
- (2) PROVIDE THE SOLUTION IN HARD COPY – DO NOT USE HANDWRITING;**
- (3) START EACH OF THE FOUR PROBLEMS ON A NEW PAGE – IT WILL MAKE THE GRADING PROCESS MUCH FASTER!!**
- (3) BRING THE HARD COPY HW2 SOLUTION TO ANDA OHLSSON'S OFFICE, 3120 SIEBEL CENTER BY WEDNESDAY, May 2, 4PM**

<b>Problem</b>	<b>Points</b>	<b>Grader</b>
<b>Problem 1</b>	/25	
<b>Problem 2</b>	/25	
<b>Problem 3</b>	/25	
<b>Problem 4</b>	/25	
<b>Total</b>	<b>/100</b>	

## 1. Problem on Input/Output (25 Points)

**1A. (9 Points)** Although DMA does not use the CPU, the maximum transfer rate is still limited. Consider reading a block from the disk. Name three factors that might ultimately limit the rate transfer.

**1B. (6 Points)** An alternative to interrupts is polling. What are the circumstances in which polling is better than interrupts? Give at least two circumstances.

**1C. (10Points)** Let us assume a disk with rotational speed of 15,000 rpm, 512 bytes per sector, 400 sectors per track and 1000 tracks on the disk, average seek time is 4ms. We want to transmit a file of size 1 MByte, that is stored contiguously on the disk. What is the transfer time for this file? What is the average access time for this file? What is the rotational delay in this case? What is the total time to read 1 sector? What is the total time to read 1 track?

## 2. Problem on Basic Memory Management (25 Points)

**2A. (13 Points)** In this problem you are to compare the storage needed to keep track of free memory using a bitmap versus using linked list. The 128 MB memory is allocated in units of  $n$  bytes. For the linked list, let us assume that memory consists of an alternative sequence of filled fragments and empty holes, each 128 KB. Also assume that each node in the linked list needs 32 bit memory address, a 16-bit length, and a 16-bit next-node field. How many bytes of storage are required for each method? Which one uses less storage space?

**2B. (12 Points)** Let us consider the swapping system in which memory consists of the following hole sizes in memory order: 10KB, 5KB, 18KB, 6KB, 7KB, 12KB, 15KB, 20KB. Which hole is taken for successive fragment requests of (a) 18KB, (b) 5KB, (c) 18KB for the following policies: first fit, best fit, and worst fit? Specify clearly what is the allocation of the fragment requests (a), (b) and (c) for each policy.

### 3. Problem on Virtual Memory (25 Points)

**3A. (15 Points)** A process has four page frames allocated to it (All the following numbers are decimal, and everything is numbered starting from zero). The time of the last loading of a page into each page frame, the time of last access to the page in each page frame, the virtual page number in each page frame, and the referenced (R) and modified (M) bits for each page frame are as shown in the table below (the times are in clock ticks from the process start at time 0 to the event – not the number of ticks since the event to the present).

Virtual page number	Page frame	Time loaded	Time referenced	R bit	M Bit
2	0	60	161	0	1
1	1	130	160	1	0
0	2	26	162	1	0
3	3	20	163	1	1

A page fault to virtual page 4 has occurred at time 164. Which page frame will have its contents replaced for each of the following memory management policies? Explain why in each case:

(a) FIFO

(b) LRU

(c) Optimal (Use the following virtual page reference string before the page fault to virtual page 4 occurred : 4,0,0,0,2,4,2,1,0,3,2)

(d) How many page faults would occur if the working set policy with LRU were used with a window size of 4 (use the future virtual page reference string, starting before the page fault to virtual page 4 occur 4,0,0,0,2,4,2,1,0,3,2). Start counting page faults with the page fault of page 4.

**3B. (10 Points)** Consider the following page table below. All numbers are decimal, everything is numbered starting from zero, and all addresses are memory byte addresses. The page size is 1024 bytes and the processor architecture is 32-bit.

Virtual page #	Valid Bit	Reference Bit	Dirty Bit	Page frame #
0	1	1	0	4
1	1	0	1	7
2	0	0	0	-
3	1	0	0	2
4	0	0	0	-
5	1	0	1	0

a. (5 Points) Describe exactly how, in general, a virtual address generated by the CPU is translated into a physical main memory address? Your answer should explain how the Valid, Reference and Dirty bits are used or set.

b. (5 Points) What physical address, if any, would each of the following virtual addresses correspond to? (do not try to handle any page faults if any). Note that the addresses are decimal addresses, not hexadecimal. addresses

- i. 1052
- ii. 2221
- iii. 5499

#### 4. Problem on File Systems (25 Points)

Consider the command line `cat /var/log/log.0 /var/log/log.1 >> /var/log/mystuff/biglog`

**4A. (3 Points)** Briefly explain in English how the 'cat' command and '>>' redirection are used and their actions on the three files.

**4B. (10 Points)** Calculate the precise amount of time (to the nearest 1/100th second) required to perform the above command. Your calculation should consider - I/O required to traverse directory and inode structure of each file; creating additional inodes and indirect blocks of inode pointers; reading the contents of the files; writing the contents of the files; updating last accessed time information on all inodes.

Assume the following - `/var` is a mount point of a local hard disk with a block size of 4096 bytes. The files `'log.0'`, `'log.1'` and `'biglog'` are initially  $2^{20}$ ,  $2^{20}$  and  $2^{10}$  bytes respectively. A pointer to a disk block requires 4 bytes. An inode has 10 direct entries, a single indirect, a double indirect and a triple indirect entry.

Only the `'cat'` program and directory entries (names & inode pointers) of `/var/` are initially cached in memory. All other information requires a disk read. Assume that caching is maximal, i.e., the least number of disk reads is performed (if there is an in memory representation the data is not re-read from disk) and no errors occur. The reading or writing time for a single disk block is always 0.01 seconds (i.e., reading then writing the same block requires 0.02s).

**4C. (12 Points)** Let us consider the following shell commands on a local inode-based file system.

```
cd /tmp
mkdir subdir
echo "hello" > subdir/file1
ln -s subdir/file1 file2
ln -s subdir/file1 file3
ln subdir/file1 file4
ln file4 file5
chmod 744 file4
touch file3
rm file5
chmod 000 subdir
```

- a) How many disk blocks were allocated in total?
- b) How many disk blocks were freed in total?
- c) Which command(s) incremented or decremented an existing inode reference count?
- d) Describe in words the permissions on the file *'file4'*
- e) Identify the last command to modify the inode of *'file3'* and describe which inode field is modified.
- f) Describe the indirect effect (if any) of the *'chmod 000 .'* command on the files *file1,file2,file3,file4*.