89%

CS570 Summer 2017 Exam #2

6/14/2017

Name [redacted]

RedID [redacted]

1. 6/6 2: 4/4 3: 4/4 4: 6/6 5: 4/4 6: 9/9 7: 4/4 8: 6/6 9: 0.5/4 10: 1/3 = 44.5/50

1. Name the six file types implemented in Unix. Give a description/example of each:
   1. Regular : Contains user data & more 1 block
   2. Directory : System files that manage other system files & any file with children
   3. Pipe file : files are connected via pipe; don't have children & only one block
   4. Link : link other files together, no children, one block
   5. device driver : files that are device drivers and no user info
   6. Socket : do not have a user info only one block

2. Given an O/S which has divided all of fast memory into 3 frames, and if this O/S uses the Clock algorithm for page replacement/swapping, then how many page faults are generated if the following sequence of pages are requested for use (show your work on back!): 8

2	5	2	3	5	4	2	5	1	2	3	2
2	2	2	2	2	2	2	2	2	2	2	
	5	5	5	5	5	5	5	1	1	1	
			3	3	4	4	4	4	4	3	
X	X		X		X		X		X		

Look at the back page

Not solution

3. Repeat #2, but use the LRU algorithm for page replacement/swapping instead. How many page faults? 7

2	5	2	3	5	4	2	5	1	2	3	2
2	2	2	2	2	4	4	4	1	1	1	1
	5	5	5	5	5	5	5	5	5	3	3
			3	3	3	2	2	2	2	2	2
X	X		X		X	X		X		X	

4. List three advantages of using page-segmented memory over segmented memory. Be specific and explain why it's an advantage:

Less fragmentation: No external fragmentation, the page segments are not broken up

Look up/security: Can refer to the page table instead of memory

Runtime efficiency: Page segments are in memory, so no time is used putting in memory

5. Given the file system architecture we've studied in the text and in lecture, with data blocks of 1024 byte size, how many disk reads would need to be performed to get the inode for the following into memory 8:
   /usr/include/linux/const.h
   note, the only node in memory is the i-node for /

dir /
inode /usr/
dir /usr/
inode /usr/include/
dir /usr/include/
inode /usr/include/linux/
dir /usr/include/linux/
inode /usr/include/linux/const.h

AQ11

Your process needs to access the following three virtual addresses (given in base 10), compute the virtual page number and the offset within that page for each address if page size = 4 KB: 20,092, 45,054, 131,072 **(show work)**

$\frac{20092}{4096} = 4.9 = 4$

$20092 - (4 \times 4096) = 3708$

$\frac{45054}{4096} = 10.9 = 10$

$45054 - (10 \times 4096) = 4094$

$\frac{131072}{4096} = 32$

$131072 - (32 \times 4096) = 0$

7. Circle T for true or F for false for each statement:

(T)/F	At the lowest level of the file system architecture, device drivers communicate directly with peripheral devices or their controllers or channels.
T/(F)	Printer is an example of block oriented I/O device
(T)/F	A memory system employing paging may suffer slightly from internal fragmentation and experiences no external fragmentation.
T/(F)	The concept of virtual memory is based solely on the paging technique, the segmentation technique cannot be used.

8. A certain LINUX filesystem has 512 byte blocks and 4 byte disk addresses. What is the maximum filesize assuming inodes have 10 direct, one single, one double, and one triple indirect addresses in each inode?
(write the equation to solve for the maximum file size, then calculate)

$\frac{512}{4} = 128$ → After finding the value we add it too single, double & triple. Also added to 10 direct

$512 \times (10 + 128 + 128^2 + 128^3) = 1082201088$

9. Fill in the blanks with the correct response:
   a. Main memory suffers from internal fragmentation
   b. In paging systems, the program is loaded into page frame of memory which the operating system will load into page frame in main memory
   c. Most virtual memory schemes make use of a special high-speed cache for page table entries, called a ~~Page Segmented memory~~ TLB
   d. In a segmentation system, each entry in a ~~block~~ contains control bits and the starting address and the length of the segment SEG. TABLES

-3.5

10. Circle the correct response:
   a. The real address of a word in memory is translated from the following portions of a virtual address:
      1. Page number and frame number
      2. Page number and offset
      3. Frame number and offset
      4. None of the above

   b. The real address of a word in memory is translated from the following portions of a virtual address:
      1. TLB miss
      2. TLB hit
      3. Page fault
      4. None of the above

   -1

   c. In a system employing a paging scheme for memory management, wasted space is due to:
      1. External fragmentation
      2. Internal fragmentation
      3. Pages and frames of different specified sizes
      4. None of the above

   -1