（装订线内不要答题）

复旦大学计算机科学技术学院

2012-2013 学年第一学期《操作系统》期末考试试卷

A 卷　　共 7 页　　　　　　2013 年 1 月

课程代码：COMP130008.01-03　　　考试形式：☐开卷　☑闭卷

（本试卷答卷时间为 120 分钟，答案必须写在试卷上，做在草稿纸上无效）

NO aids (books, notes, discussion etc.) are allowed. Ask the instructor if you have any questions.

Student ID____________________Name____________________Score_______________

Question	1	2	3		Sum
Score					

1. Choice (11×3%=33%)

(1) Internal fragmentation does NOT exist in ____B___

A. Fixed Partitioning　　B. Dynamic Partitioning
C. Paging　　D. Combined Paging and Segmentation

(2) In a paging system, when should the physical addresses be generated? ___A_____

A. Run Time　　B. Compile Time
C. Link Time　　D. Load Time

(3) With huge virtual address space, the page table will be too large to be held fully in main memory. Which one of the following mechanisms can be used to solve this problem? D____

A. Translation Lookaside Buffer (TLB)
B. Page Replacement
C. Working-set strategy
D. Multi-Level Page Table

(4) In a demand paging system, it is observed that a page reference stream leads to 8 page faults with 3 frames allocated, but 9 page faults with 4 frames. Which replacement algorithm is possibly used by the system? ___B___

A. Optimal policy　　B. FIFO policy
C. LRU policy　　D. Stack Algorithms

(5) For a given process, the length of its page reference stream is 12, which consists of 6 distinct pages. If demand paging is used and 3 frames (initially all empty) are allocated to this process, what is the lower bound for the number of page faults during the process' execution?______C_______

A. 12 B. 9

C. 6 D. 3

(6) Given the following reference string: 3 5 7 5 0 5 3 0 8 6 0 8 6 0 1 2 5 3 6 5 2 1. What is the working set by the end of the reference string, given the working-set window size being 10.

A_

A. {0 1 2 3 5 6} B. {3 5 7 0 8 6}

C. {3 5 7 0 8 6 1 2} D. {0 3 5 7 8}

(7) Suppose that a file occupies 10 disk blocks. Now we have to read the file, block by block, into a buffer area in main memory. If the buffer area has the same size as a disk block, reading a disk block into the buffer takes 100μs, transferring the data in buffer to a user area takes 50μs, analyzing a data block by CPU takes 50μs, then what is the total time for reading and analyzing the whole file (with single buffering)? ______C______

A 1100μs

B 1500μs

C 1550μs

D 2000μs

(8) Where most likely does Linux associate each device with a (special) file in the /dev directory.? B_____________

A. user program

B. device independent I/O software

C. device driver

D. interrupt handler

(9) Which one of the following statements about allocation methods is NOT true in a file system?____A________________

A. Linked allocation is very fast for both sequential access and direct access

B. Contiguous allocation suffer from external fragmentation

C. Indexed allocation involves extra storage overhead for index table

D. File can be extended easily using linked allocation.

（装订线内不要答题）

(10) If we use a large block size for file system, which of the following statement is NOT true D______.

A. File allocation table is smaller.
B. Fewer blocks must be read for a large file
C. Reduction of seek and rotational delays on average
D. Reduction of internal fragmentation

(11) Which one of the following statements is CORRECT? ______D______.

A. Symbolic linking is faster than hard linking
B. Symbolic linking requires less space than hard linking
C. Hard linking cannot link to a file on a remote machine
D. Removing a symbolic linking does not affect the file at all

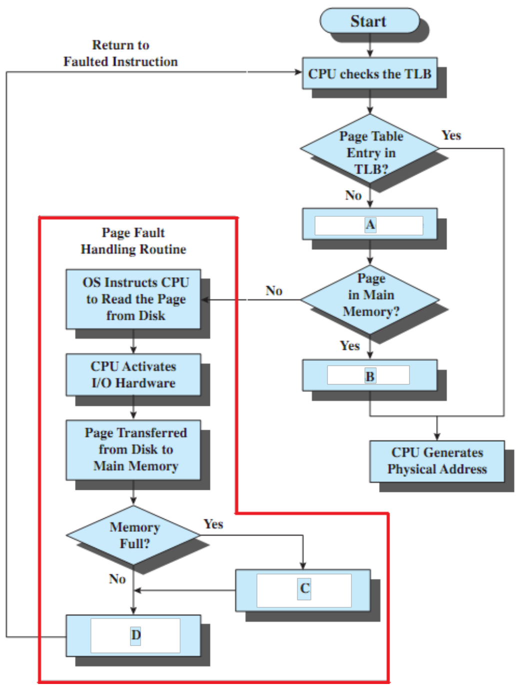
2. Short Answer (6*6%=36%)

1. Some people claim that for indexed file allocation methods with both direct and indirect pointers, doubling the block size will double the maximal file size accordingly. Is that true? Please give an explanation.

Reference Answer:

- It is not true. Since we have indirect blocks, doubling the size of indirect block results in more pointers for rest blocks. (3 points)
- Hence, doubling the block size will increase the file size more than 2 times. (3 points)

2. Below is the flowchart of page faults processing with some blanks to be filled (items A, B, C, and D). Please connect each item (A through D) with one correct task on the right.


Start
Return to Faulted Instruction
CPU checks the TLB
Page Table Entry in TLB?
Yes
No
A
Page in Main Memory?
No
Yes
B
CPU Generates Physical Address
Page Fault Handling Routine
OS Instructs CPU to Read the Page from Disk
CPU Activates I/O Hardware
Page Transferred from Disk to Main Memory
Memory Full?
Yes
No
C
D

- Access Page Table
- Perform Page Replacement
- Page Table Updated
- Update TLB

Reference Answer

A. Access Page Table
B. Update TLB
C. Perform Page Replacement
D. Page Table Updated

(1.5 points each)

3. How many disk operations are needed to fetch the i-node for the file /usr/tom/book.pdf? Assume that the i-node for the root directory is in memory, but nothing else along the path is in memory. Also assume that all directory contents fit in one disk block. Please list all the disk operations.

Reference answer

The following disk reads are needed:

directory for /

i-node for /usr

directory for /usr

i-node for /usr/tom

directory for /usr/tom

i-node for /usr/tom/book.pdf

In total, 6 disk reads are required. (one point for each)

4. Some people claim, "The best way to correct virtual memory thrashing is to increase the number of runnable processes." Please comment on the statement.

Reference Answer:

The claim is wrong. (2 points)

Thrashing results when the working sets of running processes don't fit in memory; you must decrease running processes to free up memory. (4 points)

5. A slight modification of the SCAN algorithm (or the elevator algorithm) for scheduling disk requests is to always scan in the same direction (so-called CSCAN Algorithm). In what respect is this modified algorithm better than original SCAN algorithm? Please explain it with your statistics knowledge.

Reference Answer:

In the worst case, a read/write request is not serviced for almost two full disk scans in the SCAN algorithm, while it is at most one full disk scan in the CSCAN algorithm. (3 points)

In other words, the CSCAN algorithm gives a smaller variance in response time. (3 points)

6. Rather than writing updated files to disk immediately when they are closed, many UNIX systems use a delayed write policy in which dirty disk blocks are flushed to disk once every 30 seconds. List two advantages and one disadvantage of such a scheme:

Reference answer:

- Advantage 1: The disk-scheduling algorithm (i.e. SCAN) has more dirty blocks to work with at any one time and can thus do a better job of scheduling the disk arm. (2 points)
- Advantage 2: Temporary files may be written and deleted before they ever get to disk. (2 points)
- Disadvantage: File data may be lost if the computer crashes before data is

（装订线内不要答题）

written to disk. (2 points)

3. Analysis (31%)

(1) Suppose that we have a 64-bit virtual address split as

Seg. ID	Table ID	Table ID	Table ID	Page ID	Offset
6 bits	11 bits	11 bits	11 bits	11 bits	14 bits

Please answer following questions with calculation or explanation:

a) How big is a page in this system? (1%)

b) How many segments are in this system? (1%)

c) Assume that the page tables are divided into page-sized blocks (so that they can be paged to disk). How much space have we allowed for a PTE in this system? (3%)

d) Show the format of a page table entry, complete with bits required to support the clock algorithm and copy-on-write (COW) optimizations. (4%)

e) Assume that a particular user is given a maximum-sized segment full of data. How much space is taken up by the page tables for this segment? Note: *you should leave this number as sums and products of powers of 2!* (4%)

f) Suppose the system has 16 Gigabytes of DRAM and that we use an inverted page table instead of a forward page table. Also, assume a 14-bit process ID. If we use a minimum- sized page table for this system, how many entries are there in this table? What does a page-table entry look like? (Round to nearest byte, but support clock algorithm and copy on write). (4%)

Reference answer:

a) Since the offset is 14 bits, the page is 2^{14}=16384 bytes or 16KB

b) Since there is a 6-bit segment ID, there are 2^{6}=64 possible segments.

c) Since leaves of page table contain 11 bits to point at pages (the field marked "Page ID"), a 2^{14} bit page must contain 2^{11} PTEs, which means that a PTE is simply 2^{14-11}=8 bytes in size.

d) Need as many bits of physical page address as virtual (non-offset) page address. So, physical page is 50 bits (see below). For clock algorithm, need valid (V), Use (U), Dirty (D). For copy-on-write, need Read-only bit (R). Note that V and D are needed for pretty much any paging algorithm.

Physical Page	V	R	U	D	Other Bits
50 bits	1 bit	1 bit	1 bit	1 bit	10 bits

e) Full page table will be a tree-like structure. Top-page entry (1) will point at 2^{11} page entries, each of which will point at 2^{11} page entries, each of which will point at 2^{11} page entries, each of which finally points at 2^{11} pages. Of course, all of these are a full page in size (2^{14}). So, Answer (just page tables) = $2^{14}\times(1 + 2^{11} + 2^{11}\times2^{11} + 2^{11}\times 2^{11}\times 2^{11}) = 2^{14}\times(1 + 2^{11} + 2^{22} + 2^{33})$

f) A minimum-sized page table requires one entry per physical page. 16GB=2^{34} so # of pages = 2^{34-14}= 2^{20}. Thus, a minimum, we need enough entries to cover one per page, namely 2^{20}. A page table entry needs to have a TAG (for matching the virtual page against) and a PID (to support multiprogramming). Thus, we need at leb this:

PID	VPN	PPN	V	R	U	D
14 bits	50 bits	20 bits	1 bit	1 bit	1 bit	1 bit

(2) In the clock algorithm, if the pointer is moving very frequently, what can we infer from the perspective of working set? (8%)

Answer: The fundamental reason is that the resident set is almost a subset of working set. (8%). Alternatively, the following facts may be inferred. (1) page fault occurs very frequently since the clock replacement algorithm is used frequently. (4%) (2) If the pointer is moving fast, then the program is accessing a large number of pages simultaneously. It is most likely that during the period between the point at which the bit corresponding to a page is cleared and it is checked again, the page is accessed again and therefore cannot be replaced. (4%)

(3) Shortest Seek Time First (SSTF) algorithm selects the disk I/O request that requires the least movement of the disk arm from its current position. Does it guarantee that the average seek time over a number of arm movements will be minimum? If your answer is “yes”, give your reason; otherwise (that is, if your answer is “no”), please construct a counter-example to support your answer. (6%)

Answer: no.
Example 50, 40, 65, 10, 125.