52

COMPUTING SCIENCE DEPARTMENT
CMPUT 379: Operating System Concepts

Wednesday, 26 April 2000

C379 Section B1 Final
Closed book examination:
Two hours (0900-1100)

Note:
There are 6 questions, worth a total of 75 marks.
Closed book examination
Use of a basic calculator is permitted.

Concise, clear answers are expected.

Student ID:

Prepared by: Professor T.A. Marsland (April 2000)

Marks:

Q1	Q2	Q3	Q4	Q5	Q6	Total
10	10	15	15	10	15	75

NOTE * Pages 5 and 10 are BLANK and have been removed for environmental reasons.

1 [10 marks]. Consider the following sequence of memory references from a 460-byte program (assume byte-addressing is used):

10, 110, 224, 161, 305, 458, 202, 130, 160, 289, 400, 222

a. Assuming a page size of 150 bytes, write the reference string.

b. Compute the total number of page faults for this reference string, assuming a memory allocation of 300 bytes and a local LRU algorithm.

c. Compute the total number of page faults for this reference string assuming a memory allocation of 300 bytes and the optimal replacement algorithm.

2. [10 marks]

a. Assume we have a demand-paged memory system with a TLB (Translation Lookaside Buffer) to hold the most active page-table entries. Assume also that the TLB has an access time of B seconds.

If the page table is held in primary memory, and memory access time is T seconds, give an expression for the effective access time to a resident page when P percent of all memory references are resolved by the TLB. What does your expression reduce to as P tends to 1.

b. Suppose we have a demand-paged memory, where the page table is held in registers (assume zero access time) instead of primary memory. The primary memory access time is given as T seconds. Assume it takes V seconds to service a page fault if an empty frame is available (or if the victim page/frame has not been modified), but D seconds if the victim page/frame is dirty (has been modified). Furthermore assume that the page to be replaced is dirty M% of the time.

What is the biggest acceptable page-fault rate, P, for an effective (average) primary memory access time of no more than E seconds?

3 [15 marks].

The Dining Philosopher's Café has N tables. When there are no customers, the waiter cleans the dishes. Potential customers sit at the next empty table, being careful to first interrupt the dish-washing waiter (if necessary).

a. Write pseudo code for the CUSTOMER and WAITER processes, showing how they coordinate their activities. Also specify your WAIT and FREE (P and V, UP and DOWN) primitives in enough detail to show you understand their important role.
b. Show clearly how you handle the case of no empty table for an arriving customer.
c. Carefully define the types of all variables used, and give their initial values.
d. If the Café had M waiters, what changes are necessary to your program.

4 [15 marks]

In Unix, draw a diagram to show how the per process File Descriptor Table (FDT), the system wide Open File Table (OFT) and the "in-memory" I-node Table are linked to each other and to the open files. For each table say a few words about the typical contents of an entry. It is not necessary to identify the full contents of each entry type, but just to identify important components that explain why a different table is necessary. It is vital that your diagram shows not only how parent and child processes may share an output file (the display screen), but also how two independent processes may read the same file. Clearly indicate where the file read/write pointers are kept.

5 [10 marks].

a. What are the advantages and disadvantages of a memory management system using multi-level page tables?

Suppose we have the following form for a virtual address in a two-level page table:

P1	P2	offset
3 bits	5 bits	10 bits

b. What is the page size?

c. What is the maximum size of the logical address space for a program in this system?

d. How much table space will be required for this scheme, given that a page table entry requires 3 bytes of storage?

e. State briefly the advantages and disadvantages of having a large page size versus having a small page size in a demand paging system.

6 [15 marks]. Consider the following small file system. It is stored on a disk consisting of 1 platter (1 side only) with 10 tracks (labelled 0 to 9) each holding 5 blocks (labelled 0 to 4). The block size is 1024 bytes and blocks are numbered from 0 to 49. Assume that all the (non-data) file system overhead, such as the device directory and allocation information, is stored in the first two blocks. There are currently 5 files (F1, F2, F3, F4, F5) in this system. A sixth file, F6, requiring 6000 bytes of storage is about to be created.

a. Define internal and external fragmentation as it applies to file systems in general.

A File Allocation Table (FAT) is used to hold information about file storage. A disk directory gives the first and last block for each file in the system. The current contents of the FAT and disk directory (showing the storage of files F1 to F5) are given on the next page. Note that a "free" block is identified by a -1 entry in this FAT.

File Allocation Table:

Block	Link	Block	Link	Block	Link
2	-1	18	EOF	34	30
3	-1	19	23	35	45
4	34	20	-1	36	-1
5	-1	21	-1	37	16
6	22	22	EOF	38	-1
7	-1	23	24	39	-1
8	9	24	44	40	-1
9	EOF	25	-1	41	-1
10	-1	26	-1	42	-1
11	EOF	27	EOF	43	-1
12	-1	28	13	44	11
13	8	29	-1	45	28
14	27	30	18	46	-1
15	-1	31	19	47	-1
16	6	32	-1	48	-1
17	31	33	-1	49	-1

Disk Directory:

File_Name	First_Block	Last_Block
F1	17	11
F2	35	9
F3	14	27
F4	4	18
F5	37	22

b. Describe where external and internal fragmentation occurs in this scenario.

c. How many disk accesses would be needed to obtain the third block of file F2? State and justify your assumptions.

d. What will happen when File F6 requests allocation of space from the file system?

e. Why does the disk directory store information about the last block of each file?