V1

May 9, 1996

cs450 - Operating Systems 1
Spring 1996

Final Exam
open books, open notes

Starts: **6:20 pm** Ends**: 8:00 pm**

Name:____________________________________(please print)

ID:________________________

Problem	Max points	Your mark	Comments
1	20		10 + 5 + 5
2	30		5 + 5 + 10 + 10
3	20		15 + 5
4	30		10 + 10 + 10
	100		

What happens with this paper (mark one): **Discard**

Mail at the following address:

1. Consider the following resource-allocation policy. Requests for and releases of resources are allowed at any time. If a request for resources can not be satisfied because resources are not available, then check blocked processes waiting for resources: if they have the required resources, then the needed resources are taken away and given to the requesting process. The vector of resources for which the blocked process is waiting is updated to include the resources that have been taken away.

a) Assume that the vector of available resources is A=(4, 2, 2), and the following sequence of requests (in this order):

Process	Requests	Available	Comments
0	(2, 2, 1)		
1	(1, 0, 1)		
0	(0, 0, 1)		
2	(2, 0, 0)		

In the ‘Comments’ column indicate whether the request is granted or not and whether the process is blocked or not.

b) Can deadlock happen? If yes, then give an example. If not, then explain why.

c) Can starvation happen?

2. A system with a 32 bit virtual address has a physical memory of 32 MB and uses demand paging with pages of 2 KB.

© Virgil Bistriceanu, 1996

a) Show the size of various fields in the virtual address.

b) What is the size of the page table? You may assume that each entry in the table has a size which is a multiple of byte; each entry uses four bits for various purposes (the ‘Valid’ bit, ‘Dirty’, etc.).

c) Given the following list of references (addresses issued by the CPU), indicate which will generate page-faults: initially only the page which contains the reference `0x400000` is in the memory.

Reference	Page fault?	
`0x400020`		
`0x400024`		
`0x7fffffe0`		
`0x400028`		
`0x401000`		
`0x10000004`		
`0x40002c`		
`0x401800`		

© Virgil Bistriceanu, 1996

d) Assume that the system also has a TLB with a miss rate of 95%. The memory access time for a reference that hits in the TLB is 20 ns on the average. If the address translation uses the page table stored in the memory, then it takes 200 ns for a memory reference (this time includes the translation time). We want to keep the effective memory access time under 50 ns. What is an acceptable page-fault rate for this? You may assume that replacing a ‘dirty’ page takes 50 ms while replacing a ‘clean’ page only takes 20 ms; in 30% of cases the page to be replaced is dirty.

3. A contiguous file system has a 100 MB disk (this is very little by today’s standards). Let us assume that the system allows two files to be open for writing at one time. Of course the system has no idea how large a file will be when it opens it. Therefore, to open two files, the system will allocate half of the largest free disk space (‘hole’) to the first file and the largest remaining hole to the second file.

From time to time the disk needs to be ‘crunched’, which means that scattered groups of free blocks (holes) are put together in a single, contiguous unit.

a) How many of the following operations can be completed before the first crunch is required? You may assume that in each case the first file is closed before the third one is

© Virgil Bistriceanu, 1996

opened.

	Operation	
1	create file A and write 15MB	
2	create file B and write 5 MB	
3	create file C and write 10 MB	
4	create file D and write 30MB	
5	delete file C	
6	create file E and write 10 MB	
7	delete file A	
8	create file F and write 5 MB	
9	create file G and write 35 MB	
10	delete file D	

You can use the ruled sheet attached with the exam to save some drawing effort.

b) Show the contents of the disk (assumed as a linear array of data) after the list of operations is complete.

c) Assume that only one file can be open at any time. Show the contents of the disk (assumed as a linear array of data) after the list of operations is complete if the allocation uses the ‘best fit’ policy. If crunching is needed then do it.

© Virgil Bistriceanu, 1996

4. A simple UNIX file system has a block size of 512 bytes, and each block number is 4 bytes. The first 10 entries (from 0 to 9) in the i-node are direct pointers to disk blocks. Entry number 11 is a pointer to a block (‘single indirect block’) of 7 pointers to disk blocks. Entry number 12 points to a ‘double indirect block’ which contains 7 pointers to blocks of pointers (seven in each) to disk blocks. Finally, entry number 12 in the i-node is a pointer to a ‘triple indirect block’.

a) What is the maximum file size (in bytes or multiple of byte)?

Suppose that the i-node structure for the file *myfile* contains the following block list (numbers are in decimal and the leftmost number is entry 0):

```
50 51 52 53 77 78 79 80 99 31 32 96 81
```

and the contents of some of these blocks are given in the following table:

Block #	Contents
31	27 28 29 ...
32	65 97 24 ...
80	44 45 46 ...
81	89 92 91 ...
89	94 95 88 ...
90	41 42 43 ...
91	57 58 59 ...
92	60 61 62 ...
93	67 68 69 ...
94	36 37 39 ...
95	47 48 49 ...
96	93 98 90 ...
97	83 84 85 ...
98	72 73 74 ...

© Virgil Bistriceanu, 1996

Which disk blocks must be read for the following operations? Answer both in logical blocks and in physical blocks.

b) Offset 2000 in *myfile*: read 1024 bytes:

c) Offset 70656 in *myfile*: read 1024 bytes:

© Virgil Bistriceanu, 1996

© Virgil Bistriceanu, 1996