THE UNIVERSITY OF AUCKLAND
NEW ZEALAND

COMPSCI 340 S2 C 2015
SOFTENG 370 S2 C 2015
Operating Systems
Assignment 3 – Files and Memory
6% of course mark
Due Friday 16th of October 9:30 pm

Questions

Answer the following questions. You can use any program to create the file with your answers but you MUST export the file into a PDF version for submission. The submitted file should be called `a3.pdf`.

1. For these questions consider a Unix like file system as on page 11 of lecture 18 where the number of direct blocks is 16. Each block is 4KiB and a block number consists of 4 bytes.

   For all questions show your working. If you write a program to help calculate these answers you may submit this program. Ensure that the program is clear with good comments (and your login name).

   Give the answers from a) to d) in decimal bytes e.g. 65,536 rather than 2^16.

   (a) What is the maximum size for a file which does not use an indirect block?

   (b) What is the maximum size for a file which uses the single indirect block? (not the double or triple).

   (c) What is the maximum size for a file which uses the double indirect block? (not the triple).

   (d) What is the maximum size for a file in this system?

   (e) How many levels of indirection would be needed if we were to extend this file system to accommodate files up to 1 exbibyte (1024^6) in size?

   For questions f) to j) you have to specify the number of block accesses (reads and writes) which are required to complete the task.

   Assumptions:

   The inode for the file has been found and read into memory.

   All blocks outside the inode are not in memory at the start of the task but if loaded once during the task they stay in memory until that task is completed.

   File access times do NOT have to be written to the on disk inode.

   The free list of blocks can be found in memory.

   When the file is modified, the on disk inode must also be updated.

   The on disk inode occupies one block and several changes can be written with one write.

   All device reads and writes work on individual blocks.

   (f) Reading 100 bytes starting at byte position 4,050 of the file.

   (g) Reading 100 bytes starting at byte position 4,259,820.

   (h) Reading 100 bytes starting at byte position 4,263,900.

   (i) Writing 100 bytes starting at byte position 4,259,820 when the file length is 5,000,000. This overwrites data.

   (j) Writing 100 bytes starting at byte position 4,259,820 when the file is currently 4,259,820 bytes long.

   `[10 marks]`

2. There was a simple file system which used indexed allocation to store location information for its files. There was a pointer in each file directory entry called the key block. If a file was small enough to fit in one block the key block pointed to that one data block and the file was called a seedling file. If the file was too large to fit in one block then the key block pointed to an index block. This one index block pointed to as many data blocks as could fit in the space of the block. Files using one index block were called sapling files. If the file was too large to have just one index block, the key block pointed to a block consisting of pointers to index blocks. Files with these two levels of index block were known as tree

files. Each directory entry was always in one disk block. The free blocks were maintained as a bitmap stored on the disk.

(a) Given a block size of 512 bytes, and the space for a pointer to a block of 2 bytes, what are the minimum and maximum sizes of seedling, sapling and tree files? Give the answers in bytes.

[3 marks]

(b) Unlike the information above, in the real file system the first level index block for a tree file only held 128 index block pointers. Give a reason why this might be so. Extra mark: What was the name of this file system?

[1 (possibly 2) mark(s)]

(c) When a seedling file grows into a sapling file (by adding one extra byte to the file length) how many WRITES need to be made to the directory entry and any other blocks on the disk to store the changed information? Describe each of the writes e.g. change to bitmap block.

[1 mark]

(d) Answer the same question as part (c) for a sapling file growing into a tree file.

[1 mark]

3. Produce frame usage tables for the specified algorithms and say how many page faults occurred. The page reference string is along the top row. Each row after the top row represents one frame. So there are 4 frames of memory.

   A zero entry indicates the frame is empty. An entry of "=" means the frame holds the same page as in the previous column.

	1	2	3	4	5	4	3	2	1	6	7	1	2	3	7	5	1
0	1																
0	=																
0	=																
0	=																

   (a) FIFO - First In First Out

   (b) LRU - Least Recently Used

   (c) Optimal

[6 marks]

4. Summarise, in your own words, five significant changes made to the memory management system in the transition from Windows 7 to Windows 8 and 10. Also explain why those changes were made. Write no more than a paragraph about each change.

[10 marks]

Submitting the assignment

Make sure your login name (UPI) is included in every file you submit.

Use the assignment drop box to submit. adb.auckland.ac.nz

Any work you submit must be your work and your work alone – see the information on academic integrity http://www.auckland.ac.nz/uoa/home/about/teaching-learning/honesty.

You files will be run through TurnItIn to check for plagiarism.