THE UNIVERSITY OF AUCKLAND
NEW ZEALAND

COMPSCI 340 / SOFTENG 370
Operating Systems
Assignment 3 – Files and Memory
Worth 10%
Due date: 09:30pm 23rd of October 2020

For all questions show your working.

Question 1. (11 marks)

For these questions consider a Unix like file system as on page 18 of lecture 17 where the number of direct blocks are 16. Each block is 4KiB and a block number consists of 4 bytes.
If you write a program to help calculate these answers you may submit this program. Ensure that the program is clear with good comments (with 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.

(k) Explain why each of the listed assumptions need to be made.

Question 2. (3 marks)

Over the years architectures have changed from 8 to 16 to 32 to 64-bit machines. At the same time the address spaces have moved from 16 (there were never any widely used 8-bit address spaces) to 32 to 64-bit addresses. Give an example of a CPU which provides 64-bit addresses but is backwardly compatible with code that produces 32-bit code.

What consequences does moving from a 32-bit environment to a 64-bit environment have on programs, including the operating system?

Question 3. (4 marks)

We can keep track of free memory in a similar way to free blocks on disk. We could use a bitmap where 1 bit represents a free frame of memory or we could maintain a linked list of free frames. The linked list is maintained as a list of nodes where each node has a frame number, and a pointer to the next node. Assume these pointers and numbers are 32 bits each.
Calculate the space requirements using each of those methods for the following system, give the answers in bytes:
16 GB (2^34 bytes) of RAM in 8 KB (2^13 bytes) frames. Assume that memory is currently being used in alternating chunks of size 1 MB (2^20 bytes). i.e. The first MB of RAM is used, the second is free, the third is used, etc.
How much of that space (for both approaches) would normally be in kernel memory? Why?
How would an extents version compare to the two approaches in the question above?

Question 4. (2 marks)

We have a system with 32-bit virtual addresses and 1024 processes. Assuming all pages may need to be stored in virtual memory without any sharing, what is the maximum amount of space we should allocate for swap space? How realistic is this amount?

Question 5. (1 mark)

You are given the following data about a virtual memory system:
- the TLB can hold 1024 entries and can be accessed in 1 clock cycle, 1 nanosecond.
- memory can be accessed in 50 clock cycles or 50 nanoseconds.
- a page table entry can be found in 100 clock cycles or 100 nanoseconds.
- the average page replacement time is 5 milliseconds.

If page references are handled by the TLB 99% of the time, and only 0.01% lead to a page fault, what is the effective access time?

Question 6. (2 marks)

A computer has 64-bit virtual addresses and 4MB (2^22 byte) pages. For a particular process the program and data together fit in the lowest page. The stack fits in the highest page. Assume no other pages are necessary. How many levels would be required to be allocated for a multi-level page table to cover the full address space, with each page table entry in all levels occupying 2^6 bytes? For our process what is the smallest number of pages that need to be allocated for its page table information, count the page table pages and all necessary indirect page tables. Show your working and give an explanation.

Question 7. (2 marks)

The number of instructions executed between page faults is directly proportional to the number of page frames allocated to a program. Double the amount of memory allocated to a process and double the length of time between page faults. Suppose that a normal instruction takes 1 nanosecond, but if a page fault occurs, it takes 2,000,000 nanoseconds (i.e. 2msec) to handle the fault. If a program takes 60 sec to run, during which time it gets 20,000 page faults, how long would it take to run if twice as much memory was available? Show your working.

Question 8. (8 marks)

Here is a page reference string:

1,2,3,4,5,4,3,2,1,6,7,1,2,3,7,5,1

Given a machine with 4 frames - show the contents of each frame using the following replacement algorithms on that reference string. **Also say how many page faults would occur for each algorithm.** Pages are brought in on demand and the initial load of a page counts as a page fault.

(a) FIFO - First In First Out

(b) LRU - Least Recently Used

(c) LFU - Least Frequently Used (if there are multiple pages with the same lowest frequency choose in a FIFO manner)

(d) Optimal (if there are multiple pages which are not used again choose in a FIFO manner)

Lay out your answer for each algorithm like this. A zero means the frame is free, an "=" means the content of the frame is the same as in the previous step.

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

Submitting the assignment

Extra marks (1 Mark) - Make sure your name and upi is included in the file you submit.

Submit your answers to the questions as a single pdf or doc file, called a3Answers.pdf or a3Answers.doc.

These files must be readable to **Turnitin** to enable a check for uniqueness. Otherwise you will get zero for the assignment. You may submit multiple times and your most recent submission will be marked.

Any work you submit must be your work and your work alone – see the Departmental and University policies on academic integrity.