

Dan Otieno.

CPE434-01.

HOMEWORK 5.

04/27/23.

1. In a previous homework you had a virtual address space with a page size of 64KB. For small programs this large page size results in lots of wasted space. Assume you were going to redesign this architecture for a page of 1k bytes (1024) how would you allocate the extra 6 bits to the index fields. For example you could do (0,0,+6), (0,+6,0), (+6,0,0), (+2,+2,+2), or other changes to the size of the levels. For small programs consisting of one page of text and heap, and one page of stack, which allocation of bits would be more efficient? Explain your answer.

For reference, information from the previous problem is as follows: Multi-level page tables. Assume a virtual address machine with a 32bit address. Assume the address is divided into 4 parts a,b,c,d with the first three parts used to index into a three level page table and the fourth is used to index into the location on the page. Assume the values for a,b,c,d are 4,6,6,16 respectively resulting in a 64 KB sized page. What is the total size in bytes of the page tables for the smallest program, which is a program containing a single page of text and heap, starting at location 0, and a separate page for the stack, starting at location 0xFFFFFFFF and running downwards. Assume all page table entries are 4 bytes each.

- **The most efficient allocation of bits for a small program consisting of one page of text and heap, and one page of stack would be (+2, +2, +2). Because the goal is to decrease the amount of wasted space, we can reduce the size of the page to 1k bytes, which will leave us with an additional 6 bits in the index fields. The total size of space used, would then be 48 bytes ($2^2 * 3 * 4$).**

2. Given the following page reference string:

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

What number of faults would occur for the following replacement algorithms where you have one, two, or three frames. Assume all frames are originally empty. Therefore your first pages will cost one fault each.

- LRU replacement:
- FIFO replacement

- Optimal replacement

FRAMES	LRU REPLACEMENT	FIFO REPLACEMENT	OPTIMAL REPLACEMENT
1	18	18	18
2	16	16	13
3	13	14	9
4	8	12	7
5	8	8	7
6	7	8	7
7	7	7	7

3. What are the advantages and disadvantages of loadable kernel modules? (2 points)

- **Loadable kernel modules are made up of code to extend the base kernel. They add file system support, as well as support for hardware, system calls, among other system functionalities. Loadable kernel modules help avoid memory wastage because the systems which are not in used often can be contained outside the base kernel. Loadable modules also ensure that users do not have to rebuild and reboot the base kernel each time to access a functionality. One disadvantage is that the module is fragmented each time a kernel module is inserted, which affects overall system performance.**

SOURCE.