Johnny Li

COP4600

HW 3.2

41) A computer provides each process with 65,536 bytes of address space divided into pages of 4096 bytes each. A particular program has a text size of 32,768 bytes, a data size of 16,386 bytes, and a stack size of 15,870 bytes. Will this program fit in the machine’s address space? Suppose that instead of 4096 bytes, the page size were 512 bytes, would it then fit? Each page must contain either text, data, or stack, not a mixture of two or three of them

65,536 bytes of address space Page size of 4096 bytes

Text size of 32,768 bytes Data size of 16,386 Stack size 15870 byes

Number of text pages = Text size/Page size = 32,768 bytes/4096 bytes = 8 text pages

Number of data pages = Data size/Page size = 16,386 bytes/4096 bytes = 5 data pages

Number of stack pages = Stack size/Page size = 15,870 bytes/4096 bytes = 4 text pages

Total pages needed = 8+5+4 = 17 pages

Total pages given = Address space size/Page size = 65,536 bytes/4096 bytes = 16 pages

This program will not fit in the machine’s address space as the program needs 17 pages but the maximum pages given by the address space for a program is only 16, not enough to fit all 17 pages.

However, if page size were 512 bytes then:

Number of text pages = Text size/Page size = 32,768 bytes/512 bytes = 64 text pages

Number of data pages = Data size/Page size = 16,386 bytes/512 bytes = 33 data pages

Number of stack pages = Stack size/Page size = 15,870 bytes/512 bytes = 31 text pages

Total pages needed = 64+33+31 = 128 pages

Total pages given = Address space size/Page size = 65,536 bytes/512 bytes = 128 pages

This program will fit in the machine’s address space as the program needs 128 pages and the maximum pages given by the address space for a program is 128, enough to fit all the pages.

42) It has been observed that the number of instructions executed between page faults is directly proportional to the number of page frames allocated to a program. If the available memory is doubled, the mean interval between page faults is also doubled. Suppose that a normal instruction takes 1 microsec, but if a page fault occurs, it takes 2001 μ sec (i.e., 2 msec) to handle the fault. If a program takes 60 sec to run, during which time it gets 15,000 page faults, how long would it take to run if twice as much memory were available?

Instruction = 1 msec Page Fault = 2001 usec

Program = 60 sec 15000 page faults

Double memory result in 1 msec \* 2 = 2 msec processing time

In 1 minute, 30 sec is on page fault overhead and the other 30 sec is on program run. -🡪 Double memory would half the page faults so it would result in 15 sec of page fault overhead and 15 sec with 45 sec of program run.

45) Explain the difference between internal fragmentation and external fragmentation. Which one occurs in paging systems? Which one occurs in systems using pure segmentation?

Internal fragmentation is when the memory is split into sized blocks and, where a method request for the memory, the sized block is allotted to the method but is somewhat larger than the memory requested, the distinction between allotted and requested memory. In external fragmentation, variable-sized memory blocks square measure appointed to method, the unused spaces formed between non-contiguous memory fragments are too small to serve a new process. In a paging system, internal fragmentation occurs on the last page with wasted space. In a segmentation system, external fragmentation occurs as some space is invariably lost between the segments.

46) When segmentation and paging are both being used, as in MULTICS, first the segment descriptor must be looked up, then the page descriptor. Does the TLB also work this way, with two levels of lookup?

The TLB does not work this way of first the segment descriptor must be looked up, then the page descriptor, with two levels of lookup as a search key is uses on both the segment number and the virtual page number, thus an in an single run the exact page can be found.

47) We consider a program which has the two segments shown below consisting of instructions in segment 0, and read/write data in segment 1. Segment 0 has read/execute protection, and segment 1 has just read/write protection. The memory system is a demand paged virtual memory system with virtual addresses that have a 4-bit page number, and a 10-bit offset. The page tables and protection are as follows (all numbers in the table are in decimal). For each of the following cases, either give the real (actual) memory address which results from dynamic address translation or identify the type of fault which occurs (either page or protection fault).

(a) Fetch from segment 1, page 1, offset 3

No fault occurs as the page in within the table, directly obtain value of page frame 14 offset by 3 (14,3).

(b) Store into segment 0, page 0, offset 16

A protection fault occurs as there is a write to read/execute segment in process. No address is obtained.

(c) Fetch from segment 1, page 4, offset 28

A page fault occurs as the page is not in RAM but on the disk. No address is obtained.

(d) Jump to location in segment 1, page 3, offset 32.

A protection fault occurs as there was an attempt to jump to read/write segment. No address is obtained.