CISC 3595 Operating Systems

Fall 2022
Out 12/06 In 12/12
Assignment #6

Q1. Consider a logical address space of 256 pages of 2,048 words each, mapped onto a physical memory of 128 frames.

(a) How many bits are there in the logical address?
(b) How many bits are there in the physical address?

*Logical page number 8 bits, page offset 11 bits: 19 bits*
*Physical page number 7 bits, offset 11 bits: 18 bits.*

*Remember that if the number of pages is $2^n$ and the size of a page is $2^m$, the address space will be $2^{m+n}$*

Q2. Compare the memory organization schemes of contiguous memory allocation, pure paging and pure segmentation with respect to the following issues:

- External fragmentation
- Internal fragmentation
- Ability/ease of sharing code across processes.

| | *Contiguous* | *Pure Segmentation* | *Pure Paging* |
|---|---|---|---|
| *External* | *Will suffer from this unless fixed partition* | *Will suffer from this unless fixed partition* | *Avoids* |
| *Internal* | *Avoids if fixed partition* | *Avoids this if fixed partition* | *Averages ½ page per process* |
| *Share* | *Difficult* | *Users can share segments* | *Users can share pages* |

Q3. Assuming a 1-KB page size, what are the page numbers and offsets for the following address references (provided as decimal numbers):

- 4378; (b) 19566; (c) 30000; (d) 258; (e) 16388

| ***pg#,offset*** | ***pg#*** | ***offset*** | ***pg#*** | ***offset*** |
|---|---|---|---|---|
| • *4378; 4,282* | *000100* | *0100011010* | *4378/1024,* | *4378%1024* |
| • *19566; 19,110* | *010010* | *1110100110* | *19566/1024,* | *19566%1024* |
| • *30000; 29, 304* | *011101* | *0100110000* | *30000/1024,* | *30000%1024* |
| • *258; 0,258* | *000000* | *0100000010* | *258/1024,* | *258%1024* |
| • *16388; 16,4* | *010000* | *0000000100* | *16388/1024,* | *16388%1024* |

Q4. Consider a system that uses pure demand paging. (10 pts)

- When a process first starts executing, how would you characterize the page fault rate?
- Once the working set for the process is loaded into memory, how would you characterize the page fault rate?
- Assume that a process changes its locality and the size of the new working set is too large in available free memory. Identify some options system designers could choose from to handle this situation.

*a. High page fault rate as pages are brought in for the first time. (3)*
*b. Low page fault rate since working set has all the pages that are typically used. (3)*
*c. option 1: Process can be suspended until enough memory becomes available, or another lower priority process can be selected and it suspended instead until space is available.*
*c. option 2 Can modify the time duration over which the working set to reduce working set size, this will produce a higher page fault rate*
*c. option 3 Can switch to using a page fault frequency model rather than a working set model. (4 – at least two options, 2 ea).*

Q5. Answer all the following questions for paged memory:

- Given a logical address space of 32 bits, and a page size of 26 bits, how many pages are possible?
  *32-26 bits = 6 bits for pages, so 64 pages of 26 bits (or 1GB). [4]*

- For the example in (a) above, list the addresses of the first three frames in physical memory. [4]
  *000000 00 0000 0000 0000 0000 0000 0000 , 0x2^26*
  *000001 00 0000 0000 0000 0000 0000 0000, 1x2^26*
  *000010 00 0000 0000 0000 0000 0000 0000, 2x2^26 = 2^27*
  *000011 00 0000 0000 0000 0000 0000 0000 3x2^26*
  *Remember that the frame# is prepended to the offset .*
- For the example in (a) above, what is the smallest space that could be allocated for a page table? [4]
  *chapter 9 full process in memory, 64 6-bit numbers, 384 bits needed or for 8-bits 512 or 64 bytes.*
  *chapter 10 starts with 64 6-bits plus each entry has a valid bit, a read-write privilege bit, a modified bit, which is 9 bits, so 64 9-bit entries is 576 bits or for 12-bits, 768 bits which is 96 bytes.*[6]
- Given a 256 entry page table, and a logical address space of 48 bits, how big must each physical memory frame be? [3]
  *256 entries means 8 bits per page number. So each page is 48-8 = 40 bits for each page.*
  *A frame must hold a page, so each frame must be 2^40 bits, 2^37 bytes.*

Q6. Consider a demand-paging implementation with 3 frames, all initially empty. Demonstrate the sequence of page replacements that would occur for the reference sequence below for (a) First-In-First-Out (FIFO) and (b) Least Recently Used (LRU). For both cases, calculate the total number of page faults.

Reference String: 5 0 5 4 2 0 5 0 2 3 2 5 3 1 5 3

*FIFO 12=8+4, [8; 4 for method 4 for correct] 4x0, 5x2, 4x1, 3x7*
*5 5 5 5 2 2 2 2 2 2 3 3 3 3 1 1 1*
*- 0 0 0 0 0 5 5 5 5 2 2 2 2 2 2 3*
*- - - 4 4 4 4 0 0 0 0 5 5 5 5 5*
*f f - f - f f - - f f f - f - f*
*5 non-pf in 16 refs => 11 page faults [4]*

*LRU 13*
*5 5 5 5 5 0 0 0 0 0 0 5 5 5 5 5*
*- 0 0 0 2 2 2 2 2 2 2 2 2 1 1 1*
*- - - 4 4 4 5 5 5 3 3 3 3 3 3 3*
*f f - f f f f - - f - f - f - -*

*7 non-pf in 16 refs => 9 page faults [4]*

Q7. Assume that we have a demand-paged memory. The page table is held in registers. It takes 5 milliseconds to service a page fault if an empty frame is available or if the replaced page is not modified and 20 milliseconds if the replaced page is modified. Memory-access time is 120 nanoseconds.

Assume that the page to be replaced is modified 85 percent of the time. What is the maximum acceptable page-fault rate for an effective access time of no more than 200 nanoseconds?

*First calculate the replacement cost which is the average time to service a page fault.*

*TSP = (p)*25 + (1-p)*5 = (0.85*25+0.15*5) ms = 22 = 22 ms (time to service page fault)*

*Then use the Effective Access Time formula: (1-p)*ma + p * TSP = EAT*

*EAT = (1-p)* 0.00012 ms + p * 22 (effective access time) = .0002*

*EAT = 0.00012ms -p*0.00012ms+p*22ms = 0.0002 = p*(22-.00012)ms=0.0002-0.00012*

*p*(22-.00012) = .00008%*

*p = (0.00008) / (22-0.00012) = 0.00008/21.99988 = 0.00000363638 .00036% less*