MEMORY MANAGEMENT

Chapter 7: Main Memory

CS220: OS -Spring 2021

Numericals

Course Supervisor: Anaum Hamid

Context Switch Time including Swapping

Problem:

100MB process swapping to hard disk with transfer rate of 50MB/sec from main memory.

- Swap Time (sec) = Size (mb)/ transfer rate (mb/sec)
  - Swap Time = 100/50 = 2sec = 2000ms
  - Swap[out] time of 2000 ms
  - Plus swap in of same sized process
  - Total context switch swapping component time = 4000ms (4 seconds)

Dynamic Storage-Allocation Problem

Given five memory partitions of 100Kb, 500Kb, 200Kb, 300Kb, 600Kb (in order), how would the first-fit, best-fit, and worst-fit algorithms place processes of 212 Kb, 417 Kb, 112 Kb, and 426 Kb (in order)? Which algorithm makes the most efficient use of memory?

First-fit:
212K is put in 500K partition
417K is put in 600K partition
112K is put in 288K partition (new partition 288K = 500K - 212K)
426K must wait

Best-fit:
212K is put in 300K partition
417K is put in 500K partition
112K is put in 200K partition
426K is put in 600K partition

Dynamic Storage-Allocation Problem

Worst-fit:
212K is put in 600K partition
417K is put in 500K partition
112K is put in 388K partition
426K must wait

In this example, best-fit turns out to be the best

Paging

- Calculating internal fragmentation

  Given:
  - Page size = 2,048 bytes
  - Process size = 72,766 bytes
  - 35 pages + 1,086 bytes

  Calculate:
  - Internal fragmentation of 2,048 - 1,086 = 962 bytes
  - Worst case fragmentation = 1 frame – 1 byte
  - On average fragmentation = 1 / 2 frame size
- Process view and physical memory now very different
- By implementation process can only access its own memory

Things to recall

- 2^10= 1 Thousand= 1KB
- 2^20 = 1 million = 1 MB
- 2^30 = 1 Billion = 1 GB
- 2^40 = 1 Trillion = 1 TB

2^{1}	=	2	2^{11}	=	2,048	2^{21}	=	2,097,152
2^{2}	=	4	**2^{12}**	**=**	**4,096**	2^{22}	=	4,194,304
2^{3}	=	8	2^{13}	=	8,192	2^{23}	=	8,388,608
2^{4}	**=**	16	2^{14}	=	16,384	**2^{24}**	**=**	**16,777,216**
2^{5}	=	32	2^{15}	=	32,768	2^{25}	=	33,554,432
2^{6}	=	64	**2^{16}**	**=**	**65,536**	2^{26}	=	67,108,864
2^{7}	=	128	2^{17}	=	131,072	2^{27}	=	134,217,728
2^{8}	**=**	256	2^{18}	=	262,144	**2^{28}**	**=**	**268,435,456**
2^{9}	=	512	2^{19}	=	524,288	2^{29}	=	536,870,912
2^{10}	=	1,024	**2^{20}**	**=**	**1,048,576**	2^{30}	=	1,073,741,824

Paging Numerical-1

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

a. 3085

b. 42095

c. 215201

d. 650000

e. 2000001

Page Numerical-1 (Solution2)

3085/1024= 3.0126953125
3 = pages
0.126953125 * 1024 = 13

Page size = 1 KB = 1024B

Page number	Offset
3085/1024 = 3	3085 mod 1024 = 13
42095/1024 = 41	42095 mod 1024 = 111
215201/1024 = 210	215201 mod 1024 =161
650000/1024 =634	650000 mod 1024 =784
2000001/1024 = 1953	2000001 mod 1024 = 129

Paging Numerical-2

- Consider a logical address space (LAS) of 32 pages with 1024 words per page; mapped onto a physical memory of 16 frames.

1. How many bits are required in the logical address?
2. How many bits are required in the physical address?

Answer:

1. $2^5 * 2^{10} = 2^{15} =$ 15 bits.
2. $2^4 * 2^{10} = 2^{14} =$ 14 bits.

Effective Access Time [Cont.]

- **Effective Access Time** (**EAT**)

EAT= α x memory access time (hit) + fail x memory access time (failed)

Effective Access Time [Cont.]

- Consider α = 80%, 100ns for memory access
  - EAT = 0.80 x 100 + 0.20 x (100+100) = 120ns

- Consider more realistic hit ratio -> α = 99% , 100ns for memory access
  - EAT = 0.99 x 100 + 0.01 x (100+100) = 101ns

EAT Numerical

Consider a paging system with the page table stored in memory.

1. If a memory reference takes 200 nanoseconds, how long does a paged memory reference take?
2. If we add associative registers, and 75 percent of all page-table references are found in the associative registers, what is the effective memory reference time? (Assume that finding a page-table entry in the associative registers takes zero time if the entry is there.)

EAT Numerical (Sol)

Answer:

1. 400 nanoseconds: 200 nanoseconds to access the page table and 200 nanoseconds to access the word in memory.
2. Effective access time = 0.75 × (200 nanoseconds) + 0.25 × (400 nanoseconds) = 250 nanoseconds.

Thank you!