COMP 3411

Assignment 6

Martin Atanacio - T00684924

Question 1

Part of Memory, Available for Allocation

10KB	5KB	20KB	18KB	6KB	9KB	12KB	15KB

Process A = 15KB, **Process B** = 10KB, **Process C** = 11KB

First Fit - allocate the first hole that is big enough

B	5KB	A	C	6KB	9KB	12KB	15KB
		5KB remaining	*7KB remaining*				

Best Fit - allocate the smallest hole that is big enough (searches entire list)

B	5KB	20KB	18KB	6KB	9KB	C	A
						1KB remaining	

Worst Fit - allocate the largest hole (searches entire list)

10KB	5KB	A	B	6KB	9KB	12KB	C
		5KB remaining	*8KB remaining*				*4KB remaining*

Next Fit - same as first fit, except when called again, it starts searching from where it left off

10KB	5KB	A	B	6KB	9KB	C	15KB
		5KB remaining	*8KB remaining*			*1KB remaining*	

Which algorithm makes the best use of memory space?

Best fit algorithm makes the best use of memory space, as it tries to find the smallest available segment that can accommodate the process. Thus, minimizing memory wastage due to fragmentation. However, best fit is slower than other algorithms, as it needs to search the entire memory for the smallest free partition. Additionally, it may tend to fill up memory and leave behind some tiny gaps that could potentially be used together to accommodate more processes.

Question 2

A computer has a 32-bit virtual address space with a page size of 8KB and 4 bytes per page table entry.

1. **How many pages are in the virtual address space?**
   32-bit virtual address space means that there are 32 bits available for addressing.

   Page size = 8KB (2^{13} bytes)

   No. of pages = Total address space size / Page size
   = 2^{32} / 2^{13} = 2^{19}
   = 2^{19} (524,288) pages in the virtual address space

2. **What is the maximum size of addressable physical memory in this system?**

   Max size = No. of pages * page size
   = 2^{19} * 2^{13}
   = 2^{32} bytes = 4GB

Question 3

Suppose that we have free segments with sizes: 5, 15, 25, 10, and 20. Place a program with size 16KB in the free segment using first-fit, best-fit and worst fit? Indicate which segment it will be in for each of the 3 algorithms.

1. **First-fit (allocate the first hole that is big enough)**
   Program will be placed in the 25KB free segment

2. **Best-fit (allocate the smallest hole that is big enough)**
   Program will be placed in the 20KB free segment

3. **Worst-fit (allocate the largest hole)**
   Program will be placed in the 25KB free segment

Question 4

Consider a logical address space of 4 pages of 512 words each, mapped onto a physical memory of 16 frames.

1. **How many bits are there in the logical address?**

   Total No. of bits for logical address = No. of bits per page + No. of bits per word

   No. of bits per page = $\log_2 (4)$ = 2 bits *(round up to accommodate all entries)*
   No. of bits per word = $\log_2 (512)$ = 9 bits *(round up to accommodate all entries)*

   = 2 bits + 9 bits = 11 bits

2. **How many bits are there in the physical address?**

   Total No. of bits for physical address = No. of bits per frame number + No. of bits per word

   No. of bits per frame = $\log_2 (16)$ = 4 bits *(round up to accommodate all entries)*
   No. of bits per word = $\log_2 (512)$ = 9 bits *(round up to accommodate all entries)*

   = 4 bits + 9 bits = 13 bits

Question 5

Virtual page number	Valid bit	Page frame number
0	1	4
1	1	7
2	0	1
3	1	2
4	0	8
5	1	0

Page size = 512 bytes

Steps

1. Determine the virtual page number using the **virtual address / page size**
2. Check the **validity bit** for the corresponding **virtual page number** in the page table
3. If **validity bit = 1**, use the page frame number from the page table to construct the **physical address**
    a. **Offset within page =** virtual address % page size
    b. **Physical address =** (frame # * page size) + offset

Calculate the physical addresses for each of the following virtual addresses & check validity.

A. **Virtual Address: 152**
   Virtual Page Number: 152 / 512 = 0
   Valid Bit: 1 (valid)
   Page Frame #: 4
   Offset: 152 % 512 = 152
   Physical Address: (4 * 512) + 152 = 2184
   Valid Physical Address: Yes

B. **Virtual Address: 1121**
   Virtual Page Number: 1121 / 512 = 2
   Valid Bit: 0 (invalid)
   Page Frame #: Not applicable
   Offset: Not applicable
   Physical Address: Not applicable
   Valid Physical Address: No

C. **Virtual Address: 2499**
   Virtual Page Number: 2499 / 512 = 4
   Valid Bit: 0 (invalid)
   Page Frame #: Not applicable
   Offset: Not applicable
   Physical Address: Not applicable
   Valid Physical Address: No

Question 6

Segment	Base	Limit
0	502	700
1	1379	200
2	99	125
3	1410	1550
4	2025	2750

Steps
1. Question is in the form **<Segment-Number, Offset>**
2. Grab the base according to the segment number
3. **Physical Address =** Base + Offset

What are the physical addresses for the following logical addresses, segment number, and offset?

A. 0, 131

Segment: 0
Base: 502
Offset: 131
Physical Address: 502 + 131 = 633 (finishes on 1333) // + limit to mark the end of segment

B. 1, 399

Segment: 1
Base: 1379
Offset: 399
Physical Address: 1379 + 399 = 1778 (finishes on 1978)

C. 2, 174

Segment: 2
Base: 99
Offset: 174
Physical Address: 99 + 174 = 273 (finishes on 398)

D. 3, 203

Segment: 3
Base: 1410
Offset: 203
Physical Address: 1410 + 203 = 1613 (finishes on 3163)

E. 4, 606

Segment: 4
Base: 2025
Offset: 606
Physical Address: 2025 + 606 = 2631 (finishes on 5381)

Question 7

Consider the following page-reference string in virtual memory management:

6, 2, 1, 3, 7, 6, 1, 2, 8, 3, 4, 2, 1, 5, 3

Assuming 4 frames, how many page faults would occur for the following replacement algorithms? Remember that all frames are initially empty, so the first unique page will cost one fault.

Number of Page faults are found by counting the # of red cells (pages that have been swapped out)

A. LRU replacement - look back and swap out the left-most page (least recently used)

6	2	1	3	7	6	1	2	8	3	4	2	1	5	3
6	6	6	6	7	7		7	8	8	8		1	1	1
	2	2	2	2	6		6	6	3	3		3	5	5
		1	1	1	1	HIT	1	1	1	4		4	4	3
			3	3	3		2	2	2	2	HIT	2	2	2

= 13 Page Faults

B. FIFO replacement - swap out the oldest page

6	2	1	3	7	6	1	2	8	3	4	2	1	5	3
6	6	6	6	7	7		7	7	3	3		3	3	HIT
	2	2	2	2	6		6	6	6	4		4	4	
		1	1	1	1	HIT	2	2	2	2	HIT	1	1	
			3	3	3		3	8	8	8		8	5	

= 12 Page faults

C. Optimal replacement - look ahead and swap out the page that is not going to be used yet

6	2	1	3	7	6	1	2	8	3	4	2	1	5	3
6	6	6	6	6	HIT			8	8	4			5	
	2	2	2	2			HIT	2	2	2	HIT		2	
		1	1	1		HIT		1	1	1		HIT	1	
			3	7				7	3	3			3	HIT

= 9 Page Faults