Operating Systems: Homework #5

Due on March 30, 2016 at 11:59pm

Professor Qu
Monday & Wednesday 3:30pm — 5:17pm

Nicholas Land

Problem 1

Why is the protection of processes' memory space important? Describe a scenario where absence of memory protection leads to problems.

Solution

Protection of processes' memory space is important because it prevents processes from accessing memory that have not yet been allocated. A scenario in which could cause problems with the absence of memeory protection would be when a process attempts to access memory that hasn't been allocated yet will cause the program to crash.

Problem 2

Consider a system where the virtual memory page size is 1KB (1024 bytes), and main memory consists of 4 page frames, which are empty initially. Now consider a process, which requires 8 pages of storage. At some point during its execution, the page table is as shown below:

Virtual Page #	Physical Page #	Valid Flag
0		No
1		No
2	2	Yes
3	3	Yes
4		No
5		No
6	0	Yes
7	1	Yes

1. List the virtual address ranges that will result in a page fault.
2. Give the following ordered references to the virtual addresses (i) 4500, (ii) 8000, (iii) 3000, (iv) 1100, please calculate the main memory (physical) addresses. If there is a page fault, please use LRU based page replacement to replace the page. How which page will be affected and compute the physical addresses after the page fault. We assume the reference string is . . . 2 4 7 3 0 4 3 0 7 5 0 7 6 0 2 3 6 4 7 6 3 2 before the new reference.

Solution

1. The virtual address ranges that will result in page fault are:
   Page 0: 0 – 1023
   Page 1: 1024 – 2047
   Page 4: 4096 – 5119
   Page 5: 5020 – 6143

2. References as follows:

(i) 4500 is a page fault. Reference string is:

2	4	7	3	0	4	3	0	7	5	0	7	6	0	2	3	6	4	7	6	3	2	4
2	2	2	2	0					0			0		0	0		4	4			2	2
	4	4	4	4					5			5		2	2		2	7			7	4
		7	7	7					7			7		7	3		3	3			3	3
			3	3					3			6		6	6		6	6			6	6

Virtual Page #	Physical Page #	Valid Flag
0		No
1		No
2	2	Yes
3	3	Yes
4	1	Yes
5		No
6	0	Yes
7		No

*Page # * Page Size + Offset = Virtual Address*
Offset is found by $4500 - (4 * 1024) = 404$
Therfore $\ldots 4500 = 4 * 1024 + 404$
4500 in binary is [100][0110010100] & our offset (404) in binary is 0110010100. The 100 gets replaced by 01 for our page number which translates to a physical address of [01][0110010100] which is 1428.

(ii) After the page replacement, 8000 is a page fault. Reference string is:

2	4	7	3	0	4	3	0	7	5	0	7	6	0	2	3	6	4	7	6	3	2	4	7
2	2	2	2	0					0			0		0	0		4	4			2	2	2
	4	4	4	4					5			5		2	2		2	7			7	4	4
		7	7	7					7			7		7	3		3	3			3	3	3
			3	3					3			6		6	6		6	6			6	6	7

Virtual Page #	Physical Page #	Valid Flag
0		No
1		No
2	2	Yes
3	3	Yes
4	1	Yes
5		No
6		No
7	0	Yes

Offset is found by $8000 - (7 * 1024) = 832$
Therfore $\ldots 8000 = 7 * 1024 + 832$
8000 in binary is [000111][1101000000] & our offset (832) in binary is 1101000000. The 000111 gets replaced by 00 for our page number which translates to a physical address of [00][1101000000] which is 832.

Problem 2 continued on next page...

(iii) There is no page fault on 3000 so reference string & page table remain the same.

2	4	7	3	0	4	3	0	7	5	0	7	6	0	2	3	6	4	7	6	3	2	4	7
2	2	2	2	0					0			0		0	0		4	4			2	2	2
	4	4	4	4					5			5		2	2		2	7			7	4	4
		7	7	7					7			7		7	3		3	3			3	3	3
			3	3					3			6		6	6		6	6			6	6	7

Virtual Page #	Physical Page #	Valid Flag
0		No
1		No
2	2	Yes
3	3	Yes
4	1	Yes
5		No
6		No
7	0	Yes

Offset is found by $3000 - (2 * 1024) = 952$
Therfore $\ldots 3000 = 2 * 1024 + 952$
3000 in binary is 0000101110111000. The 10 remains the same for our page number (2) which translates to a physical address of 101110111000 which is 3000

(iv) 1100 results in a page fault, reference string is:

2	4	7	3	0	4	3	0	7	5	0	7	6	0	2	3	6	4	7	6	3	2	4	7	1
2	2	2	2	0					0			0		0	0		4	4			2	2	2	2
	4	4	4	4					5			5		2	2		2	7			7	4	4	4
		7	7	7					7			7		7	3		3	3			3	3	3	1
			3	3					3			6		6	6		6	6			6	6	7	7

Virtual Page #	Physical Page #	Valid Flag
0		No
1	3	Yes
2	2	Yes
3		No
4	1	Yes
5		No
6		No
7	0	Yes

Offset is found by $1100 - (1 * 1024) = 76$
Therfore $\ldots 1100 = 1 * 1024 + 76$
3000 in binary is 0000010001001100. The 100 translates to 011 for our page number (3) which translates to a physical address of 01101001100 which is 844