Fundamentos de los Sistemas Operativos (FSO)

Departamento de Informática de Sistemas y Computadoras (DISCA)
Universitat Politècnica de València

Part 4: Memory management

Seminar 11
Virtual memory exercises

fSO

DISCA

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Contenido

- Exercise 1. Paging whitout virtual memory
- Exercise 2. Paging with virtual memory
- Exercise 3. Replacement algorithms
- Exercise 4. Replacement scope

A processor has 16-bit logical addresses managed by paging. It is made in three versions with page sizes of 256, 1024 y 4096 bytes, respectively.
A given executable file contains 2800 bytes of instructions from address 0x1000, 1198 bytes of data from address 0x3000 and reserves initially 2048 bytes for the stack from address 0x9000.

a) Compute the number of page table entries and the initial number of pages in use for every processor model

Page size (bytes)	Number of page table entries	Initial number of pages
256		
1024		
4096		

Exercise 1 (cont.)

Region	Size (bytes)	Base (hex)
Code	2800	1000
Variables	1198	3000
Stack	2048	9000

b) Build *in binary* the initial process page table when the program is executed with 4096 bytes as page size.
Cosider that physical memory size is 64 KBytes and with the process is loaded only the upper frames are allocated. The OS allocates frames in ascending order of physical addresses. It allocates first the code, then variables and finally the stack.
Page descriptors have the following content:

frame	v	r	w	x

If a page is not in use it has to be indicated by r=w=x=0

0		
1		
2		
3		
4		
5		
6		
7		
8		
9		
A		
B		
C		
D		
E		
F		

Exercise 1 (cont.)

c) Compute, if possible, the physical addresses that the MMU generates for every of the following accesses. If the translation is not possible, tell why.

Region	Size (bytes)	Base (hex)
Code	2800	1000
Variables	1198	3000
Stack	2048	9000

Access type	Logical address	Physical address	Legal access? (yes/no)
Instruction read	1000		
Instruction read	1004		
Instruction read	2000		
Instruction read	3000		
Variable read	3010		
Variable read	9010		
Variable write	1000		
Variable write	5000		

Exercise 2. Paging with virtual memory

fSO

Consider the processor from exercise 1 with 4-KByte pages, and an OS with **virtual memry**. The process starts without any frame allocated and the OS alloctes free frames in the following order: frame 0, frame 1, frame 2, etc.

Region	Size (bytes)	Base (hex)
Code	2800	1000
Variables	1198	3000
Stack	2048	9000

a) Complete the next table considering that logical addresses are emitted following the "Logical address" column order.

Access type	Logical address (hex)	Physical address (hex)	Page fault?	Legal access?
Instruction read	1000			
Instruction read	1004			
Variable read	97FC			
Instruction read	1008			
Variable write	97F8			
Instruction read	5000			

¿How many frames has the process allocated after the fifth access?

¿Can the process continue after the last access?

Exercise 2 (cont.)

b) Describe *in hexadecimal* the evolution of the process page table.
Cosider that physical memory size is 64 KBytes and with the process is loaded only the upper frames are allocated. The OS allocates frames in ascending order of physical addresses.
Page descriptors are 16-bit and have the following content:

0	0	0	0	*frame*	v	r	w	x	0	0	0	m

If a page is not in use it has to be indicated by r=w=x=0

Initial state

0	0 ? 0 0
1	0 ? 5 0
2	0 ? 0 0
3	0 ? 6 0
4	0 ? 0 0
5	0 ? 0 0
6	0 ? 0 0
7	0 ? 0 0
8	0 ? 0 0
9	0 ? 6 0
A	0 ? 0 0
B	0 ? 0 0
C	0 ? 0 0
D	0 ? 0 0
E	0 ? 0 0
F	0 ? 0 0

Exercise 2 (cont.)

	Read 0x1000	Read 0x1004	Read 0x97FC	Read 0x1008	Write 0x97F8
0					
1					
2					
3					
4					
5					
6					
7					
8					
9					
A					
B					
C					
D					
E					
F					

Exercise 3. Replacement algorithms

In a computer with **32 MB of main memory, with a memory** management policy of **paging with 4KB page size, the OS** assigns to process A **4 main memory frames (from 0 to 3)**, that are initially empty.

Answer the following items:

a) Describe the **physical memory address format.**

b) If process A generates the following logical address sequence (shown in hexadecimal):

02D4B8, 02D4B9, 02D4EB, 02D4EB, 02D86F, F0B621, F0B815, F05963, F0B832, F0BA23, D946C3, D9B1A7, D9B1A1, F0BA25, 02D4C7, 628A31, F0B328, D9B325, D73425

Obtain, for **FIFO and LRU algorithms, how many page faults are** generated and the final main memory state, telling the page allocated in every frame assigned to the process.

Exercise 4. Replacement scope (intro)

- There are two possible page replacement scopes:
  - **Local replacement:**
    - A process selects the victim between its own pages allocated into main memory frames, it can not take frames from another process.
    - The number of process allocated frames doesn't change
  - **Global replacement:**
    - A process selects the victim between whole set of main memory frames
    - The victim can belong to another process different from the one that produces the page fault

Exercise 4. Replacement scope

On a virtual memory system, with 1024 byte page size, the OS has allocated 6 frames (from 0 to 5) to two processes A y B. At time t = 10, A and B page tables have the following content:

Process A page table

	Frame	Valid bit	Counter
0		i	
1		i	
2	2	v	10
3	5	v	3
4		i	
5	4	v	5
6		i	
7		i	

Process B page table

	Frame	Valid bit	Counter
0		i	
1		i	
2		i	
3	1	v	2
4		i	
5		i	
6		i	
7		i	

Then the processes emit the following logical address sequence. Consider that all the addresses are legal:

A,100; A,4000; B,100; A,7000; B,2100; B,1028; A,5800; A,100

Obtain what pages are allocated on every frame and the physical address translation of the first and the last access in the following situations:

a) The replacement algorithm is **LRU** with **global scope**
b) The replacement algorithm is **LRU** with **local scope**. Process A has 4 frames and process B has 2 frames