COMPUTER ARCHITECTURE
2020 - 2021

TD n°5 - Correction

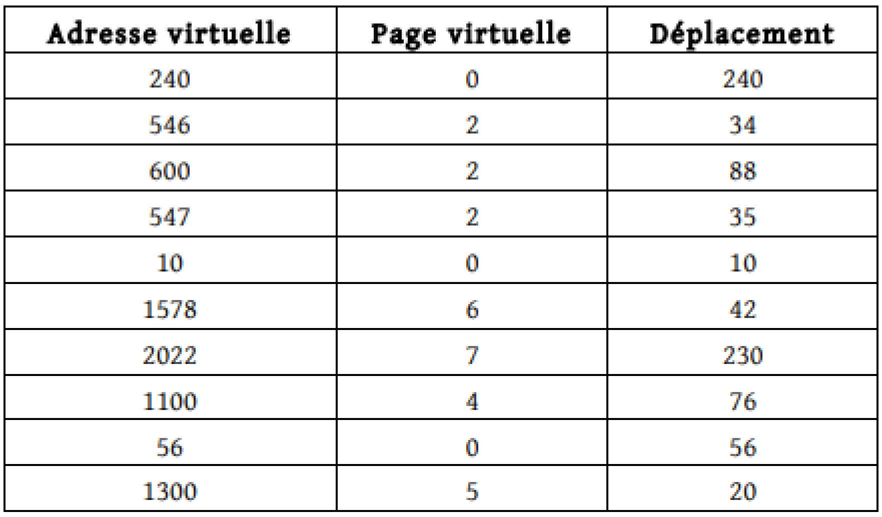
Exercise 1: manipulate the page table.

The physical memory of a computer is 1 KB. It is divided into four pages (from 0 to 3) of 256 bytes.

A process has a virtual space of 2 KB (i.e. eight pages, from 0 to 7) and successively refers to virtual addresses 240, 546, 600, 547, 10, 1578, 2022, 1100, 56, 1300.

1. Indicate, for each virtual address, the virtual page number and the value of the offset.

   Since pages are 256 bytes long, it is sufficient to take the quotient and remainder of the division of the address by 256.
   The virtual addresses are on 11 bits, of which 3 for the virtual page number and 8 for the offset.



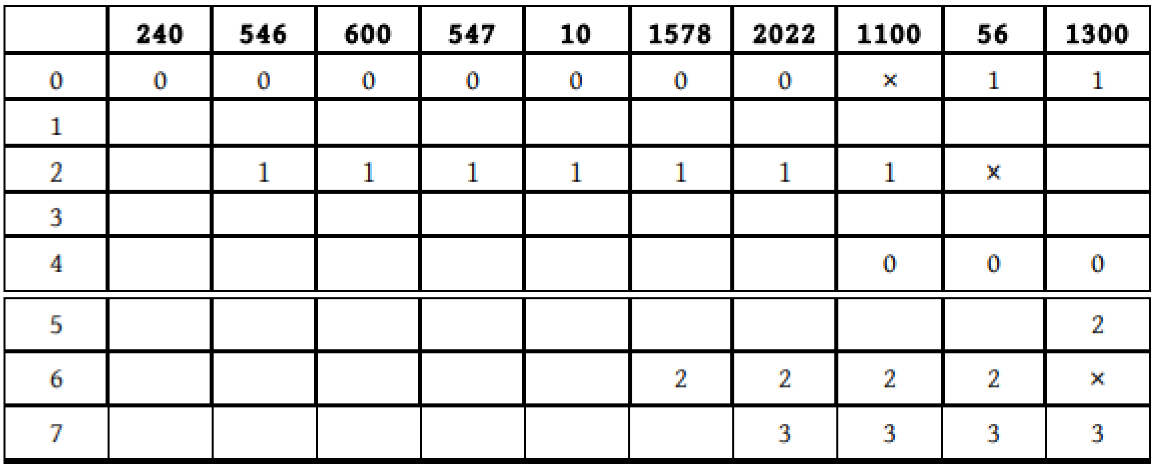
Adresse virtuelle	Page virtuelle	Déplacement
240	0	240
546	2	34
600	2	88
547	2	35
10	0	10
1578	6	42
2022	7	230
1100	4	76
56	0	56
1300	5	20

Correspondances entre adresses virtuelle et physique

2. Indicate, for each reference, the physical address that is accessed. Give the page table and the inverse table after each access for the FIFO replacement algorithm. How many page faults are there? The tables are empty at the beginning.

For the FIFO algorithm, the page table after each access is as follows. The references are made successively from left to right and each column gives the page table after the corresponding reference.

A cross (×) indicates the disappearance of the virtual page from physical memory.



	240	546	600	547	10	1578	2022	1100	56	1300
0	0	0	0	0	0	0	0	×	1	1
1										
2		1	1	1	1	1	1	1	×	
3										
4								0	0	0
5										2
6						2	2	2	2	×
7							3	3	3	3

Table des pages pour l'algorithme FIFO

When 240, located in virtual page 0, is accessed, this page is loaded into physical frame 0.
The access to 546 (virtual page 2) loads page 2 into physical frame 1.
The following accesses do not load a new page, until accessing 1578 (virtual page 6 comes into frame 2) and 2022 (virtual page 7 in frame 3).
When accessing 1100, a new virtual page (page 4) is reached.
Therefore, an old page must be evicted from the physical memory because there are no more free frames: the oldest page (FIFO algorithm) is page 0, located in frame 0.
Accessing 56 causes the eviction of page 2, replaced in frame 1 by page 0, and accessing 1300 does the same with page 6, replaced by page 5 in frame 2.

For the FIFO algorithm, the reverse table after each access is as follows:

	240	546	600	547	10	1578	2022	1100	56	1300
0	0	0	0	0	0	0	0	4	4	4
1		2	2	2	2	2	2	2	0	0
2						6	6	6	6	5
3							7	7	7	7

Table inverse pour l'algorithme FIFO

These are the physical addresses that were accessed:

	240	546	600	547	10	1578	2022	1100	56	1300
Adr.	240	290	344	291	10	554	998	76	312	532

Adresses physiques pour l'algorithme FIFO

There are seven page faults in total.
Only references 600, 547 and 10 do not cause any.

Exercise 2: overload the processor.

A computer has a main memory of five pages (from 0 to 4) and offers processes a virtual memory of ten pages, managed by the LRU algorithm.

1. Five processes are started. The one with the number 0 works only with page 0, the one with page 1, and so on. How many page faults are there?

   There is a page fault the first time a process references a page. After that, they are all in main memory.
   So, there are five page faults for the execution of all processes.

2. A sixth process with identical characteristics is started. How many page faults are there? What do you think about it?

   References are made in the order 0, 1, 2, 3, 4, then 5 which removes page 0, then 0 which removes 1, and so on. There is now a page fault with each reference! The system falls down: it spends its time bringing pages from disk into main memory at each memory reference, which slows down the execution of processes enormously.

The phenomenon may even be amplified. When the system detects a page fault, it starts the recovery of the virtual page on the disk and tries to execute another process. This will also cause a page fault.

Soon, all processes are in page faults and the CPU utilisation rate drops. It is then possible that the operating system will try to start other idle processes to increase CPU utilisation, causing even more page faults.

The crash gets worse, and the system spends most of its time switching processes and waiting for virtual pages from disk, without the processor being able to execute anything. One solution is to reduce page faults by killing or suspending some processes until the main memory is used properly.

Exercise 3: translate virtual addresses into physical addresses.

The virtual address translation simulator **SIMAV** is a program developed at the Ecole Polytechnique de Montréal that helps to understand the phenomenon of translating virtual addresses into physical addresses.

Using a simplified and imaginary schema of an architecture, composed of a virtual memory with 8-bit addresses and a physical memory addressed on 7 bits, students will be able to understand the principle on a simple basis and then extrapolate to more realistic systems.

It has two simulation modes: a single level of pages simulation or a two levels one.

Some guidance on how to use **SIMAV** is available at
https://cours.polymtl.ca/inf2610/simulateurs/Simav/HTML/aide.html.

After installing the **SIMAV** simulator on your machine (file to be copied available on Moodle), launch it, and answer the following questions with a level 1:

1. What is the number of virtual pages? 16
2. What is the number of physical frames? 8
3. What is the size of the page table? 16 x 4 bits = 64 bits
4. How big is a page and frame? 2^8 / 16 or 2^7 / 8 = 16 words
5. Do the processes share the same page table? No
6. What is the largest virtual address that can be used? $2^8 - 1 = 255$
7. What is the largest physical address that can be used? $2^7 - 1 = 127$
8. Converting address **102** without using the simulator:
   a. Give its value in binary: n° virtual page + offset = 0110 0110
   b. What is the value of the offset? 0110
   c. What is the corresponding virtual page? $0110_2 = 6_{10}$
   d. What would be the physical address if physical page number **2** is selected? n° physical frame + offset = 010 0110
9. Calculate by the simulator the addresses:
   a. Processor 1: 200 = 1100 1000 56 = 0011 1000 65 = 0100 0001
   b. Processor 2: 200 = 1100 1000 168 = 1010 1000 260 = impossible
   c. Processor 3: 200 = 1100 1000 158 = 1001 1110 22 = 0001 0110

   These are the virtual addresses, the physical addresses depending on the framework chosen for the storage.

Repeat the conversions of the addresses of the different processes (1, 2 and 3) carried out in question 8 with a level 2.