INSTITUTO SUPERIOR TÉCNICO

DEPARTAMENTO DE ENGENHARIA INFORMÁTICA

ORGANIZAÇÃO DE COMPUTADORES

LEIC

Conjunto de Exercícios VII

Memória Virtual

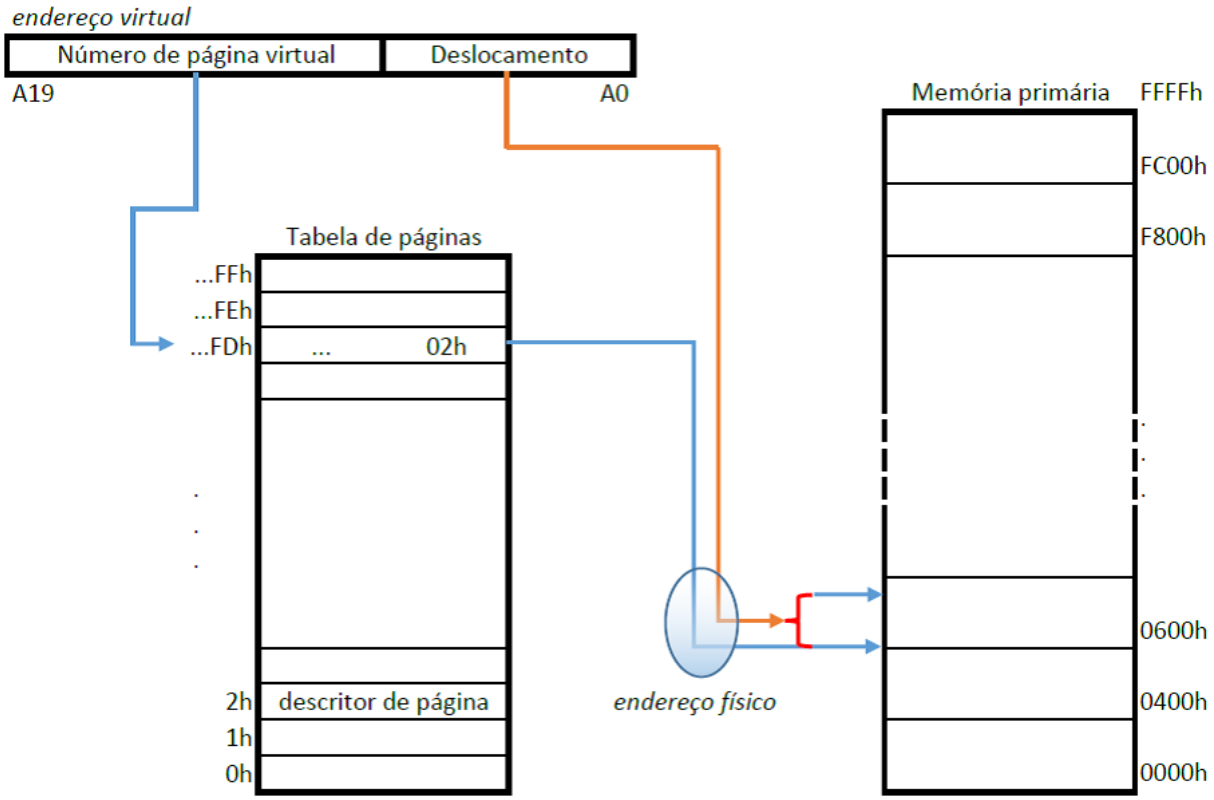
Versão 3.0

2022/2023

Exercício 1

Um processador de 16 bits tem um sistema de memória virtual paginada. O espaço de endereçamento virtual é de 1M palavras; o espaço físico tem 64K palavras; a página tem 1K palavras. O grão de endereçamento é a palavra de 16 bits.

A tradução de endereços é feita através de uma tabela de páginas simples (com um só nível) alocada na memória primária, como ilustrado na figura seguinte.



a) Indique a estrutura dos endereços virtual e físico (nome, número de bits e bit de menor peso de cada campo).

b) Indique a estrutura de um descritor de página armazenado na tabela de páginas.

c) Qual é o número máximo de páginas do espaço de endereçamento físico?

d) Quantas páginas físicas ocupa a tabela de páginas?

e) Em que índice da tabela de páginas está armazenado o descritor do endereço físico correspondente ao endereço virtual B34F2h?

f) Qual é o número mínimo de páginas físicas que devem estar disponíveis e carregadas na memória primária para executar sem faltas de página a sequência de acessos indicada na tabela seguinte (F – instruction fetch, R – data read, W – data write)? Justifique sucintamente.

Endereço virtual [h]	Tipo de acesso
08010	F
01000	R
08011	F

08012	F
08013	F
08014	F
08017	F
08018	F
02000	W
08019	F
0801A	F
0801B	F
0801C	F
0801D	F

Exercício 2*

2.1. The following table is a stream of virtual addresses in decimal notation as seen on a system. Assume 4 KB pages, a 4-entry fully associative TLB, and LRU replacement policy. If a page must be brought in from disk, the physical page is determined by incrementing the largest page number contained in the page table.

Virtual addresses	4669	227	13916	34587	48870	12608

a) Given the address stream in this table and the initial TLB and page table states presented below, show the final state of the system. Also list for each reference if it is: a hit in the TLB, a hit in the page table, or a page fault. Assume that the initial entries of the TLB were filled in the following order (by tag number): 4, 11, 7, and 3.

TLB

	Valid	**Tag**	**Physical Page Number**
0	1	11	12
1	1	7	4
2	1	3	6
3	0	4	9

Page Table

	Valid	**Physical Page or in Disk**
0	1	5
1	0	Disk
2	0	Disk
3	1	6
4	0	Disk
5	1	11
6	0	Disk
7	1	4
8	0	Disk
9	0	Disk

* Exercises 5.10.1 – 5.10.6 from the textbook

10	1	3
11	1	12

b) Repeat the previous exercise, but this time use 16 KB pages instead of 4 KB pages. What would be some of the advantages of having a larger page size? What are some of the disadvantages?

c) Show the final contents of the TLB if it is direct mapped. Assume 4 KB pages. Discuss the importance of having a TLB to achieve high performance. How would virtual memory accesses be handled if there were no TLB?

2.2. There are several parameters that impact the overall size of the page table. Listed below are several key page table parameters. The physical memory has the same size of the virtual memory space.

Parameter	Value
Virtual Address Size	32 bits
Page Size	4 KB
Page Table Entry Size	4 bytes

a) Given these parameters, calculate the total page table size for a system running 4 processes.

b) Given a two-level page table approach, calculate the page table size for a system running 4 processes. Assume that each page table entry is 4 bytes and that the first level table has 1024 entries. Assume also that the processes utilize half of the physical memory available and that the used up memory is evenly distributed among the processes.

c) A cache designer wants to increase the size of the virtually indexed, physically tagged cache installed in the system. Given the page size listed in the table above, is it possible to make a 8 KB directmapped cache, assuming 2 words per block? If not, how would the designer increase the data size of the cache?

Exercício 3*

3.1. In this exercise, we examine space / time optimizations for page tables. The following table shows parameters of a virtual memory system.

Virtual Address (bits)	Physical DRAM Installed	Page Size	PTE Size (bytes)
42	16 GB	4 KB	4

a) For a single-level page table, how many page table entries (PTEs) are needed? How much physical memory is needed for storing the page table?

b) Using multilevel page table can reduce the physical memory consumption of page tables, by only keeping active PTEs in physical memory. How many levels of page tables will be needed in this case? And how many memory references are needed for address translation if missing in TLB?

* Exercises 5.11.1 – 5.11.6 from the textbook

c) An inverted page table can be used to further optimize space and time. How many PTEs are needed to store the page table? Assuming a hash table implementation, what are the best case and the worst case numbers of memory references needed for servicing a TLB miss?

3.2. The following table shows the contents of a 4-entry TLB. (RO - read only, RW - read-write, RX - read-execute).

Entry-ID	Valid	VA Page	Modified	Protection	PA Page
1	1	140	1	RW	30
2	0	40	0	RX	34
3	1	200	0	RO	32
4	1	280	0	RW	31

a) Under what scenarios would entry 2's valid bit be set to zero?

b) What happens when an instruction writes to VA page 30?

c) What happens when an instruction writes to VA page 200?

References

[1] David Patterson and John Hennessy. *Computer Organization and Design: The Hardware/Software Interface*. Morgan Kaufmann, 4th edition, 2011.

Conjunto de Exercícios VII

Memória Virtual

Guia de Resolução

Exercício 1

a) O endereço virtual tem 20 bits (2^{20} = 1 M):

- Cada página tem 1 K (2^{10}) pelo que são necessários 10 bits para indexar o seu conteúdo. Assim, A0 – A9 (10 bits) são usados como deslocamento na página
- Os restantes bits (20-10=10), A10 – A19, identificam o número da página virtual

O endereço físico tem 16 bits (2^{16} = 64 K):

- Como cada página tem 1 K, existem 64 páginas (2^6)
- A0 – A9 (10 bits), deslocamento na página
- A10 – A15 (6 bits), número de página física

b) A tabela de páginas tem 1 K entradas indexadas pelo número de página virtual. Cada entrada contém um descritor de página com:

- número da página física onde está carregada a página virtual:

  $\log_2$ 64 (número de páginas) = 6 bits;
- bits de direitos de acesso, protecção e estado (presente, acedida, modificada, etc.).

c) 64 K palavras / 1 K palavras = 64 páginas.

d) Em princípio uma palavra de 16 bits é suficiente para armazenar toda a informação de controlo e de estado associada à página. Logo a tabela de páginas ocupa uma página de memória primária.

e) B34F2h = 1011.0011.0100.1111.0010b

Os 10 bits de maior peso (A19 – A10) correspondem ao índice da tabela, 10 1100 1101b = 2CDh .

f) Cada página com 1 K palavras cobre uma gama de endereços de XY000h – XY3FFh, XY400h – XY7FFh, ...

As gamas de endereços 08000h – 083FFh, 01000h – 013FFh e 02000h – 023FFh serão mapeadas em 3 páginas distintas.

É necessária mais uma página física para carregar a tabela de páginas.

Logo será necessário ter, pelo menos, 4 páginas em memória primária para não ocorrerem faltas de página ao executar este rasto.

Normalmente, para além dos programas dos utilizadores, um computador possui um sistema operativo precisamente para controlar a execução dos programas dos utilizadores. O sistema operativo também necessita de memória para se executar pelo que em condições mais realistas seriam necessárias mais do que 4 páginas para não ocorrerem faltas de página.

Exercício 2

2.1.

a) The table below shows the changes to the TLB and to the PT.

Access	Page No	TLB (hit/miss)	PT (hit/fault)	TLB	PT
4669	1	miss	fault	3:<1,1,13>	1:<1,13>
227	0	miss	hit	0:<1,0,5>	-
13916	3	hit	hit	-	-
34587	8	miss	fault	1:<1,8,14>	8:<1,14>
48870	11	miss	hit	3:<1,11,12>	-
12608	3	hit	hit	-	-

b) The table below shows the changes to the TLB and to the PT.

Access	Page No	TLB (hit/miss)	PT (hit/fault)	TLB	PT
4669	0	miss	hit	3:<1,0,5>	-
227	0	hit	hit	-	-
13916	0	hit	hit	-	-
34587	2	miss	miss	0:<1,2,13>	2:<1,13>
48870	2	hit	hit	-	-
12608	0	hit	hit	-	-

Advantages: more TLB and PT hits. Disadvantages: fragmentation.

c) The table below shows the changes to the TLB and to the PT.

Access	Page No	Entry	TLB (hit/miss)	PT (hit/fault)	TLB	PT
4669	1	1	miss	fault	1:<1,1,13>	1:<1,13>
227	0	0	miss	hit	0:<1,0,5>	-
13916	3	3	miss	hit	3:<1,3,6>	-
34587	8	0	miss	miss	0:<1,8,14>	8:<1,14>
48870	11	3	miss	hit	3:<1,11,12>	-
12608	3	3	miss	hit	3:<1,3,6>	-

Without a TLB, the performance would be severely degraded, because each memory access would require an additional access to read the page table and determine the requested physical address.

2.2.

a) TablesSize = NumProcs * PTESize * VASize / PageSize = $4 * 4 * 2^{32} / 2^{12} = 2^{24}$ = 16 MB

b) First, determine the number of pages used by each process:

NumPagesProc = ((UsedVASize / 2) / NumProcs) / PageSize) = $2^{32} / (2 * 4 * 2^{12}) = 2^{20} / 2^3 = 2^{17}$ pages

Since each page has an entry on a 2-level page table, the total number of pages for the page table of each process is:

TotalPagesPTProc = 1 + NumPagesProc / (PageSize / PTESize) = 1 + 2^{17} / (2^{12} / 2^{2}) = 1 + 2^{7} = 1 + 128 pages

Therefore, the total size allocated to page tables is:

TableSize = NumProc * PTESize * TotalPagesPTProc = 4 * 4KB * (1 + 128) = 2064 KB

c) The *virtual offset* (within a page) ≥ index + offset (fed into the cache).

For 4K pages, virtual offset defined by 12 bits.

Since the cache has 2-word blocks, hence 8-byte blocks, the offset has 3 bits and the index can be defined at most by 12 - 3 = 9 bits.

20 bits	12 bits	
Physical Page Number	Page Offset	
Tag	Index	Block Offset
20 bits	9 bits	3 bits

Therefore, since the maximum size for the direct-mapped memory is 2^{9} * 2^{3} = 4 KB, the designer could not have an 8 KB direct-mapped cache.

To increase the size, it is necessary to have 2-way associativity.

Exercício 3

3.1.

a) TotalPTEs = VASize / PageSize = 2^{42} / 2^{12} = 2^{30} = 1G PTEs

TotalPhysMem = PTESize * TotalPTEs = 4 * 1G = 4GB

b) If we have 4KB pages and 4 bytes per page, we can have 1024 PTEs per page, and therefore we need 10 bits to index each PTE table.

Since, 42 = 10 + 10 + 10 + 12, we need 3 levels.

Virtual Address

Directory	2nd Level	3rd Level	Page Offset
10 bits	10 bits	10 bits	12 bits

In case of TLB miss, we need 3 memory references to serve the TLB miss.

c) We need one PTE per physical page, therefore PHSize / PageSize = 2^{34} / 2^{12} = 4M PTEs.

To serve the TLB miss, the best case is 1 memory reference, the worst case is 4M (need to traverse the entire chain).

3.2.

a) TLB initialization or process context switch.

b) TLB miss.

c) Write protection exception.