Exercise 1

C) While processing the Present (P) bit in Page-Translation-Table Entry Fields (see Section 5.4.1), one of the entries in Table 8-1 is particularly relevant. Which entry is that, and why?

5.4.1 Filed Definitions …

Present (P) Bit. Bit 0. This bit indicates whether the page-translation table or physical page is loaded in physical memory. When the P bit is cleared to 0, the table or physical page is not loaded in physical memory. When the P bit is set to 1, the table or physical page is loaded in physical memory.

Software clears this bit to 0 to indicate a page table or physical page is not loaded in physical memory. A page-fault exception (#PF) occurs if an attempt is made to access a table or page when the P bit is 0. System software is responsible for loading the missing table or page into memory and setting the P bit to 1.

When the P bit is 0, indicating a not-present page, all remaining bits in the page data-structure entry are available to software.

Entries with P cleared to 0 are never cached in TLB nor will the processor set the Accessed or Dirty bit for the table entry.

…

Exception/Interrupt When P Bit is 0

Quando o bit **Present (P)** é **0** e existe uma tentativa de acesso a página na memória física, o processador aciona a exceção de **Page-fault**, vector 14. Este tipo de interrupção sinaliza ao sistema operativo que a página (ainda) não está presente na memória física. Este, carrega então a página em falta, reescreve o bit **P** a 1 e "manda" repetir a operação que gerou a interrupção.

Este mecanismo é fundamental para a implementação do espaço de endereçamento virtual de um processo, onde nem todas as páginas necessitam ser carregadas na memória física ao mesmo tempo. E, de início nem uma está. Vão sendo carrgadas conforme vão dando os *page-fault*, permitindo, assim, ao sistema operativo uma gestão da memória de forma eficiente, carregando as páginas a pedido e trocando-as para dentro e fora da memória física conforme necessário (*swap in / swap out*).

Table 8-1. Interrupt Vector Source and Cause

Vector	Exception/Interrupt	Mnemonic	Cause
0	Divide-by-Zero-Error	#DE	DIV, IDIV, AAM instructions
1	Debug	#DB	Instruction accesses and data accesses
2	Non-Maskable-Interrupt	#NMI	External NMI signal
3	Breakpoint	#BP	INT3 instruction
4	Overflow	#OF	INTO instruction
5	Bound-Range	#BR	BOUND instruction
6	Invalid-Opcode	#UD	Invalid instructions
7	Device-Not-Available	#NM	x87 instructions
8	Double-Fault	#DF	Exception during the handling of another exception or interrupt
9	Coprocessor-Segment-Overrun	—	Unsupported (Reserved)
10	Invalid-TSS	#TS	Task-state segment access and task switch
11	Segment-Not-Present	#NP	Segment register loads
12	Stack	#SS	SS register loads and stack references
13	General-Protection	#GP	Memory accesses and protection checks
14	Page-Fault	#PF	Memory accesses when paging enabled
15	Reserved	—	
16	x87 Floating-Point Exception-Pending	#MF	x87 floating-point instructions
17	Alignment-Check	#AC	Misaligned memory accesses
18	Machine-Check	#MC	Model specific
19	SIMD Floating-Point	#XF	SSE floating-point instructions
20	Reserved	—	
21	Control-Protection Exception	#CP	RET/IRET or other control transfer
22–27	Reserved	—	
28	Hypervisor Injection Exception	#HV	Event injection
29	VMM Communication Exception	#VC	Virtualization event
30	Security Exception	#SX	Security-sensitive event in host
31	Reserved	—	
0–255	External Interrupts (Maskable)	#INTR	External interrupts
0–255	Software Interrupts	—	INT*n* instruction

Figure 1: Table 8-1