EPT Initialization

בתרגיל הקודם נכנסו ויצאנו למצב VMX עם אתחול בסיסי של מבני הנתונים הרלוונטיים.
כעת, אנחנו נרחיב את ה Hypervisor לתמיכה גם ב Extended Page Tables !

DRIVER

כפי שניתן לראות השינוי שהתווסף הוא הקריאה לפונקציה Initialize_EPTP מהפונקציה DriverEntry.
באופן כללי, כדאי להכיר בגדול את ה paging ב x64 (CR3 -> PML4 -> PDPT -> PD -> PT), יש גם אפשרות לקצר את התהליך מ 4 שכבות ל 2 או (בעזרת איפוס ביטיים ייעודים) כדי לחסוך בזמנים אך לא נעשה את זה במימוש שלנו.

Initialize_EPTP

המטרה שלנו הינה מימוש "תוכנתי" למנגנון ה paging כלומר אתחול המבנים וההרשאות בשביל המעבד.
ראשית, נקצה את כל הטבלאות ההכרחיות שציינו בזיכרון non-paged (כמובן) בגודל PAGE כל אחת.
שנית, נאתחל עשרה PTE-ים כ RWX (לא אכפת לנו מהרשאות כרגע יש הרשאות פנימיות ב VM paging), פרקטית היינו צריכים רק שניים (הרצת קוד - RIP, ומחסנית - RSP).
ולבסוף, נאתחל את כל שאר השכבות שמעלינו עם אתחולים דיפולטיביים (לא אסביר מה כל אחד אומר - יש Manual).

עוצרים בשלב זה, אך ברור שנצטרך לשמור את EPTP ב VMCS הרלוונטי שנרצה להתשתמש ב EPT ב VM.

_EPTP

ניתן לראות את ההגדרה ב manual.
אציין שביט 6 די מגניב כיוון שאם הוא דלוק, המעבד ידליק את הביטים המתאימים (בטבלאות ה EPT)
במידה ונגשו \ כתבו לטווח כתובות אליו ה entry רלוונטי מצביע + ניתן לאפס אותם (נותן אינדיקציה על מה קרה ב VM).

Table 24-8. Format of Extended-Page-Table Pointer

Bit Position(s)	Field
2:0	EPT paging-structure memory type (see Section 28.2.6): 0 = Uncacheable (UC) 6 = Write-back (WB) Other values are reserved.[1]
5:3	This value is 1 less than the EPT page-walk length (see Section 28.2.2)
6	Setting this control to 1 enables accessed and dirty flags for EPT (see Section 28.2.4)[2]
11:7	Reserved
N–1:12	Bits N–1:12 of the physical address of the 4-KByte aligned EPT PML4 table[3]
63:N	Reserved

_EPT_PTE

אציין כאן את המבנה של ה PTE כיוון שכל שאר השכבות (PML4E, PDPTE, PDE) נשארות בעלי אותו מבנה כמו paging רגיל אך ה ה PTE מעט שונה.

Table 28-6. Format of an EPT Page-Table Entry that Maps a 4-KByte Page

Bit Position(s)	Contents
0	Read access; indicates whether reads are allowed from the 4-KByte page referenced by this entry
1	Write access; indicates whether writes are allowed from the 4-KByte page referenced by this entry
2	If the "mode-based execute control for EPT" VM-execution control is 0, execute access; indicates whether instruction fetches are allowed from the 4-KByte page controlled by this entry If that control is 1, execute access for supervisor-mode linear addresses; indicates whether instruction fetches are allowed from supervisor-mode linear addresses in the 4-KByte page controlled by this entry
5:3	EPT memory type for this 4-KByte page (see Section 28.2.6)
6	Ignore PAT memory type for this 4-KByte page (see Section 28.2.6)
7	Ignored
8	If bit 6 of EPTP is 1, accessed flag for EPT; indicates whether software has accessed the 4-KByte page referenced by this entry (see Section 28.2.4). Ignored if bit 6 of EPTP is 0
9	If bit 6 of EPTP is 1, dirty flag for EPT; indicates whether software has written to the 4-KByte page referenced by this entry (see Section 28.2.4). Ignored if bit 6 of EPTP is 0
10	Execute access for user-mode linear addresses. If the "mode-based execute control for EPT" VM-execution control is 1, indicates whether instruction fetches are allowed from user-mode linear addresses in the 4-KByte page controlled by this entry. If that control is 0, this bit is ignored.
11	Ignored
(N–1):12	Physical address of the 4-KByte page referenced by this entry[1]
51:N	Reserved (must be 0)
62:52	Ignored
63	Suppress #VE. If the "EPT-violation #VE" VM-execution control is 1, EPT violations caused by accesses to this page are convertible to virtualization exceptions only if this bit is 0 (see Section 25.5.6.1). If "EPT-violation #VE" VM-execution control is 0, this bit is ignored.

כעת, הריצו ובדקו שלא מקבלים BSOD, ושמקבלים את ההודעות דיבוג המתאימות...

יש לנו VM בסיסי ביותר (שעוד לא רץ) ולא באמת עושה משהו, אך הכנו את הקרקע לשימוש ב EPT :)