**סיכום פסיקות חשובות**

## פסיקות שעון (פסיקה מספר 8 )

על לוח האם של ה- PC מצוי שבב ה- timer שמיצר 18.2 פסיקות בשניה.

באופן סטנדרטי, ה- timer מחובר ל- PIC בצורה כזו שהוא מיצר פסיקה שעדיפותה היא הגבוהה ביותר מבין כל פסיקות החומרה, ומספר הוקטור שלה – 8.

ניתן להשתמש בפסיקות השעון לצורך ספירת זמן לשימושן של אפליקציות. למשל, תכניות יוכלו להגביל את ריצתן על פי זמן. השגרה של פסיקה 8 נמצאת ב- BIOS. היא מעלה את מוני הזמן ומודיעה ל- PIC על סיום הטיפול בפסיקה (end of interrupt) כדי שזה יאפשר לפסיקות השעון הבאות להגיע אל ה- : CPU

MOV AL,20 OUT 20h,AL

טיפול לא מדויק בפסיקה זו עלול לשבש את ההגעה של הפסיקות הבאות, ולכן לא נהוג להחליף את ISR 8 הסטנדרטית בשגרת משתמש.

### המקלדת

בקר המקלדת מיצר פסיקת חומרה עם כל לחיצה או עזיבה של מקש. באופן סטנדרטי, הבקר מחובר ל- PIC בצורה כזו שהוא מיצר פסיקה שמספרה – 9.(פסיקה 9 היא פסיקת **חומרה** של המקלדת)

ISR 9 של ה- BIOS קוראת scan code, שהוא בית אחד, מה- port של המקלדת, מתרגמת אותו לקוד ascii תוך התחשבות בדף התרגום המוגדר באותו זמן במערכת (אנגלית, רוסית, וכו'), ומעבירה את ה- scan code ואת ה- ascii לחוצץ של המקלדת, הקרוי PC type ahead buffer. זהו תור מעגלי בן 16 מילים בזכרון.

**The scan code is the code of the key in the keyboard and the ascii is the code for a symbol, in this way you can have the same key configured in different ways for different keyboards**

כל מילה בתור כוללת בבית הגבוה שלה את ה- scan code, ובבית הנמוך את קוד ה- ascii.

כאמור, המקלדת מייצרת פסיקה עם כל לחיצה וכל עזיבה של מקש. לא כל אירוע כזה גורם להכנסת תו לחוצץ המקלדת. למשל ללחיצה או לעזיבה של shift או ctrl אין משמעות של תו חדש שהוקלד, אך היא תשפיע על המשמעות של התו הבא שיוקלד. רוטינת השרות ISR 9 עוקבת אחרי מצב הלחיצה של מקשי בקרה שונים בעזרת דגלי בקרה שהיא מנהלת בשני הבתים שבכתובות 0417h ו- 0418h.

**KYBORD STATUS**

מצב המקלדת נשמר ע"י BIOS בשני בתים בזיכרון בכתובות קבועות.

KB\_FLAG SEGMENT:0 OFFSET 417h

KB\_FLAG\_1 SEGMENT:0 OFFSET 417h (ניתן לראות את המבנה בעמוד 30 בחוברת.)

כאשר מגיעה פסיקה עם scan code שמשמעותו לחיצה על מקש שהוא תו, ISR 9 מוצאת את ה- ascii המתאים לו על פי מצב הדגלים (למשל, אם shift לחוץ – זו אות גדולה) ועל פי דף התרגום הנוכחי.

אם אין מקום בתור להכנסת תו נוסף, משמיעה ISR 9 צפצוף, ואינה מכניסה את התו והוא אובד. תוכניות שצריכות לקרוא תוים מהמקלדת (בדרך כלל אלו תוכניות שרות שמופעלות כפסיקות תוכנה

int 21h ו- int 16h) ניגשות אל התור, ולוקחות את התווים משם.

לחיצה ממושכת על מקש מתורגמת ע"י בקר המקלדת למספר לחיצות, על פי מרווח זמן שניתן לכוונון. כלומר, מיוצרות פסיקות כאילו המקש נעזב וחזר ונלחץ מספר פעמים.

פסיקת המקלדת מאופשרת תמיד. אם נחסום אותה על ידי ה- PIC, או cli, לא נוכל להקיש תו כלשהו, אפילו לא alt-ctrl-del .

## שירותי מערכת שניגשים לתור המקלדת(פסיקת תוכנה של המקלדת INT16H)

תוכנית משתמש יכולה לגשת ישירות לתור המקלדת, תוך ידיעת מבנהו.ה- ISR השונים של ה- BIOS, בעיקר int 16h, מאפשרים גישה קלה אל התו הבא בתור במספר אופנים:

int 16h, ah=0: מוציאה את התו שבראש התור, ומחזירה ב- al את קוד ה- ascii שלו(קוד ה-asci הוא בדרך כלל מה שאנחנו צריכים), וב- ah את קוד ה- scan שלו(מספר יחודי לכל מקש 1-127). התו אינו מופיע על המסך; אין echo אוטומאטי. המסך הוא התקן נפרד מהמקלדת, וכדי לראות עליו משהו, יש לכתוב אליו במפורש. אם התור ריק, ה- ISR מחכה ומחכה (והתכנית שקראה לה אינה מתקדמת) עד שיגיע תו .

Int 16h, ah=1: בודקת אם התור ריק אם לאו, ומיד חוזרת. אם ריק, חוזרת עם ZF מורם, ואחרת חוזרת עם ZF מורד, ועם פרטי התו שבראש התור ב- ax בדומה לעיל, אלא שהתו אינו מוצא מהתור.

Int 16h, ah=2: מחזירה ב- al את דגלי הבקרה שבכתובת 0417h. את הדגלים שבכתובת 0418h יש לקרוא ישירות מהזיכרון. אין שרות BIOS לבית זה.

רוטינות השרות של DOS מאפשרות גישה ברמה גבוהה עוד יותר, עם שפע אפשרויות. לדוגמא:

int 21h, ah=1: מוציאה את התו שבראש התור, ומחזירה את קוד ה- ascii שלו ב- al. כמו כן מתבצע echo למסך. גם נבדק אם הקומבינציה שהוקלדה היא ctrl-break, ואם כן, מתבצעת הפקודה int 23h.

אם התור ריק, ה- ISR ממתינה שיגיע תו.

Int 21h, ah=8: כמוah=1 , אבל אינה מבצעת echo.

Int 21h, ah=7: כמוah=1 , אבל אינה מבצעת echo, וגם לא בודקת ctrl-break.