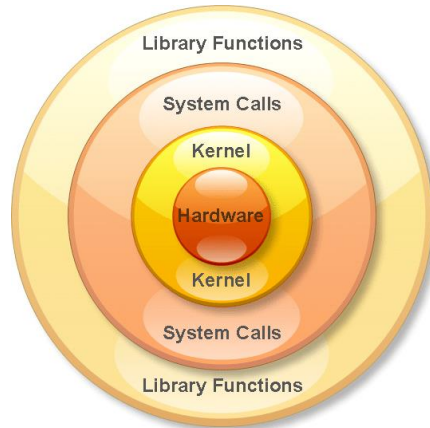


## יצירת קלט פלט



## מודל קליפות הבצל

(1) ניתן ישירות לתקשר מול התקן קלט הפלט I/O (או הבקר של התקן) בעזרת פקודות אסמבלי **in out** אל מול כתובות בבקר הנקראות גם port או מרחב כתובות קלט ופלט. ההתקנים נקראים גם התקני סביבה (peripheral). כיום פניה ישירה להתקן מורכבת מהמון פרטים טכניים ולא פרקטית.

(2) במעטפת הראשונה להתקני קלט פלט סטנדרטיים יש טיפול ברמת ה BIOS , או ROM BIOS שהוא קושחה - firmware המכילה קטע קוד ב flash (בעבר היה צרוב) הדואג לאתחול ראשוני של המחשב עם עלייתו POST (power on self test). כמו כן הוא מכיל פרוצדורות וקטעי קוד לטיפול בכל התקני קלט הפלט הסטנדרטיים (Basic Input/Output System-BIOS) לאורך השנים ה BIOS מתעדכן לתמוך בפרוטוקולי USB ופרוטוקולי תצוגה גרפיים.

(3) במעטפת השנייה יש פרוצדורות system call הכתובות בגרעין של מערכת ההפעלה (ונשארות בזיכרון עם עליית מערכת ההפעלה). פרוצדורות אלו של מערכת ההפעלה מטפלות בכל הקשור לשירותי קלט ופלט. במסגרת תפקידי ניהול מערכת ההפעלה קיימות system call לניהול זיכרון, נהול תהליכים, ניהול קבצים ועוד. אנחנו נתמקד בשירותי קלט ופלט.

(4) במעטפת הבאה ישנם פונקציות ספריה מוכנות לעבודה עם קלט ופלט כמו printf או scanf.

חשוב להבין שכאשר אנחנו מפעילים פונקציית ספריה להתקן קלט פלט סטנדרטי אז מופעלת ה system call המתאימה של מערכת ההפעלה הקוראת לפרוצדורת ה BIOS המתאימה המתקשרת מול התקן קלט פלט בפניה ישירה ל port במרחב זיכרון קלט פלט.

כמובן שיש התקני קלט ופלט ייחודיים המצריכים התקנת driver ספציפי להתקן.

אנחנו בסביבת העבודה של MARS נשתמש במעטפת השנייה של מערכת ההפעלה system call ליצירת קלט ופלט.

# SYSCALL

**Syscall** זו פקודת אסמבלי (מכונה) של מעבד ה-MIPS הגורמת לתכנית להפסיק ע"י חריגה (exception), ולעבור לקטע קוד בגרעין של מערכת ההפעלה הנקרא גם exception handler, ומשם אנחנו מסתעפים לפי הערך של אוגר \$v0 לבצע את ה syscall המתאים.

ראו הסבר מקיף ב help של ה MARS על כל נושא ה syscall

כמו כן אנחנו נסמך איש syscall לקבוצת המעבדה לתמיכה בכל נושא זה.

א על מנגנון הפסיקות חריגות (Exception/ Interrupt) נלמד בהמשך הקורס.

Service	\$v0	Arguments / Result
Print Integer	1	\$a0 = integer value to print
Print String	4	\$a0 = address of null-terminated string
Read Integer	5	\$v0 = integer read
Read String	8	\$a0 = address of input buffer \$a1 = maximum number of characters to read
Exit Program	10	
Print Char	11	\$a0 = character to print(low order byte)
Read Char	12	\$v0 = character read