מערכות הפעלה

1

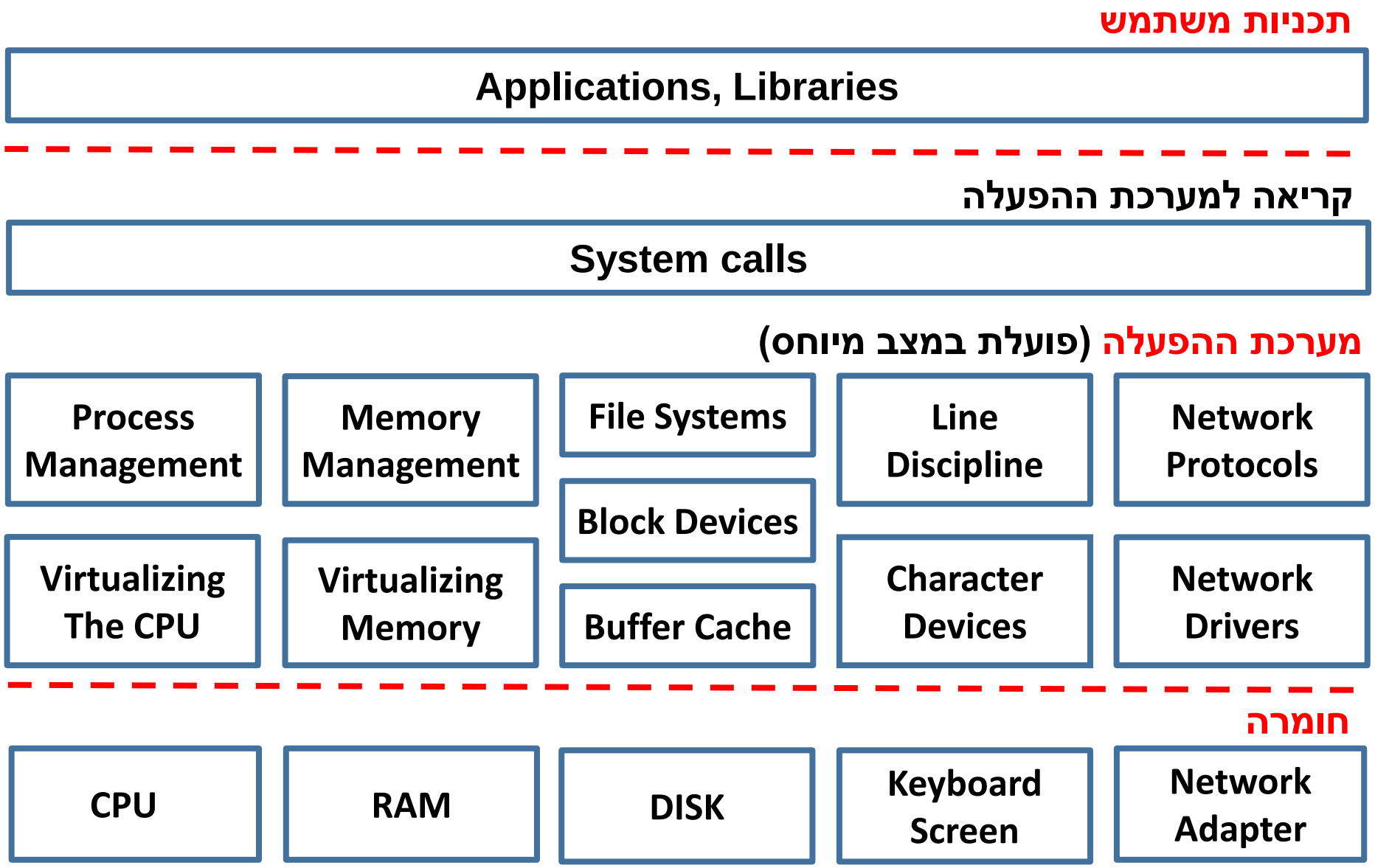
מבוא
פסיקות
קריאות מערכת

מהי מערכת הפעלה ?

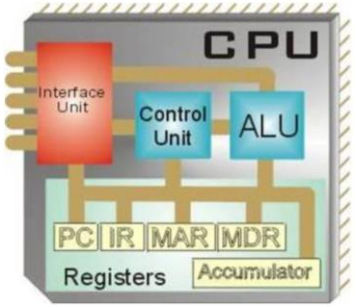
תוכנה שפועלת כמתווך בין המשתמש במחשב לבין חומרת המחשב

תפקידי מערכת ההפעלה:

- ליצור ממשק שמאפשר להשתמש בקלות בחומרת המחשב
- לחלק את משאבי המחשב: מעבד, זיכרון והתקני חומרה, בין התכניות הרצות.
- לבודד תכניות ומשתמשים זה מזה



מבנה מחשב



מעבד (CPU)

יחידה אריתמטית לוגית **(ALU)**

PC

STATUS

STACK

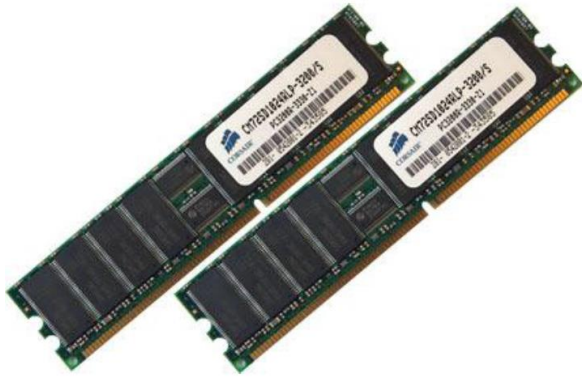
אוגרים

R0
R1
R2
Rn-1

זיכרון (RAM)

Instruction
Instruction
Instruction

Data
Data
Data



התקני קלט/פלט: מצלמה, מיקרופון, רמקול, ...

Bus

יצירת קובץ ריצה, טעינת תכנית לזיכרון

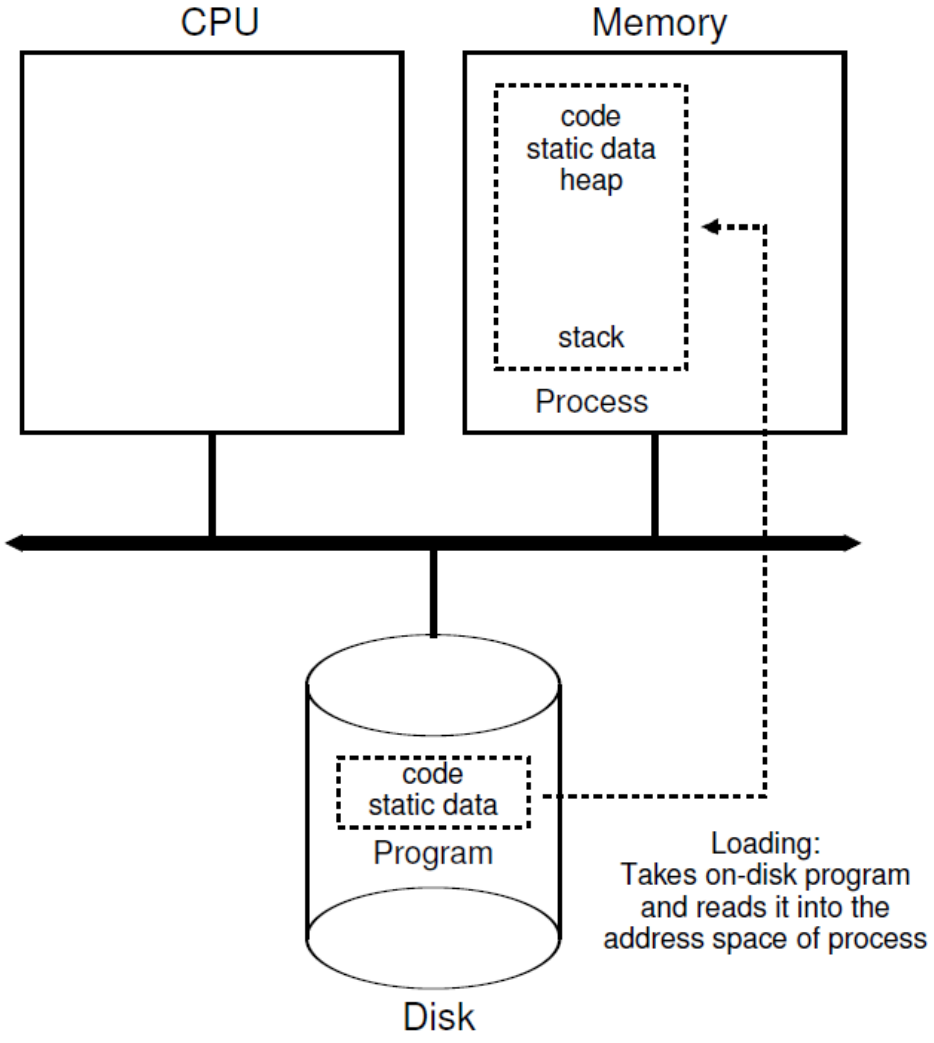
```
swap(int v[], int k)
{int temp;
   temp = v[k];
   v[k] = v[k+1];
   v[k+1] = temp;
}
```

Compiler

```
swap:
      multi $2, $5,4
      add   $2, $4,$2
      lw    $15, 0($2)
      lw    $16, 4($2)
      sw    $16, 0($2)
      sw    $15, 4($2)
      jr    $31
```

Assembler

```
00000000101000100000000100011000
00000000100000100001000000100001
10001101111000100000000000000000
10001110000100100000000000000100
10101110000100100000000000000000
10101101111000100000000000000100
00000011111000000000000000001000
```



שלבים בביצוע פקודות מכונה

1. הבאת הפקודה שה- PC מצביע עליה, מהזיכרון למעבד

Fetch Instruction

2. פיענוח הפקודה וקביעת הפעולה הנדרשת.

Decode Instruction

3. הבאת הנפעלים למעבד.

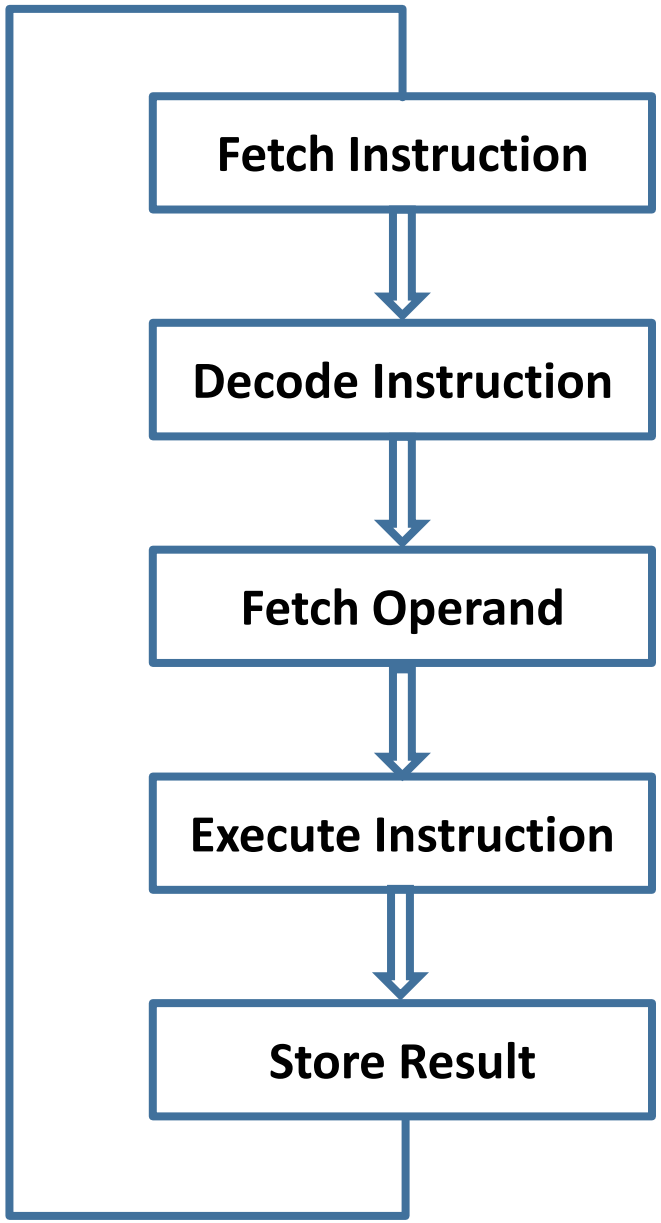
Fetch Operands

4. ביצוע הפקודה.

Execute Instruction

5. שמירת התוצאה.

Store Result



פסיקות

- פסיקה היא מאורע שגורם למעבד (CPU) להפסיק את ביצוע התכנית.
- המעבד עובר לבצע קוד (Interrupt handler) במערכת ההפעלה שמטפל באותו מאורע.
- הטיפול בפסיקה:

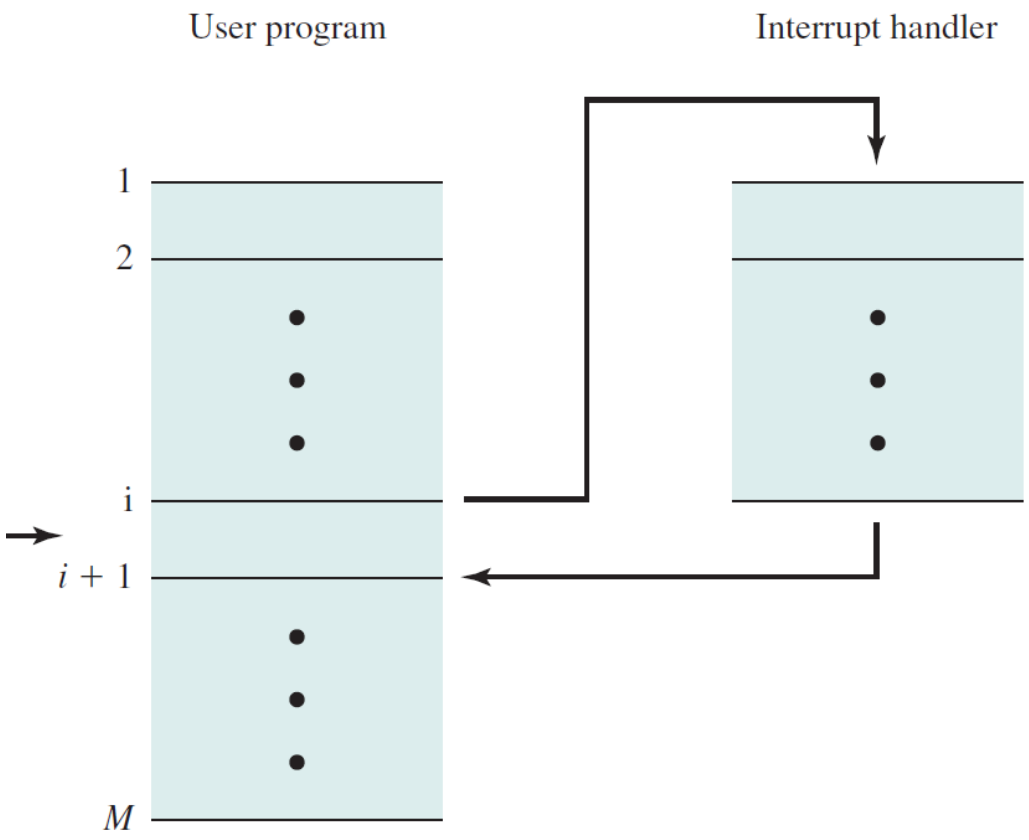
שמירת ה- PC, אוגר הסטטוס, אוגר המחסנית והאוגרים הכלליים של התכנית שהופסקה.

ביצוע הקוד המטפל בפסיקה.

החזרת ערך ה- PC, הסטטוס, המחסנית והאוגרים, וחזרה להמשך הריצה של התכנית שהופסקה.

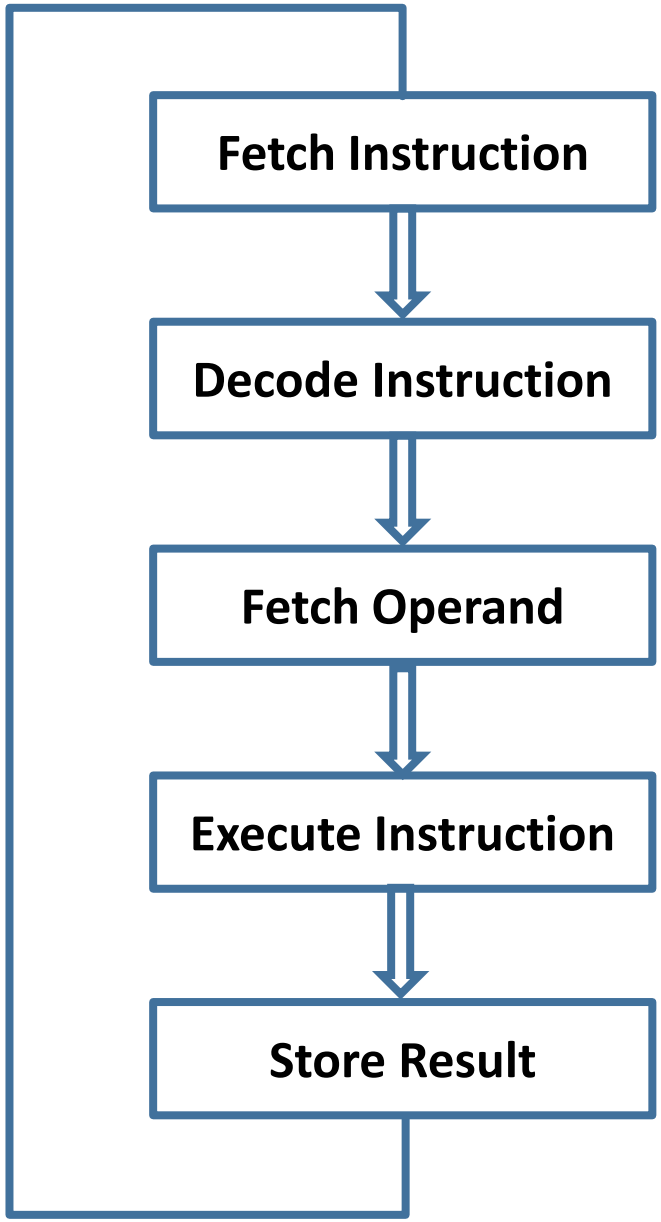
- ישנם שלושה סוגי פסיקות:

1. פסיקות פנימיות (Exceptions)
2. פסיקות קריאה למערכת ההפעלה (system call)
3. פסיקות חיצוניות (Interrupts)



1. פסיקות פנימיות (Exceptions)

- **פסיקות פנימיות** הן אירועים לא צפויים במהלך ביצוע הפקודה:
- **Fetch Instruction**
  - זיכרון לא קיים או לא נגיש
- **Decode Instruction**
  - פקודה לא מוגדרת
- **Fetch Operand**
  - זיכרון לא קיים או לא נגיש
- **Execute Instruction**
  - חלוקה ב- 0
  - גלישה
- **Store Result**
  - זיכרון לא קיים או לא נגיש



2. פסיקות קריאה למערכת ההפעלה (system call)

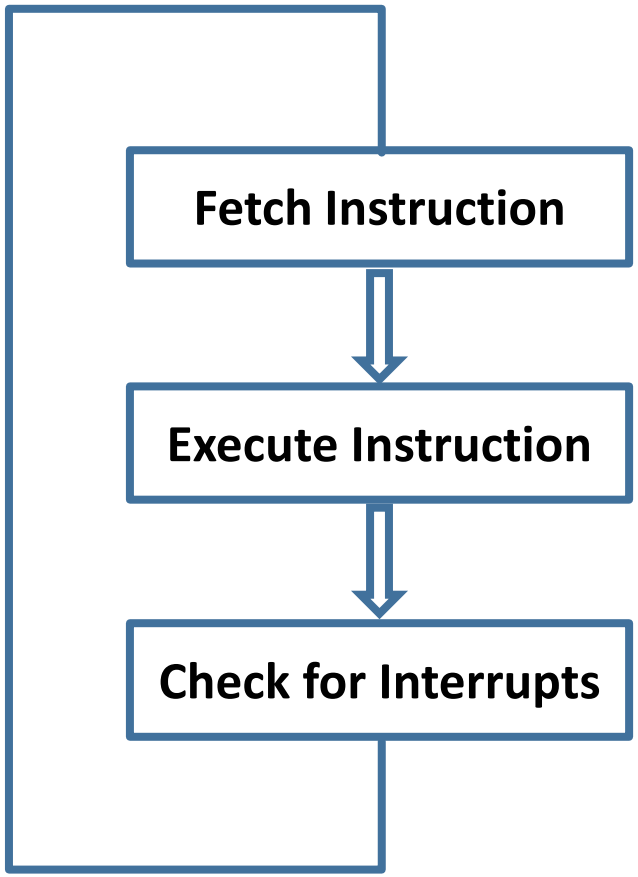
- ישנה פקודת מכונה שהביצוע שלה גורם לפסיקה, פקודה זו נקראת system call.
- כאשר תכנית משתמש מבקשת שירות ממערכת ההפעלה, כגון קריאת קובץ, היא תבקש זאת באמצעות הפקודה system call.
- מאחר שמערכת ההפעלה מספקת מאות שירותים לתכניות משתמש, כשתכנית מבצעת את הפקודה system call היא מעבירה פרמטרים שאומרים למערכת ההפעלה מה השרות הדרוש.
- בדרך כלל תכנית לא מפעילה ישירות את הפקודה system call, אלא מפעילה פונקציית ספריה, כגון ()read, שמפעילה את הפקודה.

3. פסיקות חיצוניות (Interrupts)

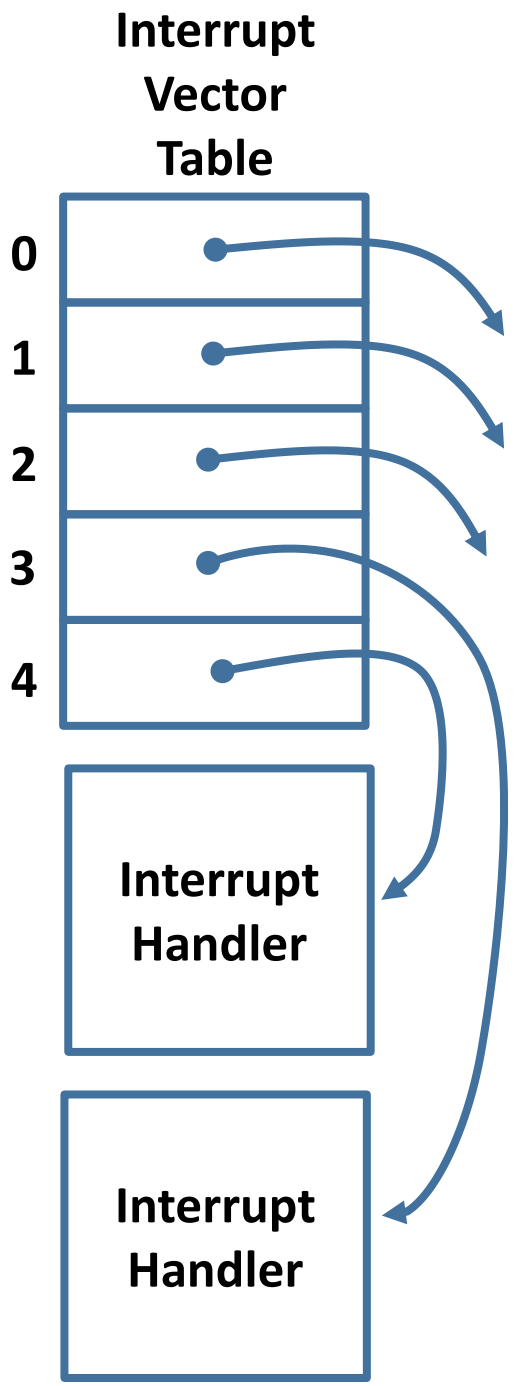
פסיקות חיצוניות אינן תלויות בריצת התכנית ומגיעות למעבד בקוו פסיקה.

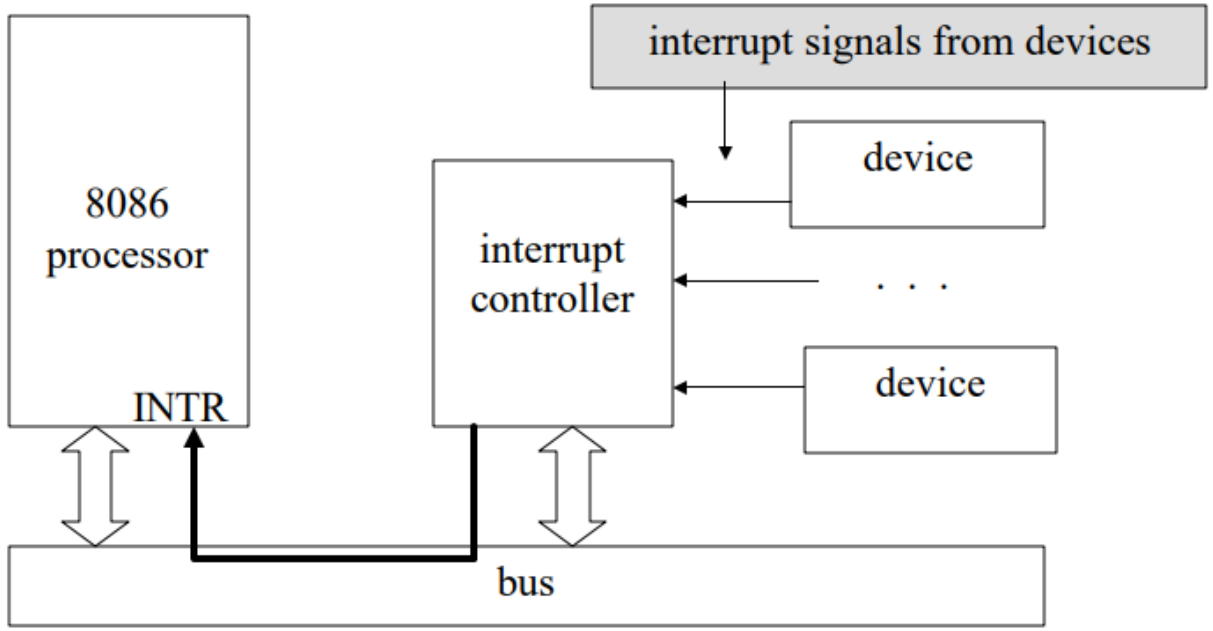
- **I/O**
  - נגרם על ידי התקן קלט/פלט.
  - דיסק מודיע למעבד על סיום פעולה.
  - מקלדת מודיע למעבד שמקש נלחץ.
  - כרטיס רשת מודיע שהגיע מידע.

- **Timer**
  - נגרם על ידי שעון המחשב.
  - מאפשר למערכת ההפעלה להפסיק ביצוע תכנית ולעבור לתכנית אחרת.



פסיקות חיצוניות

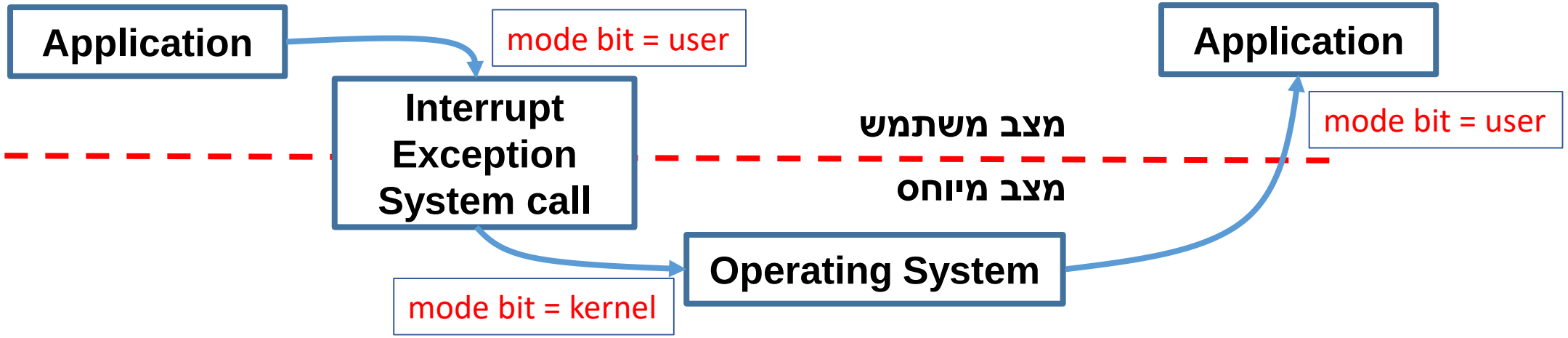




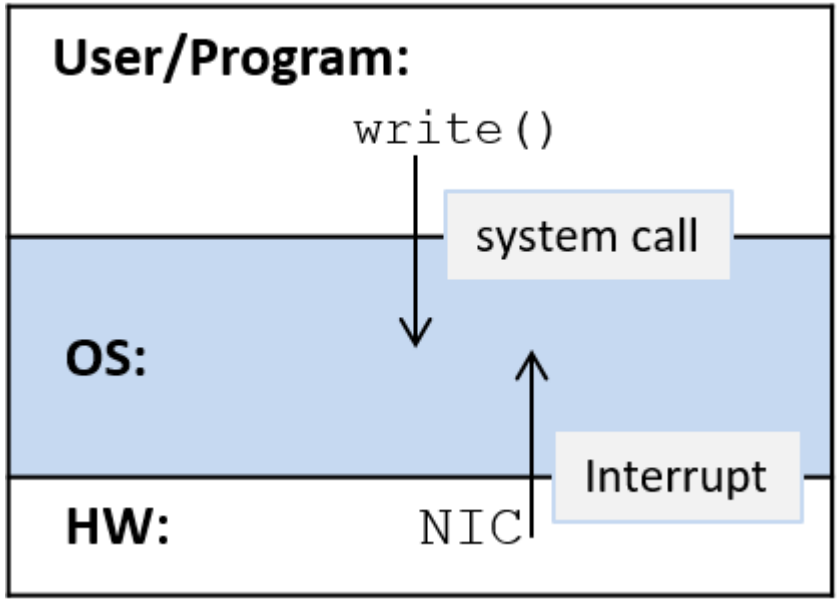
- רכיב חומרה שנקרא מנהל הפסיקות (Interrupt Controller), מאפשר לחבר כמה התקנים לקו פסיקה אחד במעבד.
- לפי הקו ממנו הגיע הפסיקה, מנהל הפסיקות שולח למעבד מספר על פס הנתונים, ושולח פסיקה לקו הפסיקה של המעבד.
- המספר שהתקבל משמש את המעבד כאינדקס לטבלת קפיצות (Interrupt Vector Table) שמערכת ההפעלה הכינה בזמן האתחול.
- לפי הכתובת שבטבלה, המעבד עובר לביצוע קוד שמטפל בפסיקה.

שני מצבי חומרה, מצב משתמש ומצב קרנל (מצב מיוחס)

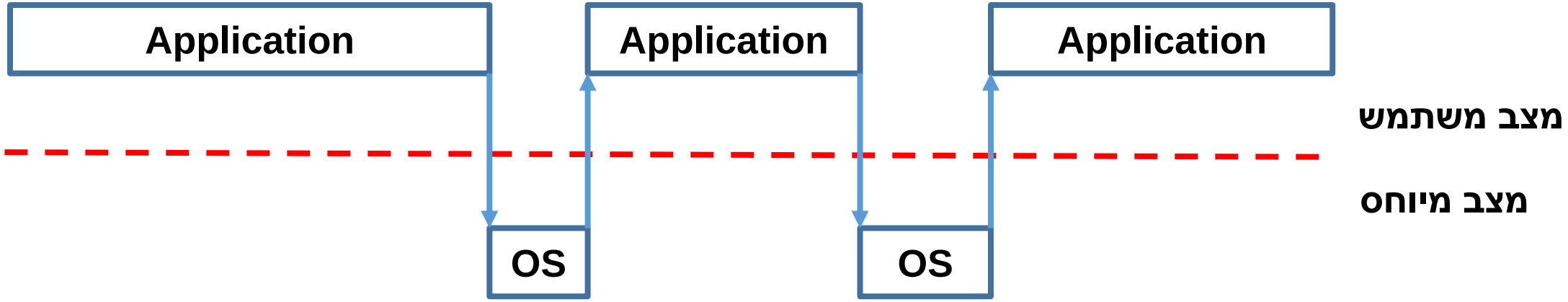
- למעבד יש שני מצבי פעולה, מצב משתמש ומצב קרנל (מיוחס).
- ישנו ביט במעבד שמשתנה בהתאם למצב המעבד.
- כשהמעבד מבצע קוד של מערכת ההפעלה הוא צריך להיות במצב קרנל.
  - במצב קרנל אפשר לבצע פעולות קלט פלט ולנהל את הגישה לזיכרון.
- כשהמעבד מבצע תכנית רגילה הוא במצב משתמש.
  - במצב משתמש אי אפשר לבצע פעולות אלו וצריך לבקש ממערכת ההפעלה.
  - אם משתמש רגיל יוכל לבצע פעולות אלו, הוא יוכל לשבש את המחשב או לגשת למידע של משתמשים אחרים.
- המנגנון שמעביר למצב קרנל הוא פסיקה, פקודת החזרה מִפסיקה (iret) מחזירה למצב משתמש.



מערכת ההפעלה מופעלת על ידי פסיקות (interrupt-driven)



- לאחר שמערכת ההפעלה איתחלה את מערכת המחשב היא מריצה תכניות, ממתינה לפסיקות ואינה רצה.
- כשמגיעה פסיקה התכנית הרצה מופסקת, המעבד עובר לבצע קוד במערכת ההפעלה, ולאחר מכן המעבד חוזר לתכנית רצה.



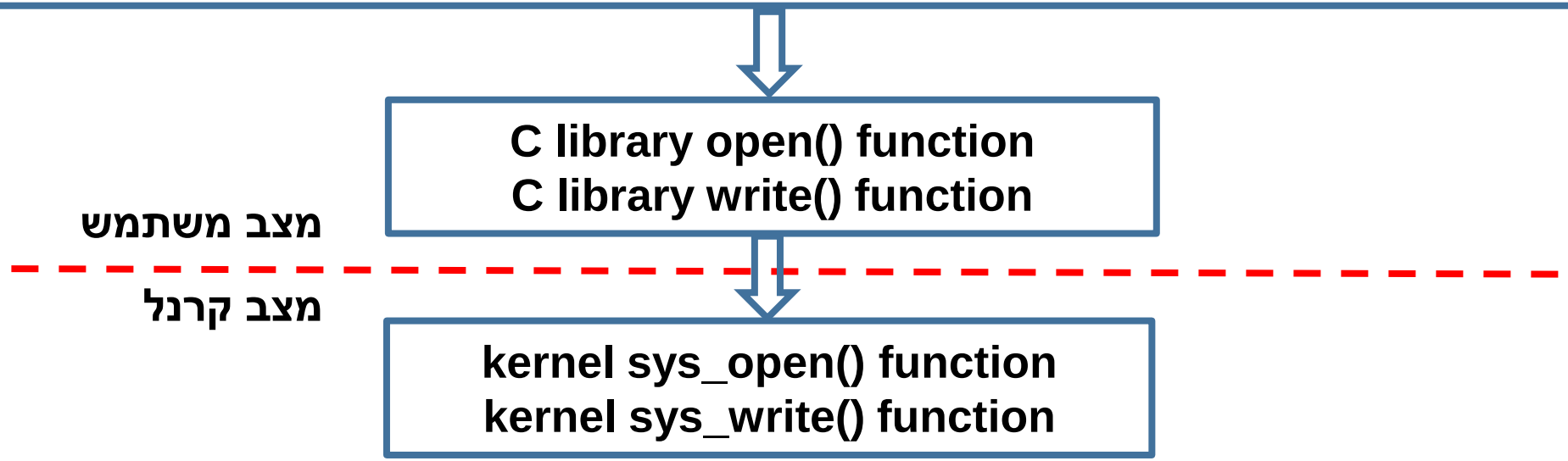
מדוע יש צורך בפקודת קריאה (syscall) למערכת ההפעלה?

- פקודות מחשב שעלולות לפגוע במשתמשים אחרים או במערכת המחשב, יכולות להתבצע רק במצב מיוחס.
- תכניות משתמש פועלת במצב משתמש ולא יכולות לבצע פקודות מיוחסות.
  - אם תכנית משתמש צריכה שרות שדורש לבצע פקודה מיוחסת, לדוגמה לקרוא קובץ, עליה לפנות למערכת ההפעלה.
- הפניה למערכת ההפעלה צריכה להעביר את המעבד למצב מיוחס ולכן לא יכולה להתבצע באמצעות קריאה לפונקציית ספריה.
- היא מתבצעת על ידי פקודת המעבד syscall, פקודה זו מעבירה את הביצוע למערכת ההפעלה **וגם** מעבירה את המעבד למצב מיוחס.
  - כעת מערכת ההפעלה יכולה לבצע בצורה מבוקרת את בקשת תכנית המשתמש.
- בסיום הקריאה, מערכת ההפעלה מבצעת פקודת מעבד (iret) שמחזירה מ-syscall, פקודה זו מחזירה את הביצוע לתכנית המשתמש **וגם** מחזירה את המעבד למצב משתמש.

קריאת מערכת: open(), write()

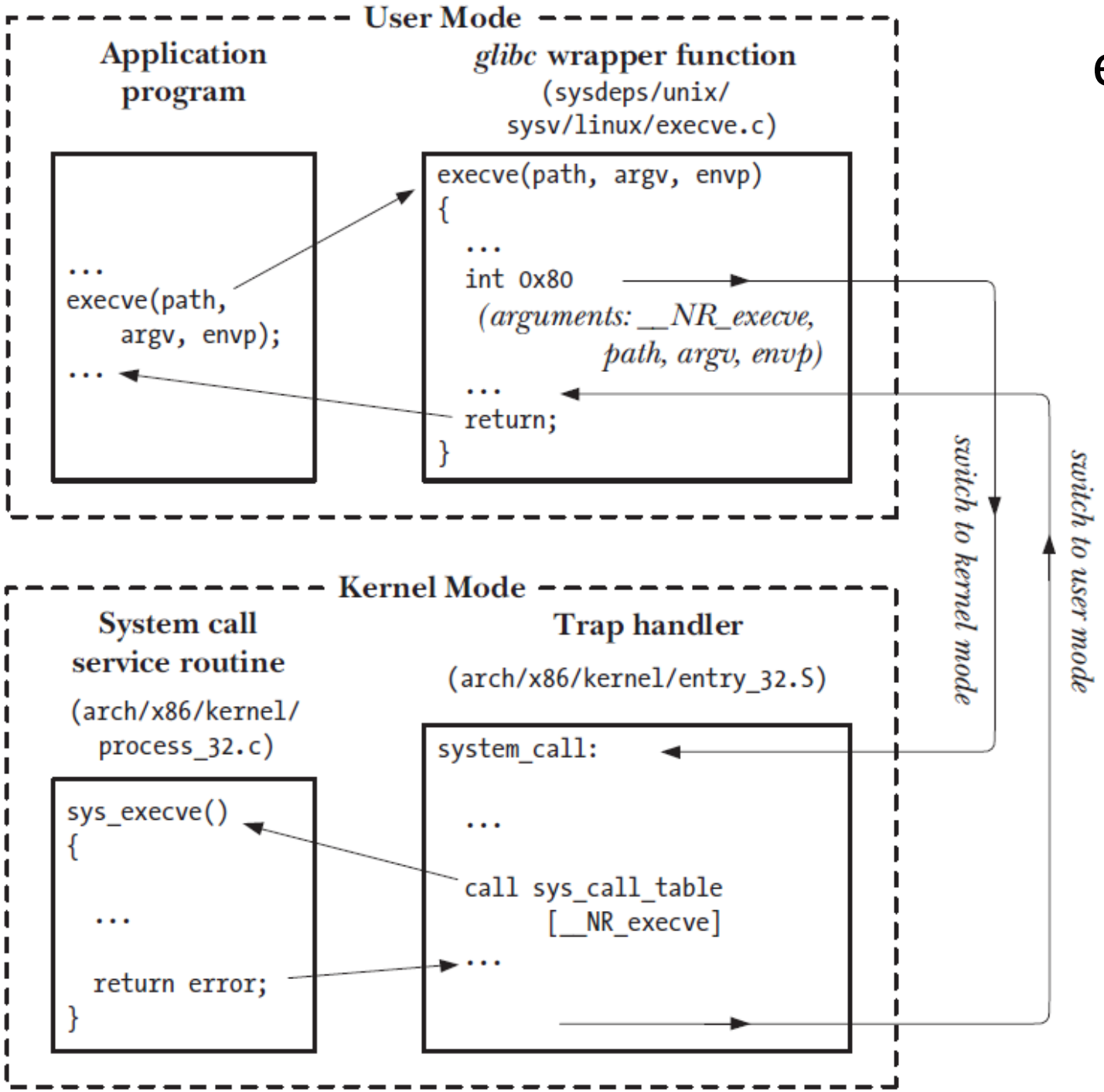
- כדי לבצע system call צריך להשתמש בפקודות אסמבלי, כדי להקל, הספרייה מספקת פונקציות שעוטפות את הקריאות למערכת ההפעלה.

```
main()
{
    int file_desc = open("myfile.txt", O_WRONLY);
    write(file_desc, "Hello World\n", 12);
}
```



קריאת מערכת: execve()

- במעבד x86 קריאת מערכת מתבצעת על ידי הפקודה int 0x80
- 0x80 (128) מכיל את כתובת הפונקציה system_call שמטפלת בכל קריאות המערכת.
- הפרמטרים לקריאה מועברים באמצעות אוגרים, מספר הפסיקה מועבר באוגר eax
- הערך המוחזר מהקריאה מוחזר למשתמש באוגר eax



UNIX	Win32	Description
fork	CreateProcess	Create a new process
waitpid	WaitForSingleObject	Can wait for a process to exit
execve	(none)	CreateProcess = fork + execve
exit	ExitProcess	Terminate execution
open	CreateFile	Create a file or open an existing file
close	CloseHandle	Close a file
read	ReadFile	Read data from a file
write	WriteFile	Write data to a file
lseek	SetFilePointer	Move the file pointer
stat	GetFileAttributesEx	Get various file attributes
mkdir	CreateDirectory	Create a new directory
rmdir	RemoveDirectory	Remove an empty directory
link	(none)	Win32 does not support links
unlink	DeleteFile	Destroy an existing file
mount	(none)	Win32 does not support mount
umount	(none)	Win32 does not support mount
chdir	SetCurrentDirectory	Change the current working directory
chmod	(none)	Win32 does not support security (although NT does)
kill	(none)	Win32 does not support signals
time	GetLocalTime	Get the current time

קריאות מערכת של Unix ושל Windows