מערכות הפעלה

1

מבוא
פסיקות
קריאות מערכת

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.

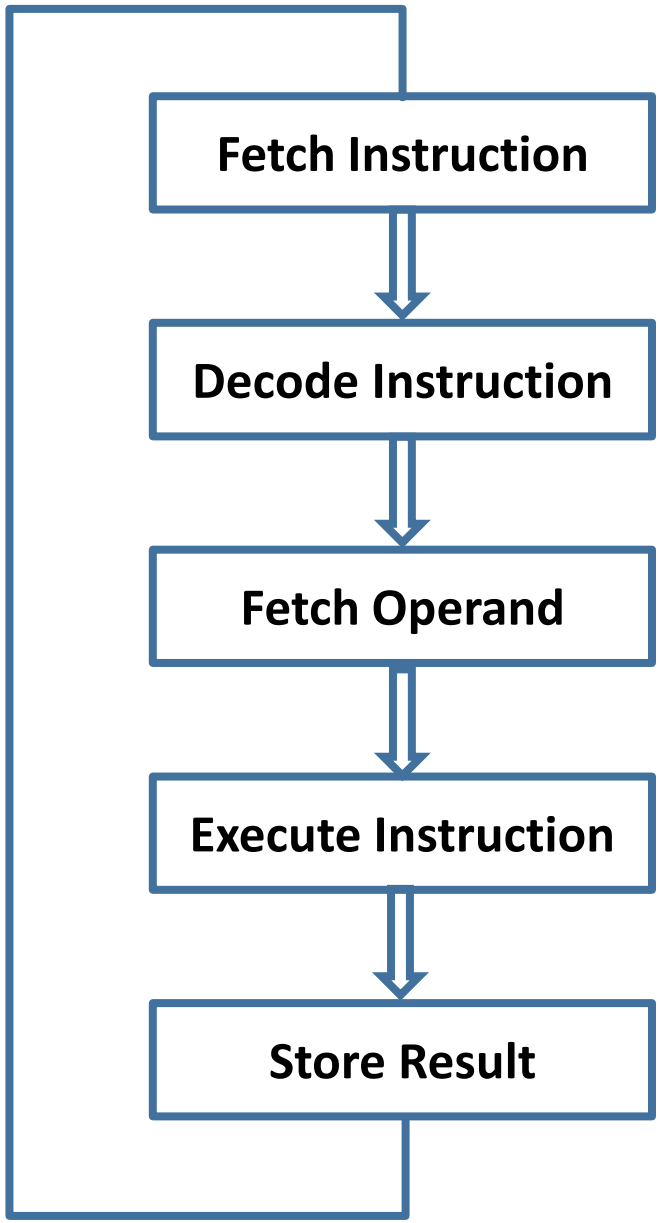
Fetch Operands

4. ביצוע הפקודה.

Execute Instruction

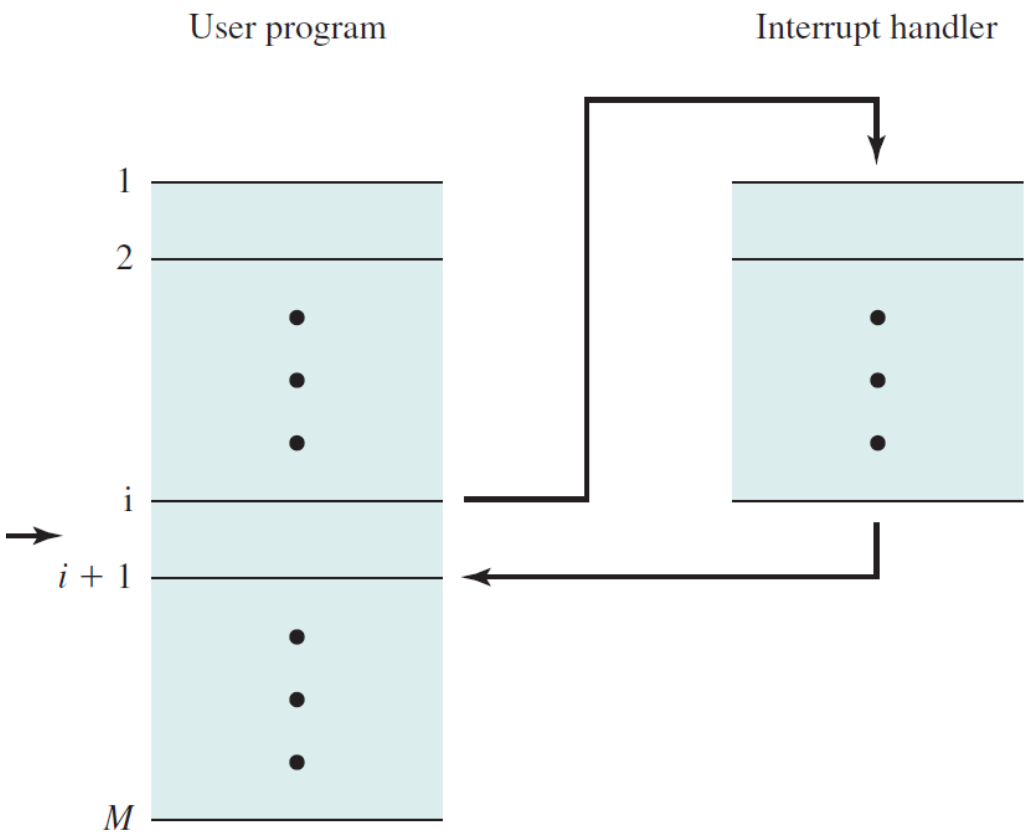
5. שמירת התוצאה.

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:

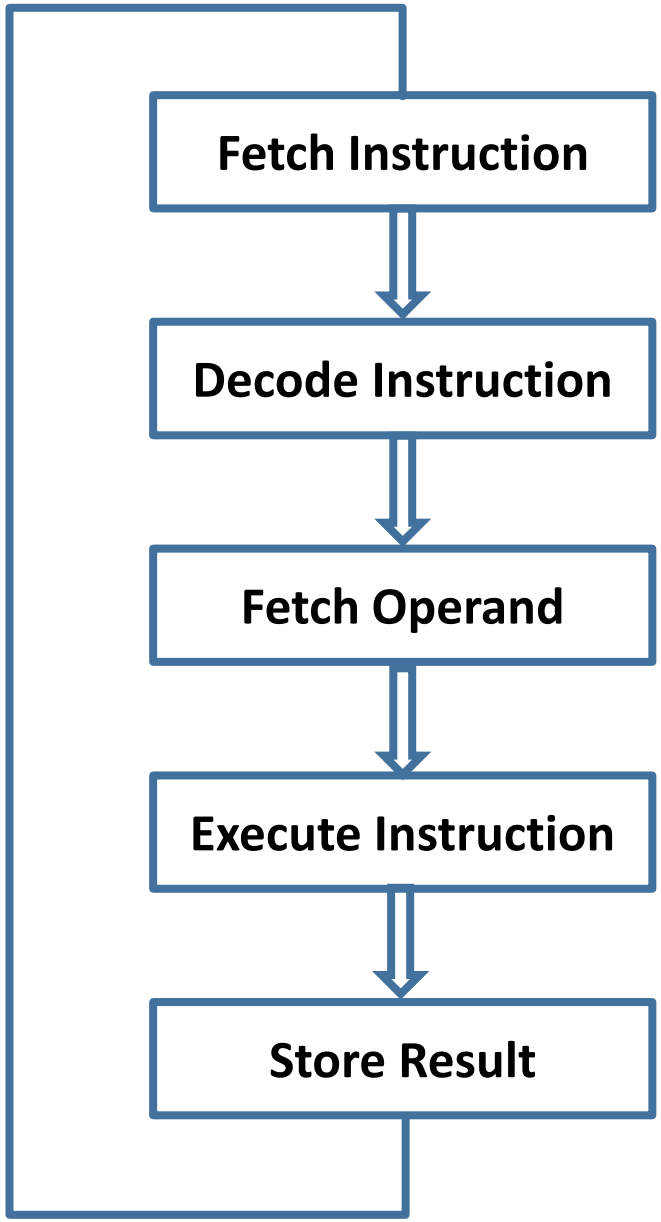
1. שמירת ה- PC, אוגר הסטטוס, אוגר המחסנית והאוגרים הכלליים של התכנית שהופסקה.
2. ביצוע הקוד המטפל בפסיקה.
3. החזרת ערך ה- PC, הסטטוס, המחסנית והאוגרים, וחזרה להמשך הריצה של התכנית שהופסקה.

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

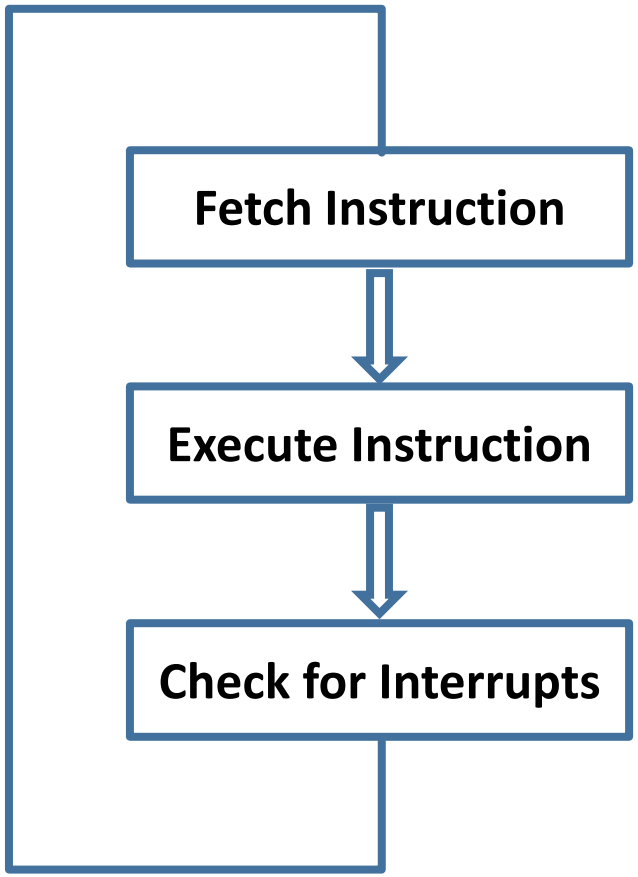


2. פסיקות קריאה למערכת ההפעלה (system call)

- ישנה פקודת מכונה שהביצוע שלה גורם לפסיקה (ומעבר למערכת ההפעלה), 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.

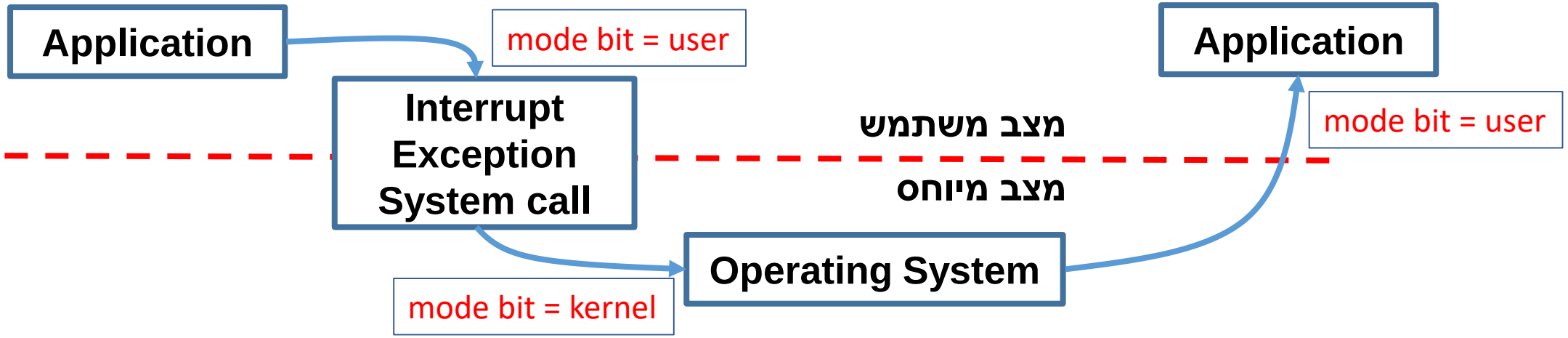


- **I/O**
  - נגרם על ידי התקן קלט/פלט.
  - דיסק מודיע למעבד על סיום פעולה.
  - מקלדת מודיע למעבד שמקש נלחץ.
  - כרטיס רשת מודיע שהגיע מידע.

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

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.



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);
}
```