

# מערכות הפעלה

סיון טולדו

**פרק 1**

**מבוא**

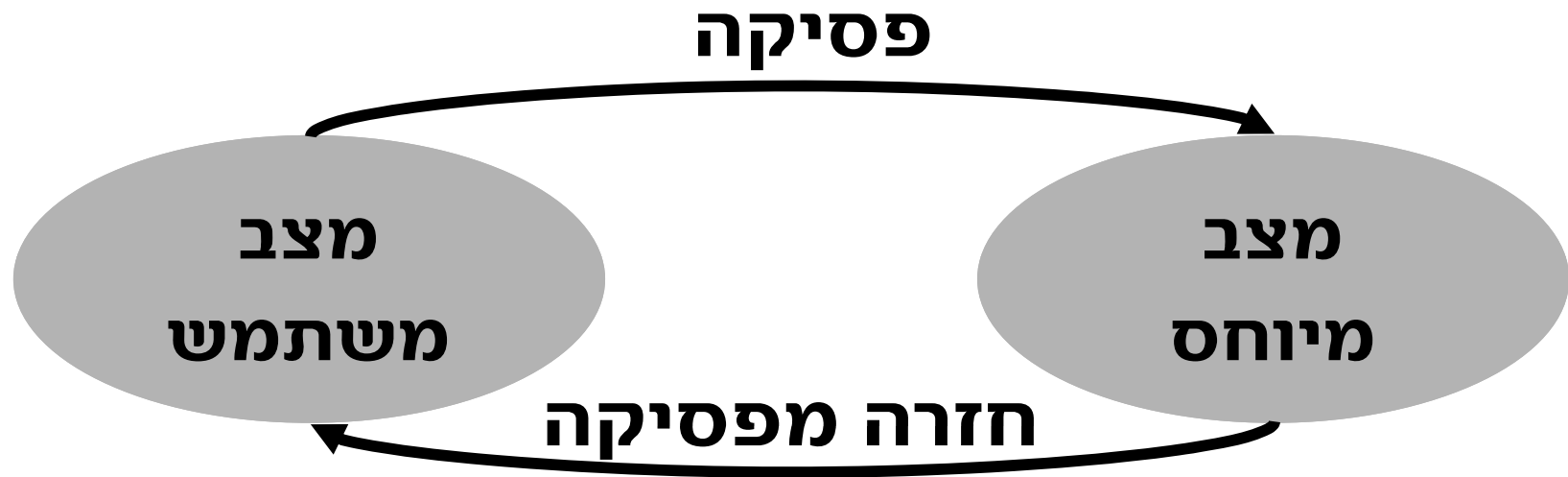
# מערכת ההפעלה

- ❖ מגינה על חומרה על מנת לתחם תקלות ועל מנת להגן על מידע
- ❖ מנהלת את משאבי החומרה (זיכרון, זמן מעבד, דיסקים, רשת תקשורת) ביעילות והגינות
- ❖ מספקת לתוכניות ממשקים אחידים ונוחים לשימוש בהתקני חומרה

# חשיבות ההגנה על חומרה

- ❖ תוכנית אחת לא יכולה לקרוא או לכתוב מהזיכרון של אחרת
- ❖ תקלה בתוכנית אחת לא משפיעה על אחרות
- ❖ בפרט, גם אם תוכנית נתקעת בלולאה אין-סופית תוכניות אחרות ממשיכות לרוץ

# הגנה על חומרה: המעבד



פקודות מסויימות אסורות  
גישה לאוגרים מסויימים בלבד  
איזורים מסויימים בזיכרון בלבד

כל פקודות המכונה מותרות  
גישה לכל האוגרים אפשרית  
גישה לכל הזיכרון אפשרית

# הגנה על חומרה: המעבד



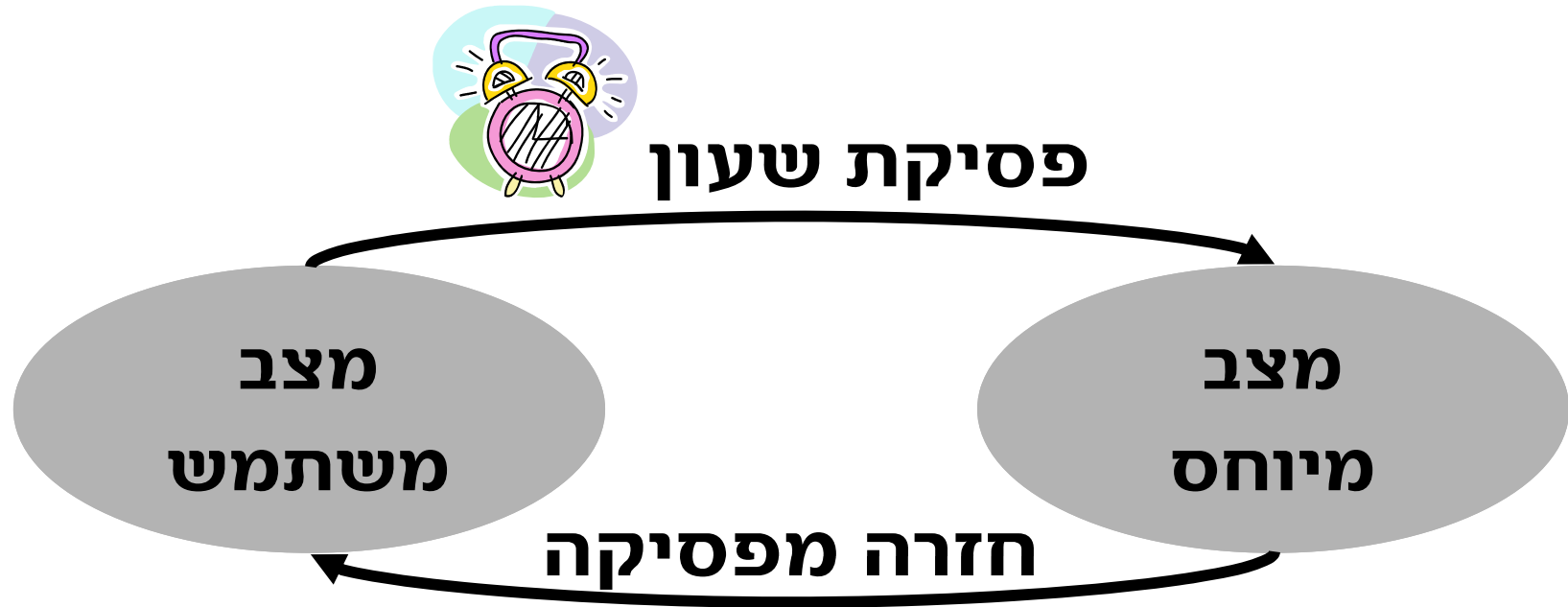
פסיקת שעות



שגרה של מערכת ההפעלה

כתובת השגרה שמורה באוגר מוגן

# הגנה על חומרה: המעבד



שגרה של מערכת ההפעלה

כתובת השגרה שמורה באוגר מוגן

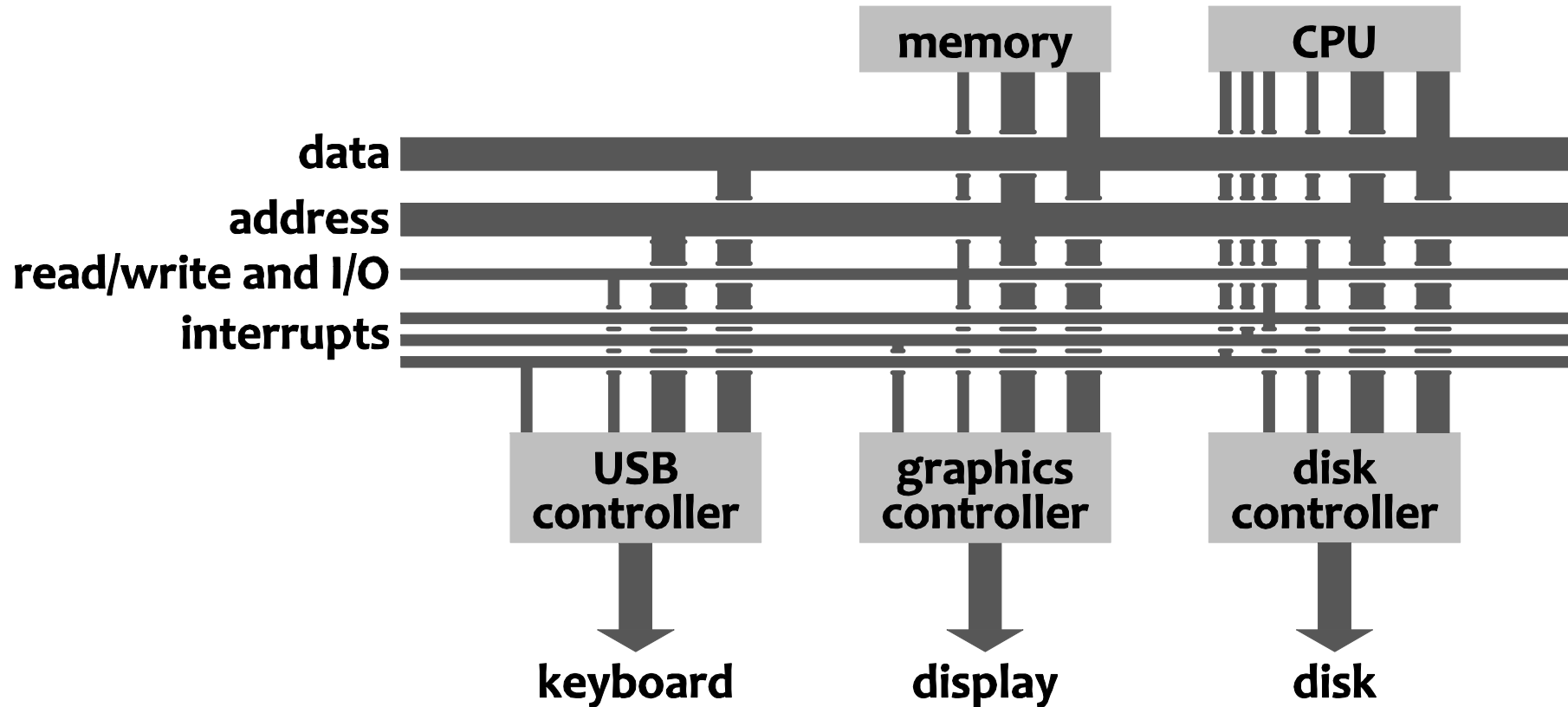
מערכת ההפעלה רצה מדי פעם, לא "ברקע"

# הגנה על זיכרון

- ❖ שיתוף פעולה בין מערכת ההפעלה והמעבד
- ❖ כל גישה לזיכרון נבדקת על ידי המעבד מול טבלאות שמערכת ההפעלה מתחזקת
- ❖ פירוט בפרק 2



# שליטה בהתקנים חיצוניים



❖ התקנים חיצוניים מחוברים לבקרים שמחוברים לפס התקשורת המרכזי של המחשב (לעיתים יש למחשב מספר פסים)

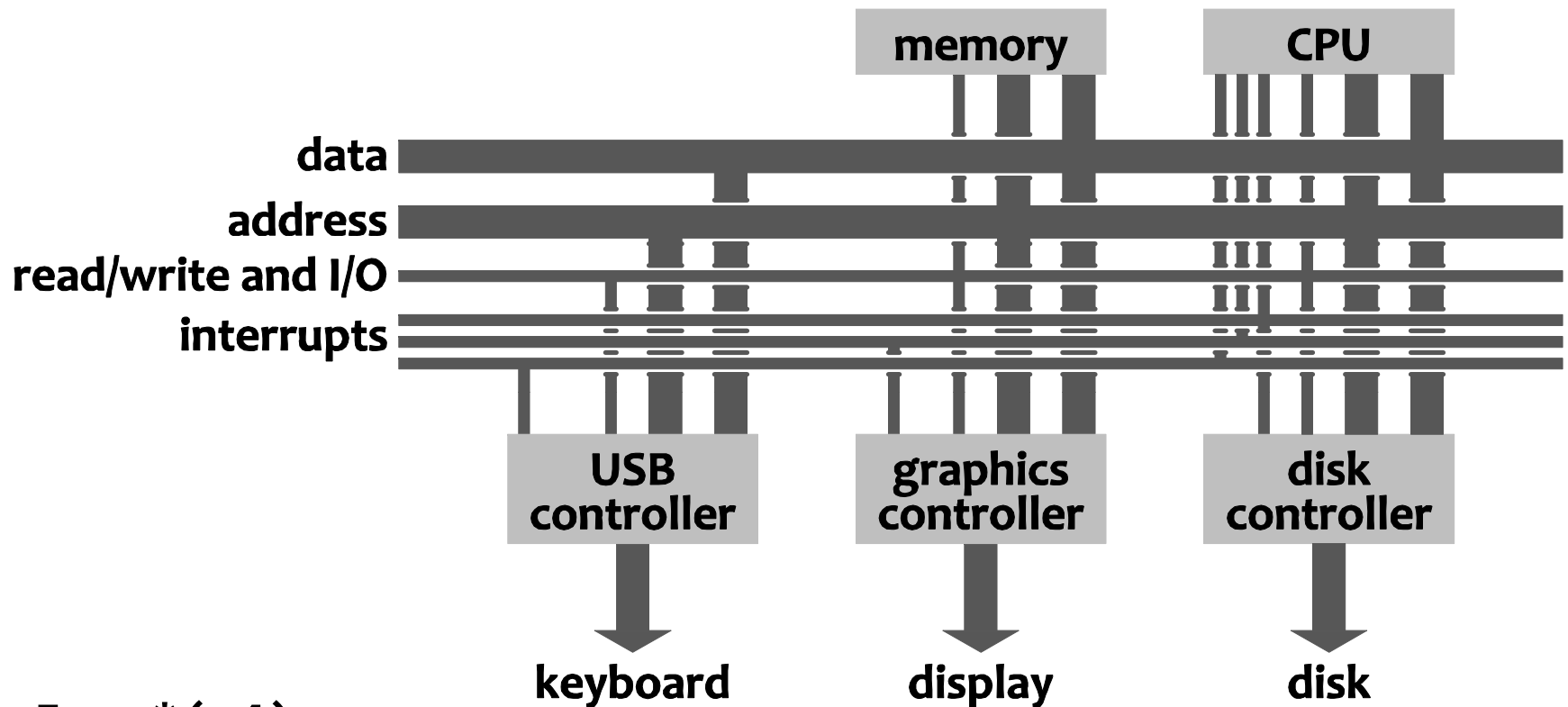
# תקשורת עם בקרים

❖ מידע מועבר בין המעבד ובקרים בעזרת פקודות קריאה/כתיבה מיוחדות שמדליקות את סיבית הקלט/פלט, או בעזרת פקודות קריאה/כתיבה מזיכרון

❖ כל בקר מגיב לתחום כתובות מסוים

```
load    r5 = *(r4)
store   *(r4) = r6
add     r3 = r3 + r8
bc      if (r3) goto loop
ioread  r5 = *(r4)
iowrite *(r4) = r6
```

# תקשורת על הפס



```
load    r5 = *(r4)
store   *(r4) = r6
add     r3 = r3 + r8
bc      if (r3) goto loop
ioread  r5 = *(r4)
iowrite *(r4) = r6
```

# הגנה על התקנים חיצוניים

- ❖ פקודות קלט/פלט שמדליקות את סיבית הקלט/פלט בפס מותרות רק במצב מיוחס
- ❖ בקרים שמגיבים לפקודות קריאה/כתיבה מזיכרון מוגנים בעזרת מנגנון ההגנה על זיכרון
- ❖ על מנת לגשת להתקן חיצוני, תוכניות קוראות לשגרה של מערכת ההפעלה
- ❖ הקריאה מתבצעת בנוהל מיוחד שמעביר את המעבד למצב מיוחס, על מנת שמערכת ההפעלה תוכל לתקשר עם הבקר

# קריאות מערכת

- ❖ נוהל קריאה לשגרות של מערכת ההפעלה
- ❖ משתמש במנגנון הפסיקות או במנגנון דומה
- ❖ מעביר את המעבד למצב מיוחס; חזרה מקריאת מערכת מחזירה את המעבד למצב משתמש
- ❖ ציון השגרה שרוצים לקרוא לה על ידי מזהה מספרי ולא על ידי כתובת (למשל, `read==1`, `write==1`, וכדומה)
- ❖ אין שימוש בכתובת על מנת למנוע מעבר לשגרה שאינה של מערכת ההפעלה במצב מיוחס

# ממשקים אחידים לגישה לחומרה

- ❖ מגוון אדיר של התקני חומרה דומים אך לא זהים
- ❖ תוכניות זקוקות לממשק אחיד לגישה לחומרה על מנת שיוכלו לרוץ ללא שינוי על מחשבים עם התקנים שונים
- ❖ פתרון: ספריית שגרות בעלות ממשק אחיד שכל אחת מהן ניגשת לסוג אחר של התקן חומרה
- ❖ הספרייה הזו היא חלק ממערכת ההפעלה, בעיקר משיקולי ביצועים
- ❖ הממשק לתוכניות בעיקר על ידי קריאות מערכת

# ממשקים משופרים לגישה לחומרה

- ❖ הממשק שהשגרות הללו מספקות יכול לספק שירותים שהחומרה אינה מסוגלת לספק בעצמה
- ❖ למשל, אחסון או העברת נתונים אמינה בעזרת חומרה לא אמינה
- ❖ יכולות כאלה כלולות לפעמים במערכות הפעלה משיקולי ביצועים או שווק
- ❖ אבל לפעמים הן ממומשות על מחשב ייעודי שהוא חלק מהתקן החומרה (מערך דיסקים, דיסק מבוסס זיכרון הבזק, כרטיס רשת "חכם", וכו')

# ניהול יעיל והוגן של חומרה

- ❖ יעילות: שאיפה למרב את הנצילות של משאבי חומרה לטובת תוכניות רגילות ומשתמשים; מזעור כמות המשאבים שמערכת ההפעלה עצמה צורכת או שמתבזבזים סתם בגלל ניהול כושל
- ❖ הגינות: חלוקה הוגנת של המשאבים בין משתמשים; חלוקה הוגנת של משאבים בין תוכניות שרצות בו-זמנית





Ken Thompson &  
Denis Ritchie

# מערכות Unix

- ❖ מגוון גדול של מערכות הפעלה תואמות זו לזו ברמת קוד המקור (לא ברמת קוד המכונה), שמוצא כולן ממערכות הפעלה בשם unix שפותחה במעבדות המחקר של חברת Bell
- ❖ AIX של IBM, Solaris של Sun, HP-UX של HP, True64 של SGI, IRIX של SGI, Compaq
- ❖ מערכת ההפעלה של המקינטוש (החל מגרסה 10) היא מערכת יוניקס עם ממשק ייחודי
- ❖ ממחשבים אישיים ותחנות עבודה עד שרתי 7/24 עם מאות מעבדים

Linus Torvalds

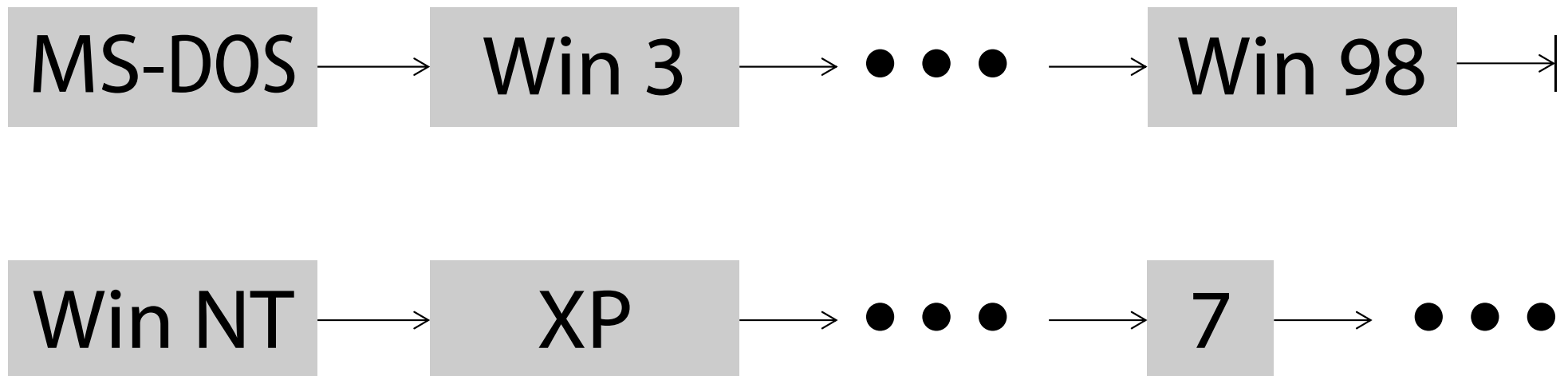


# מערכות Linux

- ❖ מערכת הפעלה תואמת-יוניקס שפיתחה החל ב-1991 על ידי סטודנט פיני
- ❖ מחולקת חינם עם קוד המקור שלה
- ❖ חבילות הפצה (distributions) כוללות את מערכת ההפעלה, תוכנת התקנה, ותוכנות שימושיות
- ❖ מטלפונים (Android) ועד שרתים, כולל כמעט כל מחשב שיכול להריץ חלונות

# מערכות Windows

- ❖ מפותחות ומשווקות על ידי חברת מיקרוסופט.
- ❖ גרסאות למחשבי כף יד/טלפונים, למחשבים אישיים (XP, 7) ולשרתים קטנים ובינוניים (גרסאות מותאמות)
- ❖ תאימות גבוהה ברמת קוד המכונה בין הגרסאות השונות



# ישויות עיקריות במערכת ההפעלה

❖ תהליך: מחשב וירטואלי לתוכנית יחידה ומשתמש יחיד

❖ חוט: מעבד של מחשב וירטואלי מרובה מעבדים

❖ קובץ: התקן אחסון וירטואלי

❖ קשר: ערוץ תקשורת וירטואלי בין תהליכים, אולי במחשבים

שונים

❖ חלון: מקלדת, עכבר, ותצוגה וירטואליים. המשתמש יכול לסמן

שהוא מעוניין באינטראקציה עם חלון זה או אחר

❖ תור הדפסה: מדפסת וירטואלית

❖ משתמשים וקבוצות משתמשים לבקרת גישה למשאבים ומידע

# מערכות הפעלה אחרות

- ❖ מערכות הפעלה למחשבים מרכזיים (mainframes); בעיקר לצורך שימור מערכות מחשוב ארגוניות ישנות
- ❖ מערכות הפעלה למערכות זמן אמת; שליטה בכור גרעיני, בטייס אוטומטי, או בטיל דורשת יכולות תזמון מדויקות שאין למערכות הפעלה רגילות; אם אפשר לסבול כשלים נדירים ניתן להשתמש בגרסאות מתאימות של חלונות או לינוקס
- ❖ מערכות הפעלה להרצת מערכות הפעלה אחרות: virtual machines; נדון בכך בהמשך הקורס

# סיכום

- ❖ מערכות ההפעלה מגינה על החומרה על מנת לתחם תקלות ולהגן על מידע
- ❖ מערכת ההפעלה מנהלת את החומרה ביעילות והגינות
- ❖ מערכת ההפעלה מספקת ממשקים אחידים ונוחים לגישה לחומרה
- ❖ רוב מערכות ההפעלה כיום שייכות לשתי משפחות עיקריות (לינוקס/יוניקס וחלונות), לכולן יכולות בסיסיות דומות, וכולן משתמשות באותם מנגנונים
- ❖ ההבדלים העיקריים כיום בין מערכות שונות מתבטאים בסגנון הממשקים וביכולות בקצוות: שרתים גדולים במיוחד ומחשבים קטנים במיוחד