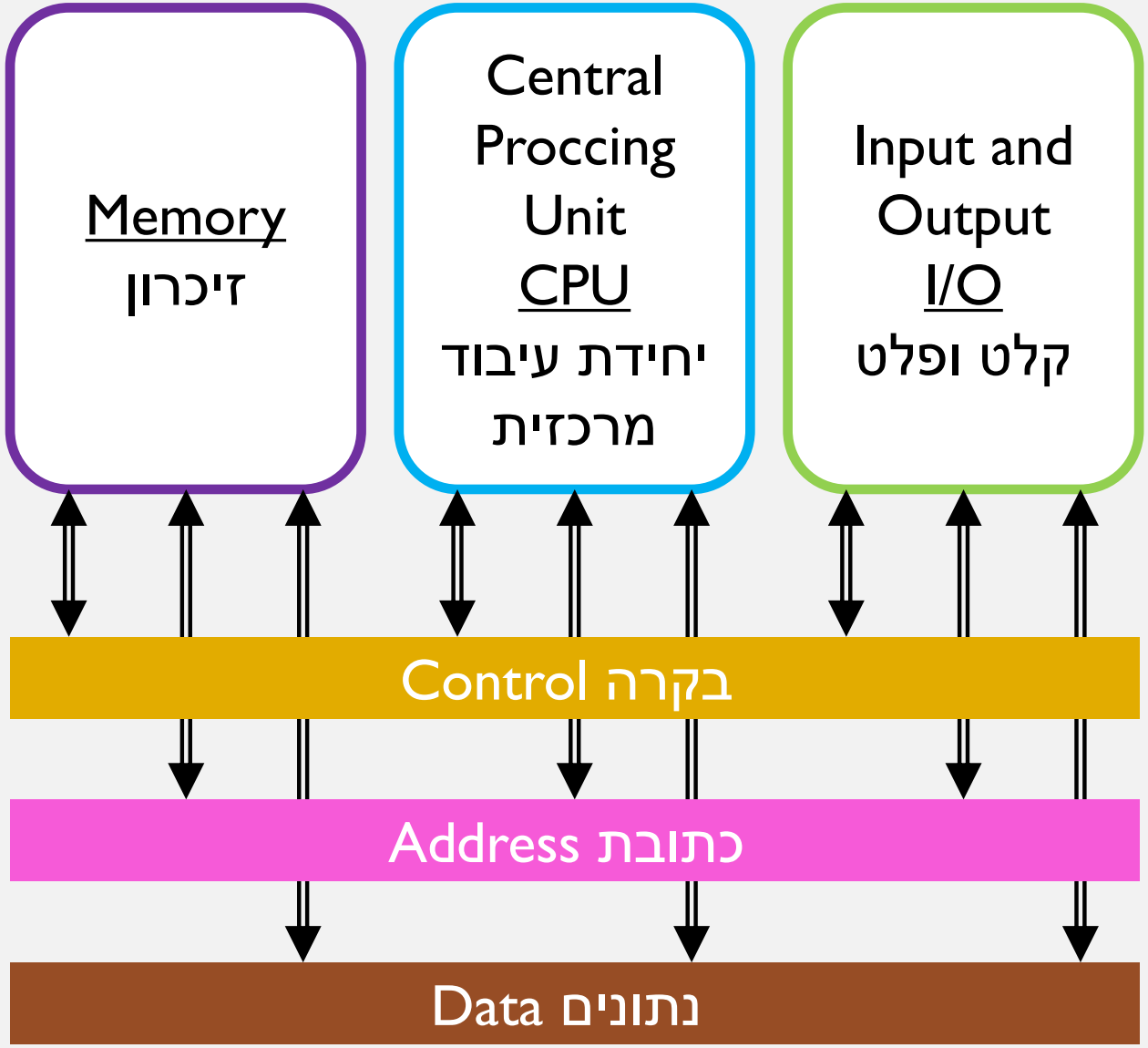
מערכות הפעלה

מבוא

VNA - תזכורת



- ארכיטקטורת פון-ניומן (Von-Neumann Architecture), או בקיצור VNA, הוא מודל ארכיטקטורה
- המודל מורכב מ:
  - שלוש יחידות (unit)
  - שלושה פסים (bus)
- בין כל יחידה וכל פס יש תקשורת דו-כיוונית
- שימו לב: אין תקשורת ישירה בין היחידות
- שני הרכיבים העיקריים של הCPU הם:
  - הALU (Arithmetic Logic Unit) האחראית על ביצוע חישובים
  - הCU (Control Unit) האחראית על הרצת תוכניות

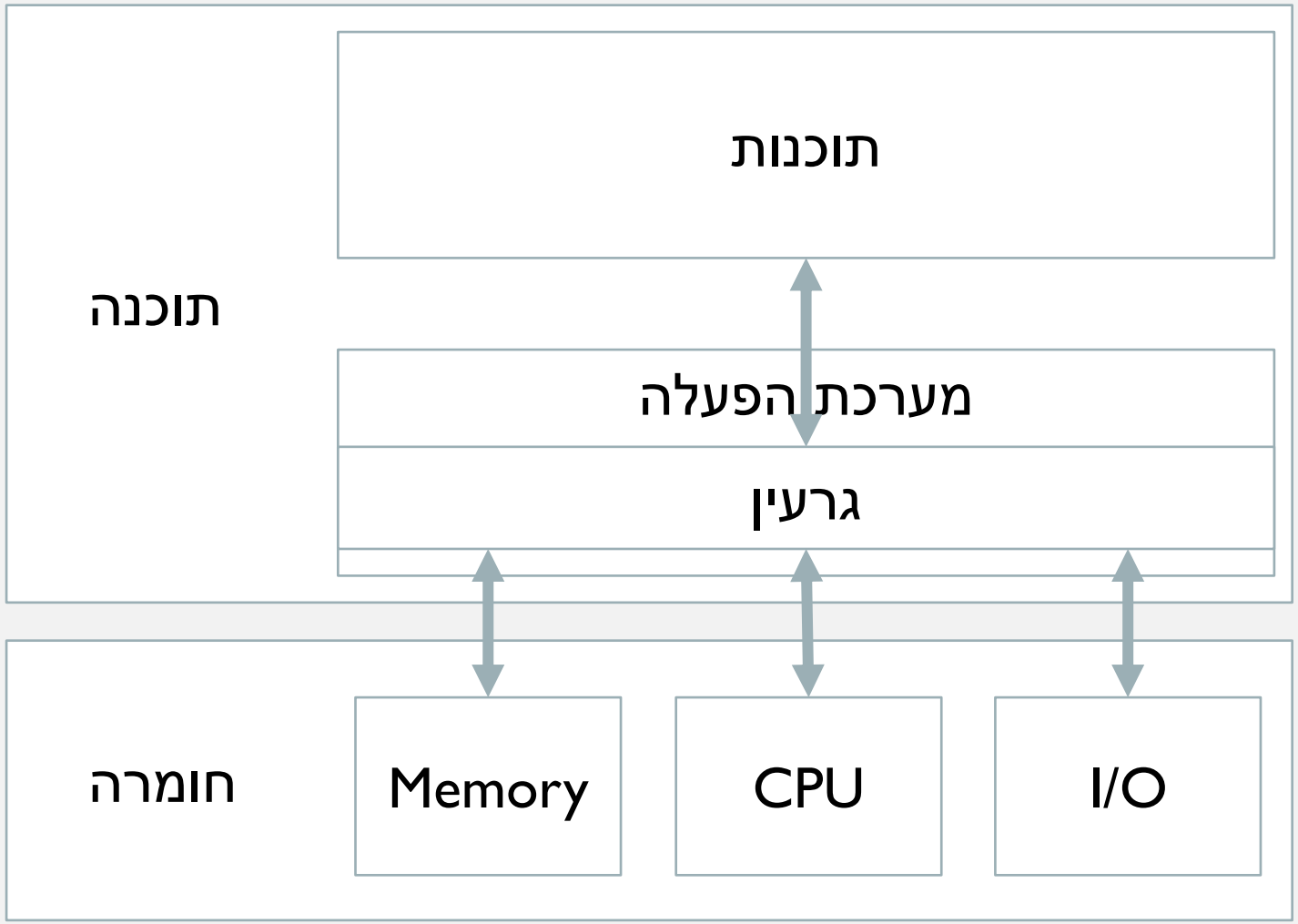
תהליך תוכנה

- תוכנית היא סדרה של פקודות
- כדי להריץ תוכנית היא נטענת לזיכרון
- הפקודות שלה נקראות ומופעלות אחת אחרי השנייה
- זה נקרא תהליך (process)
- מודל זה מאוד מוגבל:
  - הוא מאפשר הרצה של תוכנית אחת בלבד – תוכניות לא יכולות לרוץ במקביל
  - כל מחשב יכול להכיל תוכנית אחת בדיוק, אם נרצה להריץ תוכנית אחרת, נצטרך להחליף את התוכנית השמורה בזיכרון
  - במקרה הטוב, לטעון אותה מממקום אחר במחשב, במקרה הרע, לכתוב אותה למחשב
- מודל זה מתאים למחשבים יעודיים, אשר כל מטרתם היא הרצה של תוכנית מסוימת
  - לדוגמא: מחשבון דיגיטלי

מחשבים כללים

- רוב המחשבים שאנו פוגשים בחיי היום יום הם מחשבים כללים:
  - כגון: מחשב אישי, טלפון, טאבלט, קונסולת משחקים, טלוויזיה חכמה, סטרימר
- הם מאחסנים מגוון רחב של תוכנות שניתן להריץ, וניתן להריץ מספר תהליכים במקביל – כלומר ניתן להריץ מספר תוכנות במקביל ו/או את אותה תוכנה מספר פעמים במקביל
- לדוגמא, במחשב האישי, אני יכול במקביל:
  - לערוך מספר קבצי טקסט, כולם בעזרת אותה תוכנת עריכה
  - להשתמש בדפדן אינטרנט כדי לגלוש באתרים עם מידע רלוונטי
  - להשתמש בתוכנת מדיה כדי לשמוע מוזיקה בזמן שאני מבצע את העריכות
  - לסנכרן תיקיות במחשב עם גיבוי השמור בשרת אינטרנט
- מערכת הפעלה היא תוכנה המאפשרת לנהל את רכיבי המחשב, ולהפעיל מספר תהליכים במקביל

מחשבים כללים



- המחשב מורכב משני חלקים:
- חומרה (Hardware)
- תוכנה (Software), אשר מורכבת מ:
  - מערכת הפעלה (Operating System)
  - תוכנות/אפליקציות (Programs/Applications)
- מערכת ההפעלה מהווה את הגורם המקשר בין התוכנות לחומרה
- הגרעין (kernel) הוא הרכיב העיקרי של מערכת ההפעלה
- הוא אחראי על התקשורת בין הרכיבים השונים

מערכת הפעלה

- למערכת הפעלה 3 תפקידים עיקריים:

1. הקצאת משאבים
2. תזמון
3. תשתית

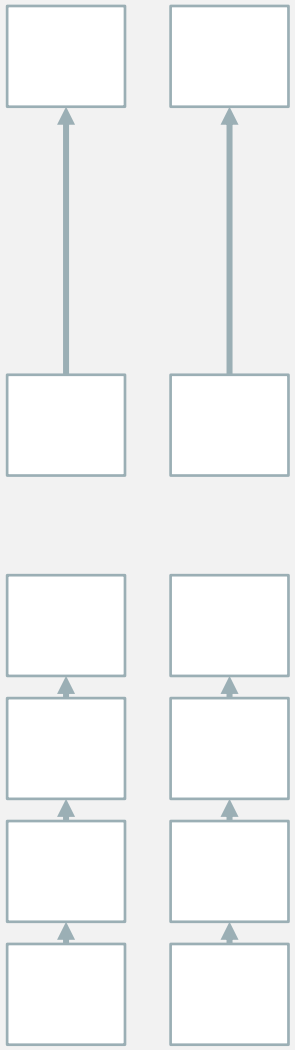
הקצאת משאבים

- כל תהליך צורך משאבים כגון זיכרון וערוצי תקשורת
- אם לכל התהליכים יש גישה בלתי מוגבלת למשאבים, יכולות להיווצר בעיות
- למשל עם שני תהליכים מקצים משתנה בגודל byte, ושניהם יקצו אותו לאותו מקום בזכרון, כאשר תהליך אחד ישנה את ערך המשתנה, התהליך השני יושפע מזה
- מערכת ההפעלה מקצה לכל תהליך שרץ חלק ממשאבי המחשב
- למשל כל ריצה של תוכנה (כולל שתי ריצות של אותה תוכנה) מקבלות איזור אחר מהזיכרון בו הן משתמשות

תזמון

- לעיתים תהליכים שונים צריכים לעשות שימוש באותו משאב, וזה לא משאב שניתן לחלק
- לדוגמא, שני תהליכים המבצעים חישובים צריכים להשתמש בALU
- את הALU לא ניתן לחלק
- מערכת ההפעלה מבצעת תזמון, לכל תהליך היא מקצה זמן שבו הוא, ורק הוא, יכול להשתמש ברכיב
- באופן זה ניתן לבצע תהליכים מקבילים

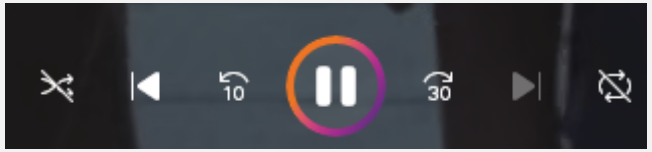
ריצה מקבילית



- כפי שתיארנו במחשב אנו מריצים תהליכים רבים במקביל
- אמנם מחשבים מודרנים רבים מכילים מספר רכיבים היכולים לבצע את אותה עבודה (מרובי ליבות), אך לרוב כמות התהליכים גדולה הרבה יותר מכמות הליבות
- נניח שאנו רוצים להריץ שני תהליכים (עם ליבה אחת). כיצד ניתן לבצע זאת?
- פתרון פשוט הוא להריץ את התהליך הראשון, כשהוא מסיים להריץ את השני
- בפתרון זה לא באמת נקבל תהליך מקבילי
- במקום זאת, מערכת ההפעלה עושה זאת בצעדים
  - התהליך הראשון מתקדם קצת ואז התהליך השני מתקדם קצת
  - התהליך הראשון מתקדם עוד קצת ואז התהליך השני מתקדם עוד קצת
  - וכך הלאה
- בעצם שני התהליכים לא באמת רצים במקביל, אלא מתחלפים בינהם
- המהירות שבה זה מתבצע כלך כך גדולה שלנו זה "מרגיש" כאילו הם רצים במקביל

תשתית

- מערכת ההפעלה מספרת תשתית שבאמצעותה כל רכיבי המחשב השונים מתקשרים:
  - החומרה
  - התוכנות
  - המשתמש
- היא מבצעת זאת בעזרת ממשקים (interface)
- ממשק הוא כלי באמצעותו רכיב אחד יכול לבצע פעולות של רכיב אחר או לקבל מידע עליו
- דוגמאות:
- שלט טלוויזיה הוא בעצם ממשק משתמש לטלוויזיה – הוא מאפשר למשתמש להפעיל פעולות שונות של הטלוויזיה
- ממשק תוכנה המאפשר למשתמש לשלוט בניגון מדיה



- מערכת ההפעלה מגדירה מספר ממשקים

ממשק משתמש

- ממשק משתמש מספק למשתמש את היכולת לבצע פעולות במחשב, להעביר אליו נתונים ולקבל מידע עליו
- עושה שימוש בעזרי פלט (בעיקר מסך/ים, אך גם עזרים נוספים כגון רמקול ומדפסת) כדי לספק למשתמש מידע
- עושה שימוש בעזרי קלט (בעיקר מקלדת ועכבר, אך גם עזרים נוספים, כגון מיקרופן ומצלמה) כדי לקבל בקשות ונתונים מהמשתמש

ממשק משתמש

- ניתן לחלק ממשקי משתמש לשתי קטגוריות עיקריות
  - ממשק פקודות (CLI – Command Line Interface) – אשר מאפשר למשתמש להזין פקודות באמצעות המקלדת, ולראות נתונים ע"י טקסט
  - ממשק גרפי (GUI – Graphical User Interface) – אשר מציג למשתש נתונים באופן גרפי, ומאפשר למשתמש לבצע פעולות בעזרת עזרי פלט נוספים כגון עכבר
- לממשקים גרפים יש יתרונות רבים:
  - נוטים להיות הרבה יותר נוחים ואינטואיטיביים לשימוש ע"י המשתמש
  - מאפשרים הצגה נוחה יותר של מספר נתונים במקביל
  - מאפשרים יכולות עיצוב עשירות יותר (הם "יפים" יותר)
  - רוב מערכות ההפעלה המודרניות מספרות ממשק משתמש גרפי
  - עם זאת, הן עדיין מספקות את היכולת להשתמש בממשק פקודות
    - לעיתים פעולות מסויימות עשויות להיות נוחות יתר לביצוע באופן זה
    - לעיתים יש פעולות מסויימות שניתן לבצע רק דרך ממשק זה

ממשק חומרה

- ממשק החומרה אחראי על תרגום פעולות של רכיבי חומרה לשימוש התוכנה
- לדוגמא, מה קורה שאנו מזיזים את העכבר?
  - מערכת ההפעלה מקבלת מהעכבר (רכיב חומרה) נתונים על האופן שבו הוא זז
  - ממשק החומרה מתרגם נתונים אלה כדי להבין באיזה כיוון ומרחק הוא זז
  - מערכת ההפעלה מחשבת מה צריך להיות מיקומו החדש של העכבר
  - ממשק החומרה מתחבר למסך, וגורם לו לעדכן את מה שהוא מציג:
    - במקום שבו הופיע סמן העבר קודם, הוא לא יופיע
    - במקום שבוא הוא צריך להופיע, הוא יופיע

ממשק חומרה

- ממשק החומרה פועל באמצעות רכיבים הנקראים מנהלי התקן (Driver)
- מנהל התקן הוא תוכנה אשר "מלמדת" את ממשק החומרה את "השפה" של ההתקן:
  - איך ההתקן מעביר מידע (לדוגמא: איך העכבר מתאר את התזוזה שלו)
  - איך ההתקן "מבין" מידע (לדוגמא: איך להגיד למדפסת להדפיס צבע מסויים)
- באופן עקרוני לכל התקן חומרה, יש להתקין driver ייעודי עבורו
- בעבר, רכיב חומרה היה מסופק עם עזר זיכרון (כגון דיסק) אשר הכיל התקנה לdriver עבורו, והיה צריך להתקין את הdriver לפני שהיה אפשר לעשות שימוש ברכיב
- כיום, עם סטנדרטיזציה של מנהלי התקן, והתקדמויות ביכלות אחסון ואינטרנט, אנו לא צריכים לבצע זאת בעצמנו. בחיבור התקן, לרוב אחד מהבאים מתקיים:
  - מערכת ההפעלה כבר מכילה driver שמתאים להתקן זה
  - ההתקן עצמו מכיל התקנה לdriver והוא מותקן אוטמטית כשההתקן מחובר לראשונה
  - מערכת ההפעלה מוצאת באינטרנט ומתקינה את הdriver מאחורי הקלעים

ממשק תוכנה

- ממשק תוכנה (API – Application Programming Interface) מאפשר למתכנתים לבצע פעולות ולקבל מידע מרכיבים שונים של המחשב
- כגון: ציור על המסך, קריאת קובץ, ביצוע חישובים
- הAPI מאפשר למשתמש לבצע את הבקשות מבלי לקחת בחשבון את רכיבי החומרה המסויימם במחשב שבו התוכנה תרוץ
- לדוגמא: המתכנת רוצה שהתוכנה תצייר ריבוע כחול בגודל 100x100:
  - המתכנת משתמש בAPI כדי לבקש ממערכת ההפעלה לבצע את המשימה
  - מערכת ההפעלה (בעזרת ממשק החומרה) שולחת למסך המסויים המחובר למחשב את ההודעה שתגרום לו לבצע זאת, בהתאם לאיך צריך "לנסח" אותה עבור המסך המסויים

אבטחת מידע

- אספקט חשוב של מערכות הפעלה הוא אבטחת מידע
- מחשבים רבים מכילים מידע רגיש, אשר לא היינו רוצים שיהיו נגישים לכל אחד
  - החל מתמונות מביכות ועד נתונים סודיים שיכולים לשמש לסוגי התקפות שונות
- בנוסף מערכות קריטיות רבות משתמשות במחשבים (כגון בנקים ומערכות רפואיות)
- מסיבה זאת, קיים סיכון גדול שייעשה במערכות מחשב שימוש לא ראוי (בזדון או בשוגג) שיוביל לפגיעה חמורה. לדוגמא
  - מחיקה של מסכמים חשובים
  - קריאה של נתונים סודיים
  - גניבה של כספים
  - השבתה של מערכת קריטית על ידי גורם זדוני (במטרה גרום לנזק שינבע מהשבתת המערכת, או כדי לאפשר סחיטה של בעלי המערכת)

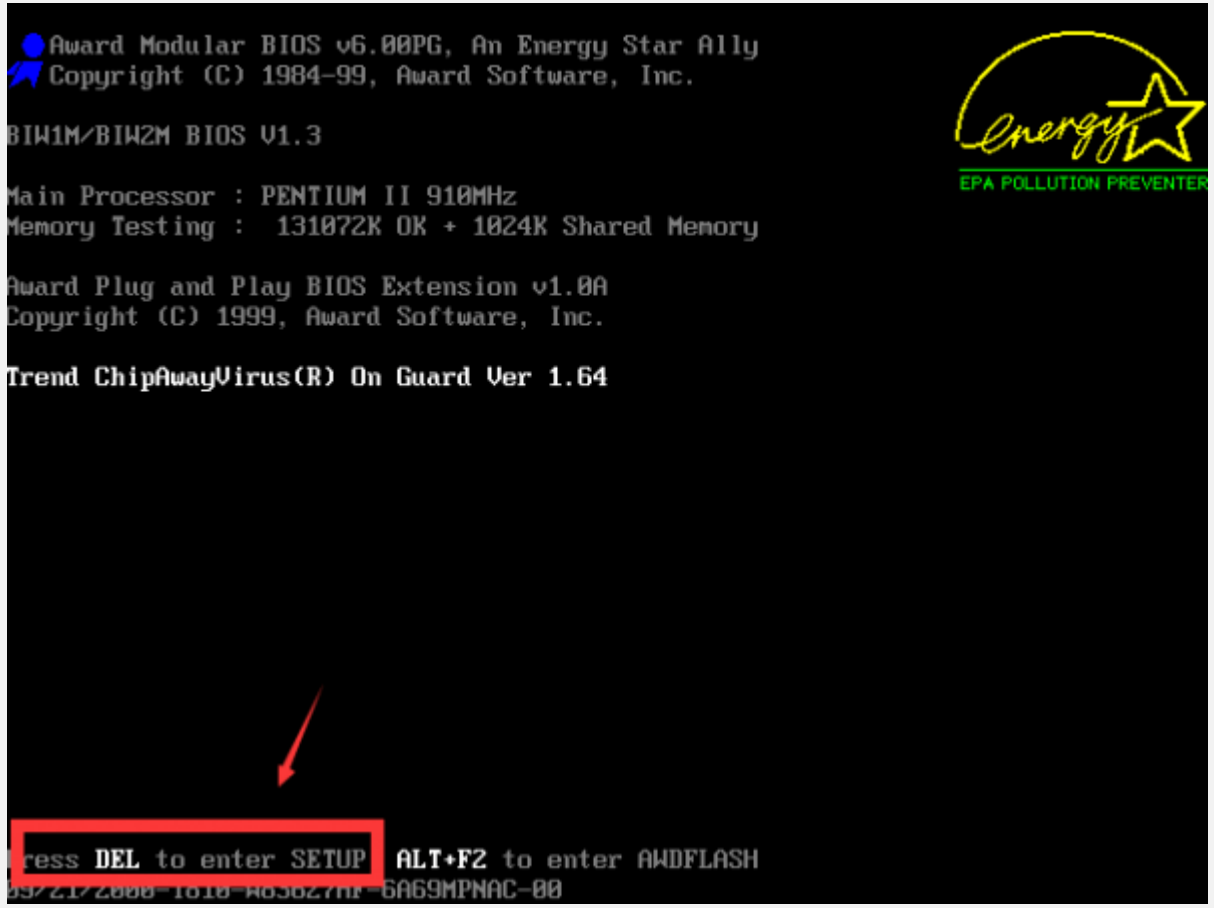
אבטחת מידע

- מערכות הפעלה כוללות מנגנונים שונים המיועדים להגן על המערכת מפני בעיות אלו
- כגון:
  - סיסמא: מאפשרת גישה למערכת רק עבור מי שיודע את הסיסמא, ובכך מונעת מגורמים זרים לגשת אליה
  - ניהול הרשאות: במערכת ההפעלה מוגדרות רמות הרשאה שונות למשתמשים, כאשר לכל פעולה מוגדרת מה רמת ההרשאה הנדרשת כדי לבצע אותה. לדוגמא
    - בארגונים גדולים העושים שימוש במערכות מחשוב מרובות משתמשים, בדרך כלל רוב העובדים לא יכולים להתקין תוכנה חדשה במחשב, באופן זה קטן הסיכון שעובד יתקין (בזגון או בשוגג) תוכנה אשר תעשה נזק למערכת או תקבל גישה לנתונים סודיים
  - חומת אש (Firewall): תוכנה המיועדת למנוע גישה דרך האינטרנט של גורמים זדוניים

BIOS

- Firmware (בעברית קושחה) היא תוכנה המותקנת בחומרה באופן ישיר
- מבחינת גמישותה לשינויים, קושחה היא מצב ביניים בין תוכנה - שקל מאוד לשנותה - ובין חומרה - שלא ניתן לשנותה
- קושחה היא תוכנה שצרובה פיזית על רכיב חומרה והיא לא תמיד ניתנת לשינוי
- כאשר מחשב נדלק, הוא מריץ קושחה הצרובה בו בשם BIOS (Basic Input-Output System – מערכת קלט-פלט בסיסית) אשר מבצעת מספר משימות:
  - זיהוי התקני חומרה
  - קביעת הגדרות חומרה בסיסיות והפעלה או כיבוי של התקנים שונים
  - קביעת סדר סריקת התקני האחסון, כלומר באיזה התקן יחפש המחשב את מערכת ההפעלה
  - בדיקה בסיסית של תקינות החומרה
  - טעינת רשימה של פונקציות מהדיסק הקשיח אל כרטיסי הזיכרון, כולל מידע עבור מערכת ההפעלה
  - הרצת מערכת ההפעלה

BIOS



- תהליכים מסויימים של המחשב ממשיכים לרוץ גם כשהוא כבוי (כגון שעון)
- לשם כך מערכת הBIOS מחוברת לסוללה, אשר מאפשרת לה להריץ תהליכים אלה גם כשהמחשב כבוי
- בהדלקת מחשב, לרוב ניתן לעצור את הריצה הסטדנרטית ובמקום זה לפתוח ממשק של תוכנת הBIOS המאפשר שינוי ועדכון של הגדרות שונות (כגון, איזה טמפרטורה מוגדרת כחימום יתר, בה המחשב ייכבה) וטיפול בבעיות המונעות ריצה תקינה של מערכת הפעלה

מערכת הפעלה צרובה

- תוכנה צרובה (Embedded Program) היא תוכנה אשר צרובה באופן ישיר על המחשב, במקום להיות מאוחסנת ברכיב זיכרון
- כאשר במחשב יש תכונה צרובה, שינוי של התוכנה (להחליף לתוכנה אחרת, או לעדכן את התוכנה) הופך להיות יותר מורכב/יקר ולעיתים בלתי אפשרי
- היתרון של תוכנה צרובה הוא בביצועים, תוכנה צרובה יכולה לפעול בצורה הרבה יותר יעיליה
- מחשבים פשוטים המיועדים להרצת תוכנה אחת בלבד (לדוגמא: מחשבון דיגיטלי) פעמים רבות משתמשים בתוכנה צרובה, על מנת לשפר ביצועים או לחסוך ברכיבים הנדרשים לבניית המחשב
- מערכת הפעלה צרובה היא מערכת הצרובה ישירות על הרכיב, ובכך
  - מצד אחד משפרת את ניצול המשאבים
  - מצד שני, מספקת חלק מהיתרונות של מערכת הפעלה כגון הרצת תוכנות שונות והרצת תהליכים במקביל
- נמצאות במוצרי חשמל רבים כגון מצלמות ומדפסות, וברכיבים חשמליים של מכשירים שונים, כגון רכיבים של מכונויות

תקשורת בין מחשבים

- כיום רוב המחשבים מחוברים למחשבים אחרים באמצעות רשת
- רשת מקומית (LAN – Local Area Network), למשל בעזרת כבלים או טכנולוגית WiFi
  - מספר מחשבים ביתיים ו/או מכשירים "חכמים" שונים כולם מחוברים ברשת ביתית
  - עשרות או מאות מחשבים של ארגון (כגון חברה מסחרית, מוסד ממשלתי ומוסד אקדמאי) מחוברים ברשת
- רשת מרוחקת (WAN – Wide Area Network), למשל בעזרת סיבים אופטיים, קווי טלפון ורשת סלולרית
  - לרוב רשתות מקומיות יהיו גם מחוברות לרשתות מרוחקות, כגון רשת האינטרנט
  - אחד התפקידים של מערכת ההפעלה הוא לנהל את התקשורת בין המחשבים

תקשורת בין מחשבים

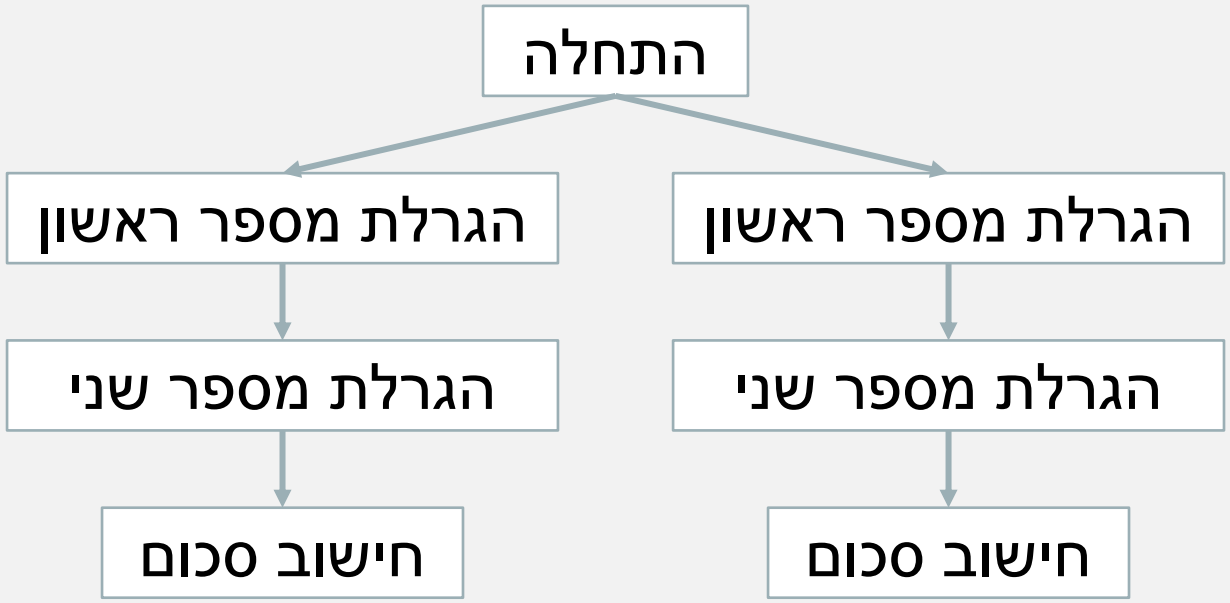
- ברשת מחשבים מקומית או רחבה, לתהליכים השונים הרצים על המחשבים השונים אין מרחב זיכרון משותף או שעון משותף
- לכן האופן שבו תהליכים כאלה יוכלו לשתף פעולה ביניהם חייב להיות שונה לחלוטין מהאופן בו תהליכים הרצים על אותו מחשב משתפים פעולה
- לכן מרכיב התיקשורת במערכת ההפעלה צריך להתמודד עם בעיות מורכבות מסוג אחר לגמרי
- מצד שני, רשתות מחשבים תורמות באופן משמעותי ליעילותו ולמהירותו של כל מחשב ומחשב בודד, בכך שהם מגדילים את כמות המשאבים העומדים לרשותו
- על ידי התחברות לרשת, מחשב בודד מגדיל בצורה משמעותית את כמות הקבצים והתוכניות העומדים לרשותו
- במקרים רבים, המחשב הבודד יוכל גם להפעיל תוכניות על מעבדים נוספים, ולקבל שרותים אשר הוא עצמו אינו מסוגל לספק

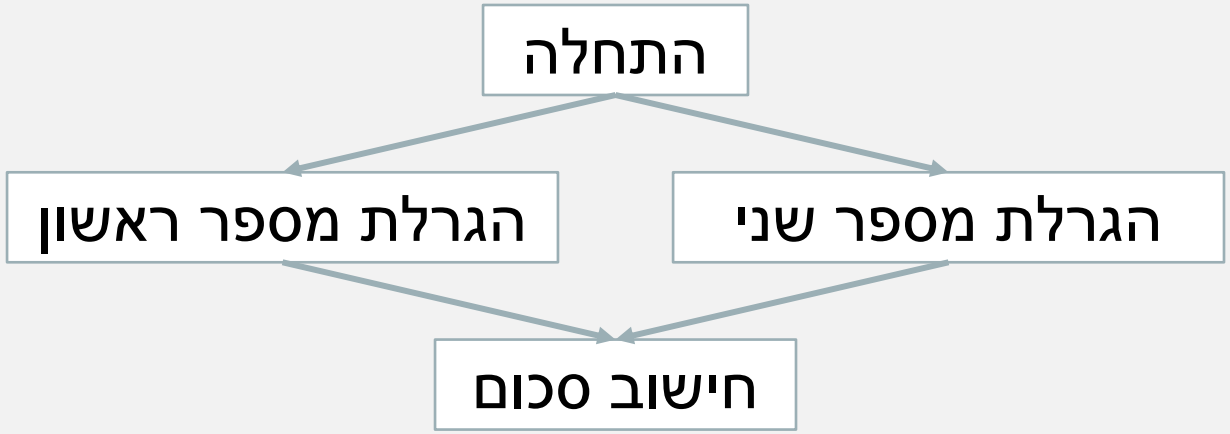
מושגים חשובים

- **תוכנה מרובת משתמשים Multi-user software**
- תוכנה אשר מאפשרת למשתמשים רבים להשתמש בה
  - מערכות הפעלה במחשבים אישיים הן לרוב תוכנות מרובות משתמשים – כמה משתמשים יכולים להשתמש באותו מחשב ב"גרסא שלהם" של מערכת ההפעלה (למשל הם יראו קבצים שונים)
- **Multiprocessors**
- מחשבים המכלים מספר מעבדים (או ליבות) המאפשרים הרצה של תהליכים שונים במקביל
- **Multiprogramming**
- הטכניקה בה אותו מחשב מריץ על מעבד אחד מספר תוכנות (הרצה מקבילית)
- **Multitasking**
- דומה לMultiprogramming, אך בעוד שMultiprogramming, מתייחס להרצה במקביל של מספר תהליכים, Multitasking מתייחס להרצה במקביל של מספר משימות באותו תהליך

מושגים חשובים

- דוגמא אינטואיטיבית להבדל בין Multiprogramming ו Multitasking
- נניח שיש לנו תוכנה אשר מגרילה שני מספרים ומחשבת את סכומם
- Multiprogramming:
  - הרצה של שני תהליכים שונים עבור התוכנה במקביל. כל אחד מהם יגריל מספר ראשון, ואז יגריל מספר שני, ולבסוף יחשב את סכומם
- Multitasking:
  - הרצה של תהליך אחד כאשר המשימות של הגרלת המספר הראשון והגרלת המספר השני מתבצעות במקביל, ולאחר ששתיהם הושלמו, מחושב הסכום





מערכות הפעלה נפוצות

- **מערכות הפעלה נפוצות עבור מחשבים ביתיים**
- Windows - מערכת ההפעלה הנפוצה ביותר במחשבים אישיים – נוחה לשימוש עבור המשתמש "הממוצע"
- macOS - מערכת ההפעלה בה משתמשים מחשבים של חברת Apple
- Linux – מערכת הפעלה המספקת שליטה הרבה יותר גדולה במחשב. פופולרית לשימוש ע"י משתמשים "מקצועיים"
- בניגוד לWindows ו macOS, לLinux הפצות רבות המספרות ממשקים שונים אשר כולם יושבים על אותו בסיס
- מאפיין ייחודי נוסף של Linux היא שהיא תוכנת קוד פתוח – הקוד המלא שלה נגיש בחינם לכל אחד

מערכות הפעלה נפוצות

- **מערכות הפעלה נפוצות עבור מכשירים "חכמים"**
- iOS – מערכת ההפעלה עבור מכשירים "חכמים" של Apple
- Android – מערכת ההפעלה הנפוצה ביותר למכשירים "חכמים"
  - מספקת יכולות שליטה עשירות יותר במכשיר לעומת iOS
  - מאפשרת יותר מגוון
- לדוגמא, גם טלפונים "חכמים" של Samsung וגם טלפונים "חכמים" של Xiaomi משתמשים באנדרויד, אבל יש הבדלים בממשק – כל חברה לפי ההחלטות שלה

קריאת מערכת

- קריאת מערכת (System Call) היא בקשה שתהליך יכול לבקש מהkernel כדי לבצע פעולה
- לכל תהליך יש רמת הרשאה שמגדירה מה הוא יכול לבצע, כל מערכת הפעלה מגדירה מה רמות ההרשאה המוגדרות בה
- לרוב, ככל שהרמה גבוהה יותר, התהליך יכול לבצע פחות פעולות, כאשר תהליך בעל רמת הרשאה 0, יכול לבצע כל פעולה
- לכל פעולה מוגדרת מה רמת ההרשאה הנדרשת עבורה
- כאשר תהליך רוצה לבצע פעולה שאין לו הרשאה מתאימה עבורה, הוא צריך לבצע קריאת מערכת
- כ"כלל אצבע"
  - פעולות המתקשרות עם עזרי פלט/קלט (כגון מסך וגישה לקונן הקשיח) יותר מוגבלות ותהליך יצטרך לבצע קריאת מערכת עבורם
  - פעולות שאינן מתקשרות עם עזרי קלט/פלט (כגון חישוב או השמה למשתנה) פחות מוגבלות ותהליך יכול לבצע אותם ללא קריאת מערכת

קריאת מערכת

- קטגוריות של קריאות מערכת:

1. שליטה בתהליכים. למשל: הרצת והפסקת תהליכים
2. ניהול קבצים. למשל: קריאה וכתיבה לקובץ
3. ניהול התקנים. למשל: קליטת נתונים מהמקלדת
4. קבלת מידע. למשל: קבלת הזמן הנוכחי משעון המחשב
5. תקשורת. למשל: שליחת הודעה
6. אבטחה. למשל: בדיקת הרשאות של קובץ