



TEL AVIV UNIVERSITY

מערכות הפעלה

ערן טרומר
סמסטר א' תשע"ב

הרצאה 2

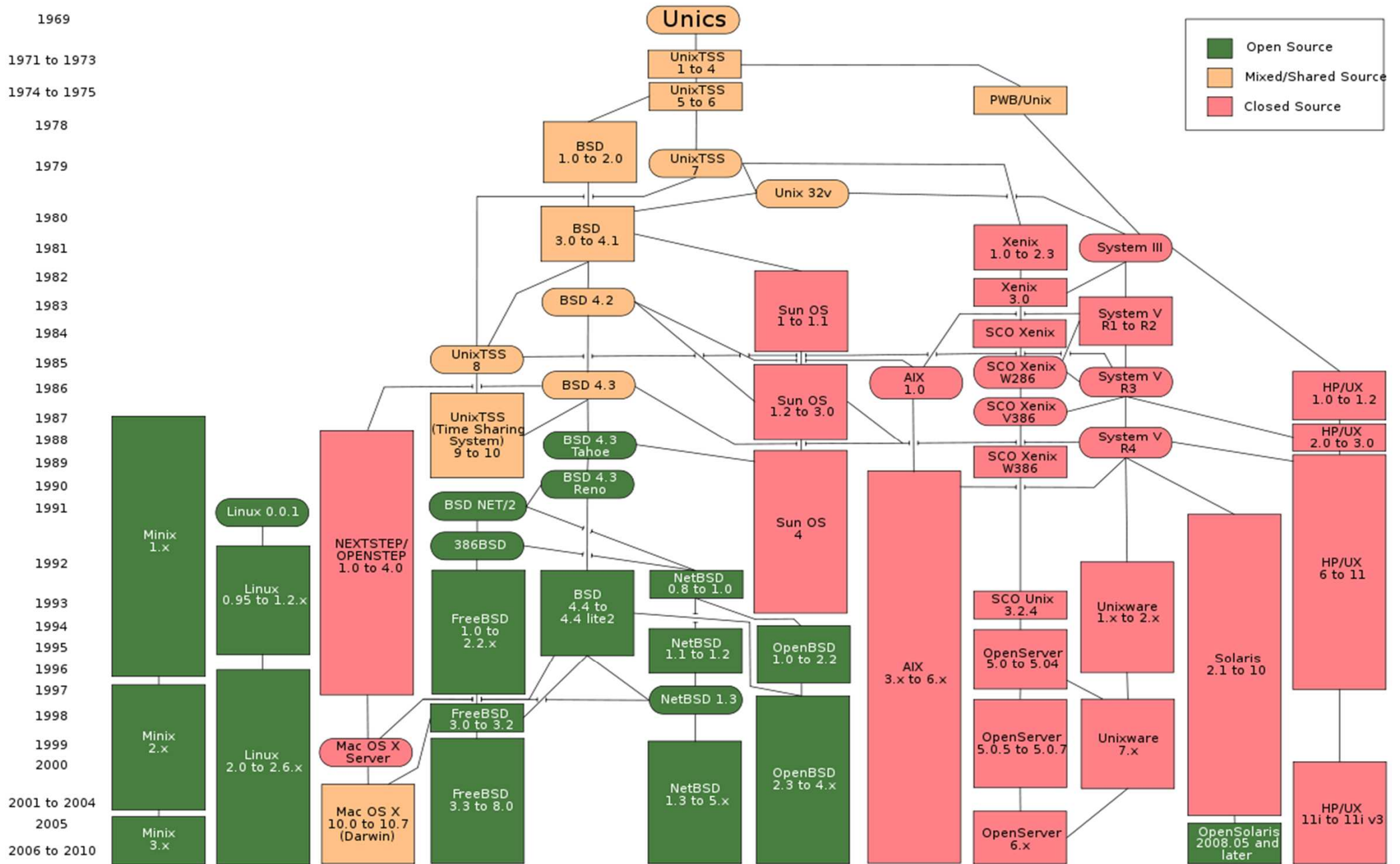
מבוא (המשך)

קלט/פלט

מערכות Unix

- ❖ מגוון גדול של מערכות הפעלה תואמות זו לזו ברמת קוד המקור (לא ברמת קוד המכונה), שמוצאן מערכות הפעלה בשם Unix שפותחה במעבדות המחקר של חברת Bell
- ❖ AIX של IBM, Solaris של Sun, HP-UX של HP, Tru64 של SGI, IRIX של Compaq
- ❖ מחשבים אישיים, תחנות עבודה, שרתי 7/24 עם מאות מעבדים, מחשבי-על עם מאות אלפי מעבדים
- ❖ תקן POSIX (קריאות מערכת, מערכות קבצים, ספריות, תוכניות בסיסיות)
- ❖ מערכת ההפעלה של המקינטוש (החל מגרסה 10) היא מערכת יוניקס עם ממשק ייחודי

משפחת Unix (תמונה חלקית)



מערכות Linux

- ❖ מערכת הפעלה תואמת-יוניקס שפותחה החל מ-1991 על ידי סטודנט פניני בשם Linus Torvalds
- ❖ מחולקת חינם עם קוד המקור שלה
- ❖ חבילות הפצה (distributions) כוללות את מערכת ההפעלה, תוכנת התקנה, ותוכנות שימושיות
- ❖ ממחשבי כף יד ועד שרתים ומחשבי-על, כולל כמעט כל מחשב שיכול להריץ חלונות
- ❖ תוכנה חופשית. GNU General Public License (copyleft)
- ❖ קוד מקור לליבה: www.kernel.org
- ❖ הפצות שונות (Debian, Ubuntu, RedHat, CentOS, ...)
- ❖ הליבה של Android היא גרסה משופצרת של Linux

CP/M-1 DOS

❖ 1970's – 1980's

❖ מערכת הפעלה מינימלית מאד עבוד מחשבים זעירים, עשרות עד מאות קילובייטים של זיכרון

❖ אין הגנה על זיכרון או התקנים, אין מנגנון הרשאות ומשתמשים, אין ריבוי משימות

❖ צעד ענק לאחור בהנדסת מערכות הפעלה

❖ תחילת מהפכת "המחשב האישי"

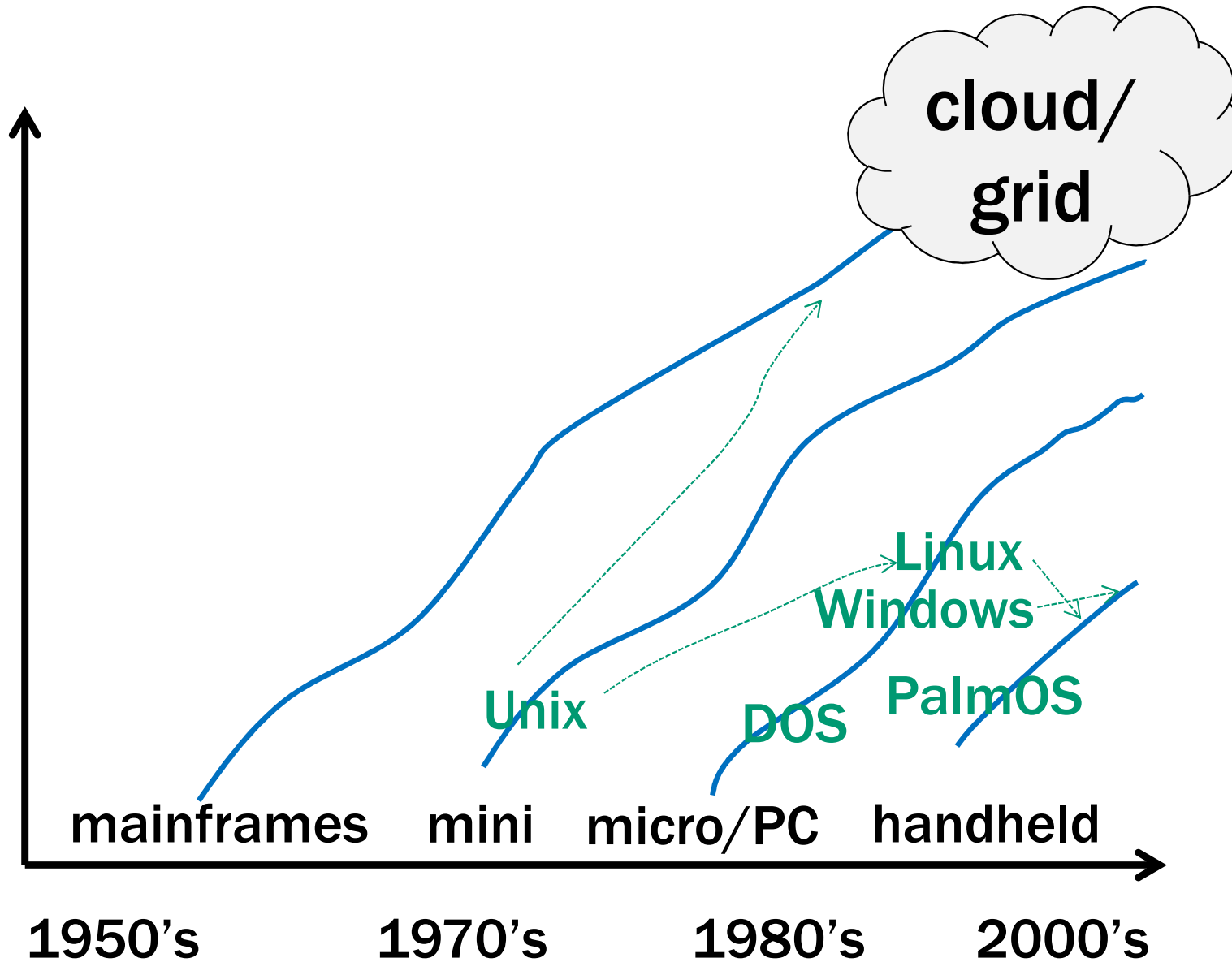
מערכות Windows

- ❖ מפותחות ומשווקות על ידי חברת מיקרוסופט
- ❖ גרסאות למחשבי כף יד/טלפונים, למחשבים אישיים (Windows 7, Vista, XP) ולשרתים קטנים ובינוניים
- ❖ משפחות:
 - Windows 3.1x, 95, 98, ME (השורשים ב-DOS עדיין מבצבים)
 - Windows NT, Windows 2000, ...
 - Windows Phone
- ❖ תאימות גבוהה ברמת קוד המכונה בין הגרסאות השונות
 - יתרונות: הפצה ובתחזוקה
 - חסרון: מטען היסטורי של ממשקים ישנים ומעקפים

מערכות הפעלה אחרות

- ❖ מערכות הפעלה למחשבים מרכזיים (mainframes); בעיקר לצורך שימור מערכות מחשוב ארגוניות ישנות. דוגמה: VAX
- ❖ מערכות הפעלה למערכות זמן אמת; שליטה בכלי טייס, רובוטים וכו'; דורשת יכולות תזמון מדויקות שאין למערכות הפעלה רגילות; אם אפשר לסבול כשלים נדירים ניתן להשתמש בגרסאות מתאימות של חלונות או לינוקס. דוגמה: VxWorks
- ❖ מערכות הפעלה למחשבים חלשים ומוגבלי אנרגיה (גם כאן יש גרסאות מתאימות של חלונות או לינוקס או יוניקס)
- ❖ מערכות הפעלה להרצת מערכות הפעלה אחרות: Virtual machines; נדון בכך בהמשך הקורס

מגמות היסטוריות



ישויות עיקריות במערכת ההפעלה

❖ תהליך (process): מחשב וירטואלי לתוכנית יחידה
(שונה ממכונה וירטואלית, virtual machine)

❖ חוט (thread): מעבד של מחשב וירטואלי מרובה מעבדים

❖ קובץ (file): התקן אחסון וירטואלי; מאורגנים במערכת קבצים

❖ קשר (connection): ערוץ תקשורת וירטואלי בין תהליכים, אולי
במחשבים שונים

❖ חלון (window): מקלדת, עכבר, ותצוגה וירטואליים. המשתמש
יכול לסמן שהוא מעוניין באינטראקציה עם חלון זה או אחר

❖ תור הדפסה (printer queue): מדפסת וירטואלית

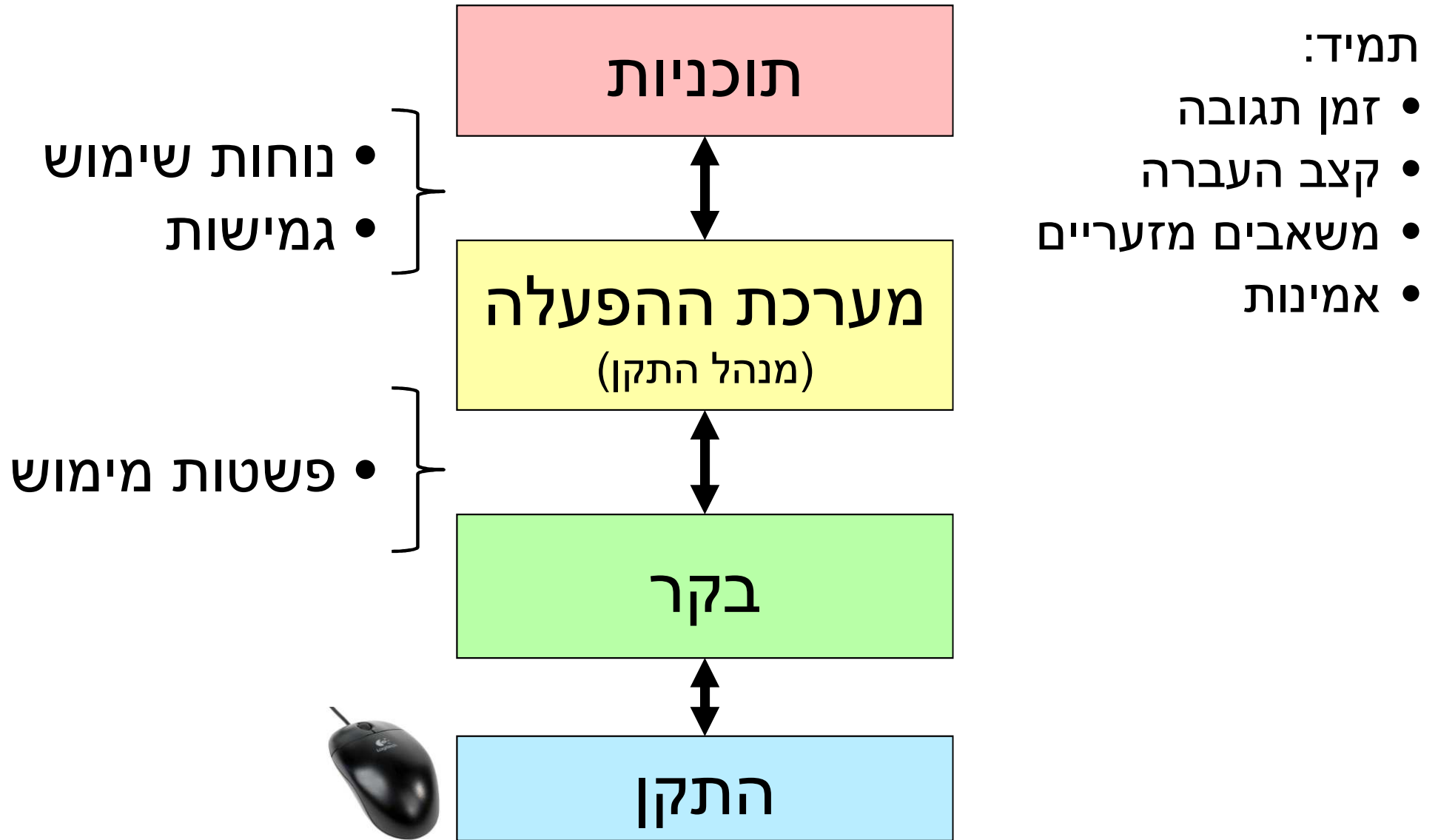
❖ משתמשים (users) וקבוצות משתמשים (groups) לבקרת גישה
למשאבים ומידע

סיכום עד כה

- ❖ 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 לחומרה
- ❖ רוב מערכות ההפעלה כיום שייכות לשתי משפחות עיקריות (לינוקס/יוניקס וחלונות), לכולן יכולות בסיסיות דומות, וכולן משתמשות באותם מנגנונים
- ❖ ההבדלים העיקריים כיום:
 - בסגנון הממשקים
 - יכולות בקצוות (שרתים גדולים במיוחד ומחשבים קטנים במיוחד)
 - תאימות, גמישות, הצע תוכנה, קהילת המפתחים והמשתמשים.

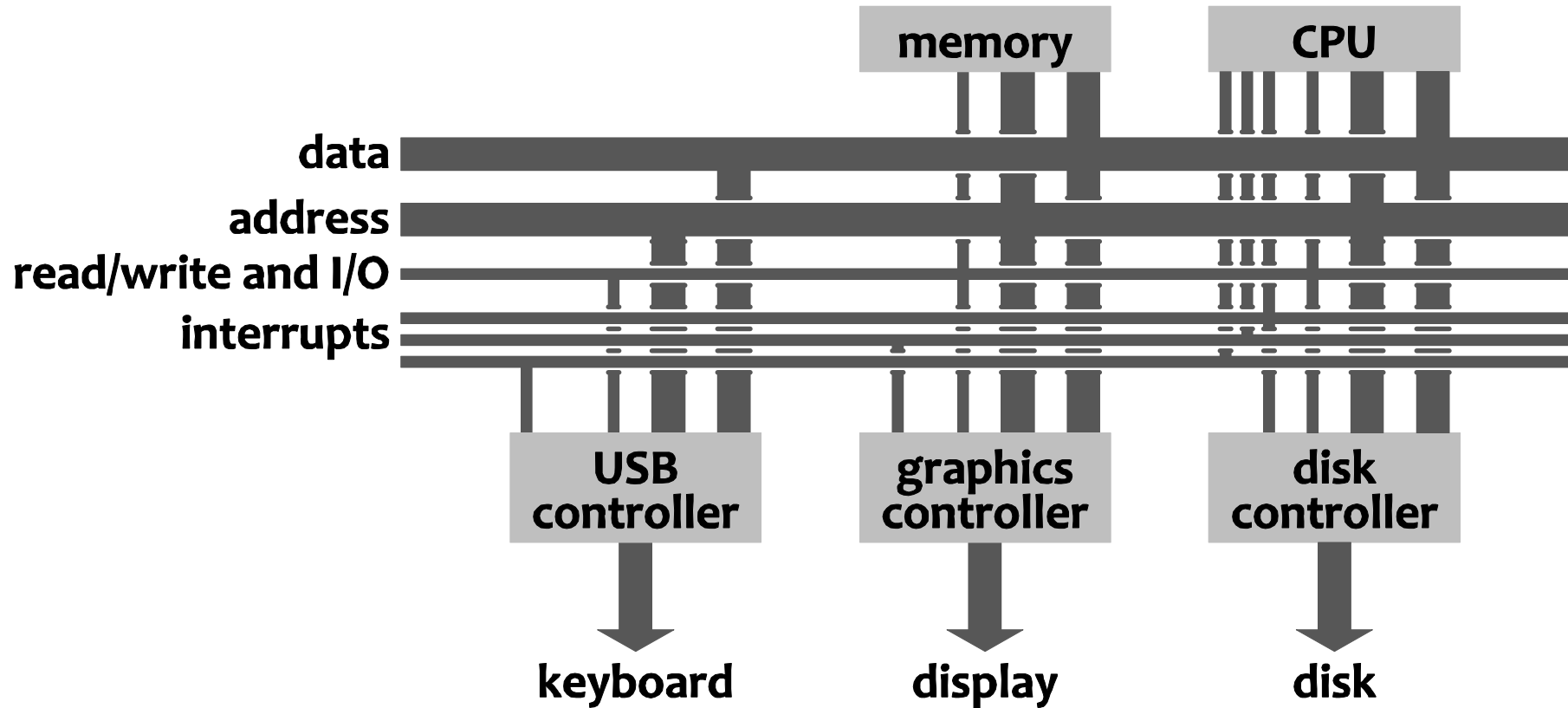
קלט/פלט

מטרות מערכת הקלט/פלט



העברת נתונים מהמעבד

אל ומאת בקרים



דגימה (polling)

❖ מערכת ההפעלה דוגמת את הבקר מדי פעם על מנת לבדוק האם קרה אירוע שמצריך תקשורת עם הבקר (הגיעו נתונים, הבקר מוכן לקבל פקודות נוספות, וכדומה)

חסרונות:

- ❖ דגימה איטית מדי עלולה להוביל לאובדן נתונים
 - פסיקת שעון רק כ-100 פעמים בשנייה
- ❖ דגימה כשלא קורה כלום מבזבזת משאבי מחשב
- ❖ לכן, שיטת תקשורת זאת כמעט שאינה בשימוש כיום

פסיקות (interrupts)

- ❖ הבקר מודיע למעבד בעזרת קו תקשורת מיוחד בפס שקרא אירוע שמצריך תקשורת עמו. המעבד מגיב בהפעלת שגרת פסיקה של מערכת ההפעלה שמטפלת באירוע
- ❖ תגובה מיידית (כמעט)
- ❖ בשימוש נרחב מאד
- ❖ חסרון: תקורה לשמירה ושחזור של מצב המעבד. לכן אי אפשר להעביר מידע רב טיפין-טיפין בפסיקות

אבחנה בין פסיקות שונות

❖ כאשר מספר בקרים משתפים קו פסיקה, מערכת ההפעלה צריכה לברר איזה בקר הפעיל את הפסיקה

❖ ניתן לחסוך את הבדיקה על ידי שימוש במספר קווי פסיקה בפס

❖ במעבדים עם כניסת פסיקה אחת (למשל מעבדי x86) משתמשים בבקר פסיקות שיש לו כמה כניסות פסיקה ויציאה אחת. בקר הפסיקות מודיע למעבד שאחד הבקרים הפעיל פסיקה, ומערכת ההפעלה יכולה לברר בעזרתו איזה בקר הפעיל את הפסיקה

מניעת הפעלה רקורסיבית של שגרות פסיקה

- ❖ פסיקה משעה את השגרה שרצה ומפעילה את שגרת הפסיקה
- ❖ רצוי לא להשעות את שגרת הפסיקה עצמה ולהפעילה שוב אם מתרחשת פסיקה נוספת לפני שהיא חוזרת
- ❖ הפעלה רקורסיבית עלולה לגרום להשחתת מבני נתונים (עוד על כך בפרק 5)
- ❖ מערכות הפעלה פשוטות משעות את הטיפול בפסיקות כאשר שגרת פסיקה רצה
- ❖ יש מערכות הפעלה המסווגות פסיקות על פי עדיפויות ומשעות רק פסיקות בעדיפות נמוכה יותר מזו המטופלת
 - פסיקת שעון היא בעדיפות עליונה

הפעלה דחוויה של שגרות

❖ אם הטיפול בפסיקה ארוך, עדיף לבצע כמה שיותר מהטיפול מחוץ לשגרת הפסיקה, על מנת שלא לדחות טיפול בפסיקות אחרות

❖ מערכת ההפעלה מבצעת בשגרת הפסיקה את הטיפול שיש לבצע מייד ואורזת ייצוג של שאר הטיפול במבנה נתונים שמייצג הפעלה דחוויה של שגרה; בזמן זה קבלת הפסיקות מושעת

❖ מערכת ההפעלה מבצעת את הקריאות הדחוויות לאחר הטיפול בפסיקות אך לפני החזרה לתוכנית המשתמש

❖ זהו למעשה מנגנון תזמון זעיר שמבטיח טיפול מהיר באירועים דחופים (פסיקות)

גישה ישירה לזיכרון (direct memory access)

- ❖ מערכת ההפעלה מורה לבקר להעביר בלוק גדול של נתונים בין התקן חיצוני (דיסק, רשת) ובין הזיכרון. הבקר מעביר את הנתונים ללא התערבות נוספת של המעבד
- ❖ יתרון: קצב גבוה ללא עומס על המעבד
- ❖ חיסרון: מייקר את הבקר
- ❖ דורש ניהול זכרון זהיר מאד כדי שהתקנים לא יכתבו/יקראו כתובות לא רצויות
- ❖ בשימוש נרחב בגישה לדיסקים, רשתות תקשורת מהירות, FireWire וכו'