



TEL AVIV UNIVERSITY

# מערכות הפעלה

ערן טרומר  
סמסטר א' תשע"ב

## 1 הרצאה

מבוא

# מקורות מידע

- ❖ הרצאות: עקרונות, רציונל, השוואה; דגש על Unix
- ❖ תרגילים: יישום, הדגמות, הכנה לתרגילים; דגש על Windows



❖ ספר הקורס:

מערכות הפעלה / סיון טולדו

ספרים אחרים:

- Tanenbaum / Modern Operating Systems
- Silberschatz / Operating System Concepts

❖ שקפי הרצאות

❖ שקפים והדגמות בתרגולים

❖ אתר הקורס: <http://cs.tau.ac.il/~apartzin/os2012/OS.htm>

❖ רשימת תפוצה: [http://groups.google.com/group/os\\_tau\\_2012a](http://groups.google.com/group/os_tau_2012a)

❖ הגדרות ממשק ומקורות נוספים – בהמשך

# ציון

- ❖ 7 תרגילים, שבועיים להגשה (יופחת ציון על איחור)
- ❖ עבודה עצמאית
- ❖ הגשת תשובות נכונות ועצמאיות לכל התרגילים היא תנאי לגישה למבחן
- ❖ ציון 100% מבחן
- ❖ כרבע מהמבחן יהיה קל למי שפתר את התרגילים יסודית

# מטרת הקורס

- ❖ הבנת תפקיד מערכת ההפעלה
- ❖ הבנת רבדי מערכת ההפעלה ומימשקיהם
- ❖ הבנת החלופות (המוצדקות תיאורתית וגם המציאותיות)
- ❖ הכרת ממשקי המערכת לתכנות יישומים
- ❖ התנסות בתכנות בשכבה נמוכה

# מערכת ההפעלה

- ❖ מגינה על חומרה על מנת לתחם תקלות ולהגביל השפעה בין תוכניות
- ❖ מנהלת את משאבי החומרה (זיכרון, זמן מעבד, דיסקים, רשת תקשורת) ביעילות והגינות
- ❖ מספקת לתוכניות ממשקים אחידים ונוחים לשימוש בהתקני חומרה (אבסטרקציה)

יש להבחין בין:

- ❖ ליבת מערכת ההפעלה (מוקד הקורס)
- ❖ שרותי המערכת ואתחולה (נזכיר)
- ❖ יישומים נלווים וחבילות הפצה (לא רלוונטי)

# חשיבות ההגנה על חומרה

ריבוי תוכנות ומשתמשים

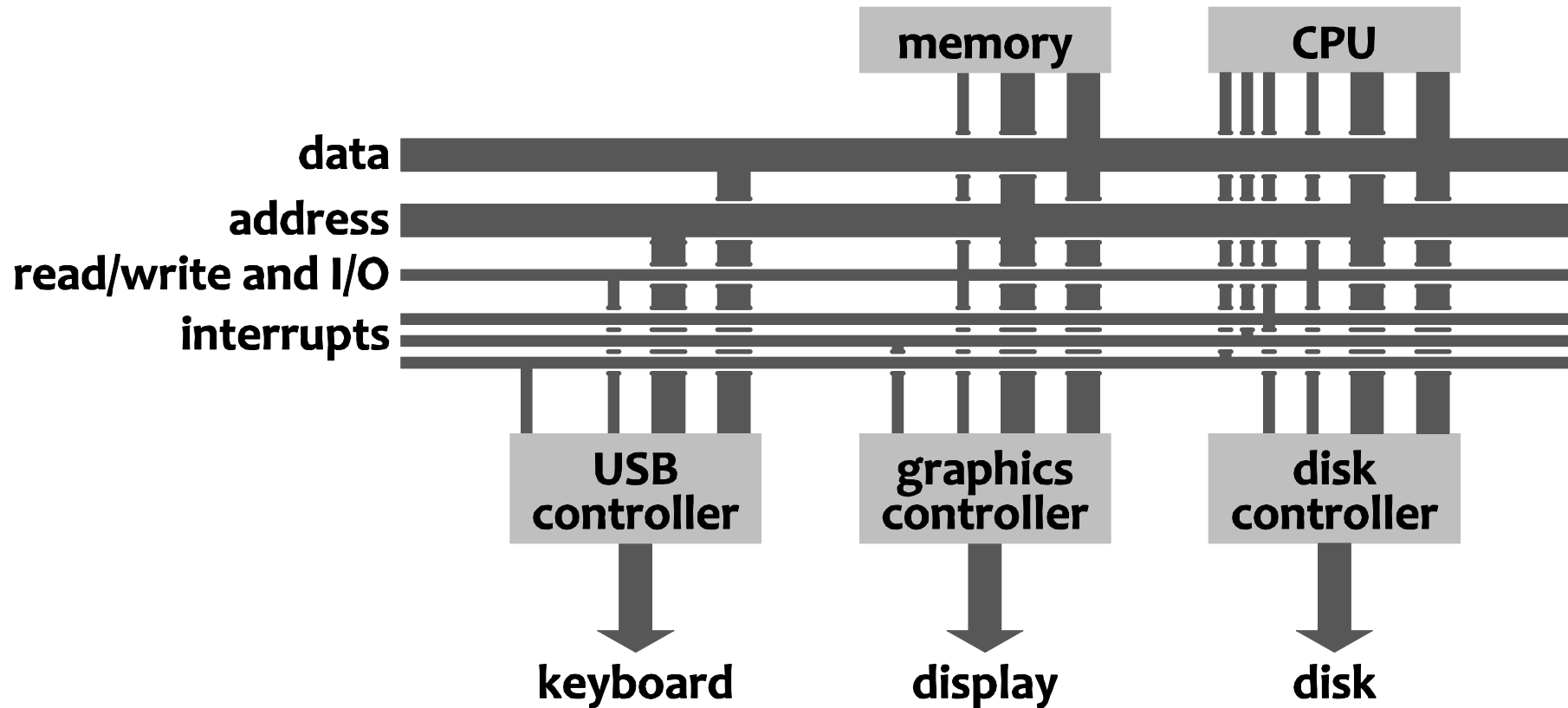
❖ תוכנית אחת לא יכולה לקרוא או לכתוב מידע של אחרת

❖ תקלה בתוכנית אחת לא משפיעה על אחרות

❖ השתלטות על משאב על ידי תוכנית אחת לא תוקעת תוכניות

אחרות

# 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
  - פקודות קריאה/כתיבה מזיכרון
- ❖ כל בקר מגיב לתחום כתובות מסוים

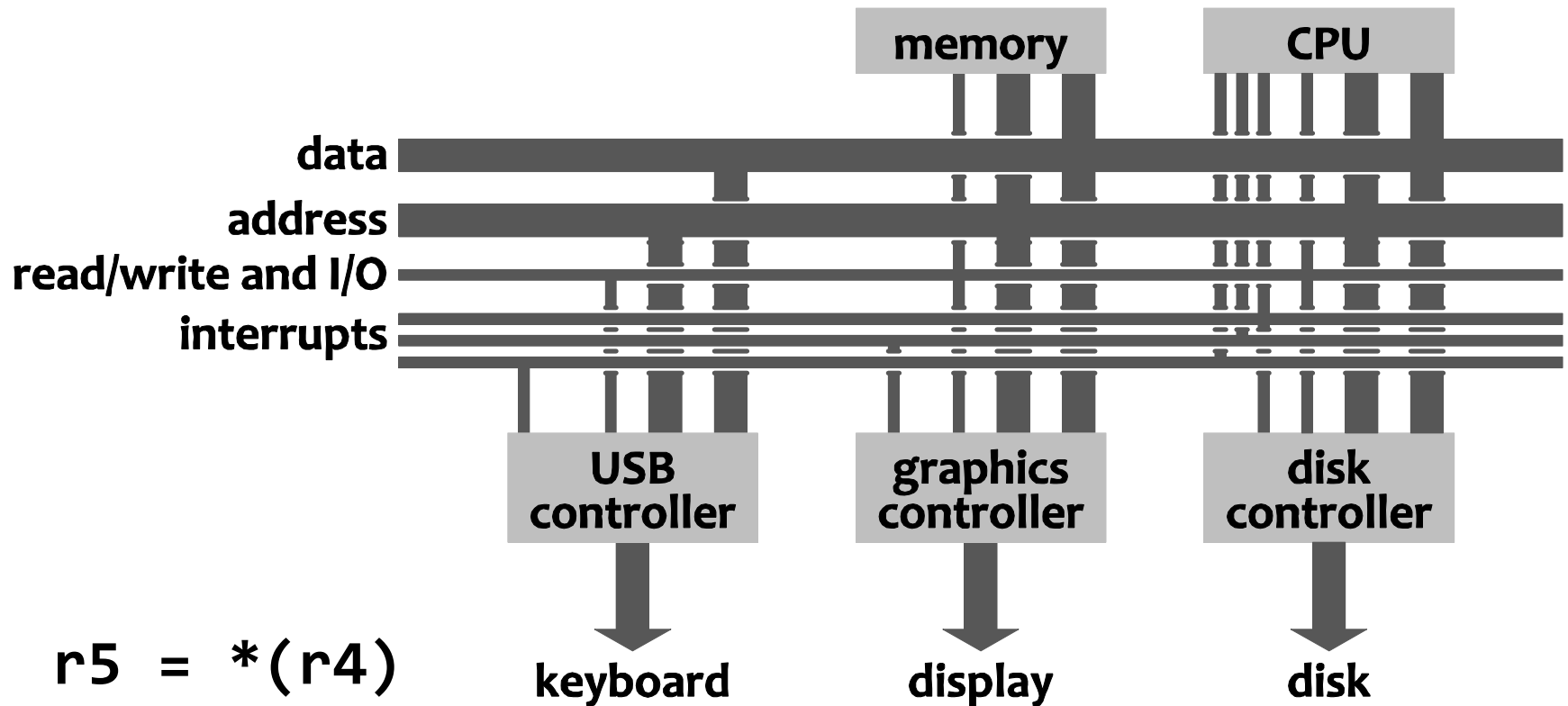
```
load    r5 = *(r4)
store   *(r4) = r6
mov     r3 = r8
add     r3 = r3 + r8
ioread  r5 = *(r4)
iowrite *(r4) = r6
```

כתובת במרחב  
הזיכרון

כתובת במרחב IO  
(port)



# תקשורת על הפס



load  $r5 = *(r4)$   
store  $*(r4) = r6$   
mov  $r3 = r8$   
add  $r3 = r3 + r8$   
ioread  $r5 = *(r4)$   
iowrite  $*(r4) = r6$

התקנים חיצוניים



בקרים



פס התקשורת המרכזי



מעבד

# הגנה על התקנים חיצוניים – מצב מיוחס

- ❖ שני מצבי ריצה: מצב משתמש (user mode) ומצב מיוחס (kernel mode)
- ❖ במצב מיוחס הכל מותר – המעבד מציית לכל פקודה
- ❖ במצב משתמש פקודות מסוימות אסורות
- ❖ 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 (נדון בהמשך)

# גישת להתקנים

- ❖ על מנת לגשת להתקן חיצוני, תוכניות קוראות לשגרה של מערכת ההפעלה
- ❖ הקריאה מתבצעת בנוהל מיוחד שמעביר את המעבד למצב מיוחד, על מנת שמערכת ההפעלה תוכל לתקשר עם הבקר
- ❖ מערכת ההפעלה לרוב תאפשר לא תרשה גישה ישירה לבקר
- ❖ מערכת ההפעלה מכילה מנהל התקן (device driver), אשר מתווך בין התוכנית לבקר

# קריאות מערכת (system calls)

- ❖ נוהל קריאה לשגרות של מערכת ההפעלה
- ❖ משתמש במנגנון פסיקת תוכנה (software interrupt) או דומה
- ❖ מעביר את המעבד למצב מיוחד; חזרה מקריאת מערכת מחזירה את המעבד למצב משתמש
- ❖ ציון השגרה שרוצים לקרוא לה על ידי מזהה מספרי ולא על ידי כתובת (למשל `3=read`, `4=write`, `5=open`, וכדומה)
- ❖ אין שימוש בכתובת על מנת למנוע מעבר לשגרה שאינה של מערכת ההפעלה במצב מיוחד
- ❖ בזמן קריאת המערכת, המעבד מריץ קוד של מערכת הפעלה, והתוכנית בהקפאה

# קריאות מערכת - דוגמה

User program writes to terminal (user mode)

```
...  
mov r0 = 4      ; read system call  
mov r1 = 1      ; terminal output  
int             ; software interrupt  
cmp r0, 0       ; check error code  
bne error       ; jump to error handler if error code is nonzero  
; now the read data is in r1  
...
```

Interrupt handler for software interrupt (kernel mode)

```
handle_interrupt:  
cmp r0, 3       ; is it a read call?  
je sys_read     ; yes, handle it  
Cmp r0, 4       ; is it write call?  
je sys_write    ; yes, handle it  
...
```

# הגנה על חומרה: המעבד

❖ אם תוכנית נתקעת בלולאה אין-סופית תוכניות אחרות ממשיכות לרוץ

❖ אחת לפרק זמן קבוע פסיקת שעון מעבירה את המעבד לביצוע שגרה של מערכת ההפעלה; המעבר לשגרת הפסיקה מעביר את המעבד למצב מיוחס, חזרה מחזירה אותו למצב הקודם

❖ הכתובת של השגרה הזו שמורה באוגר שלא ניתן לשנות במצב משתמש

❖ מערכת ההפעלה אינה "רצה ברקע"

❖ אותו מנגנון מגן על תהליכים פנימיים במערכת ההפעלה

(לשם פשטות אנו כגע מניחים מעבד בודד, ליבה בודדת, בלי SMT)

# הגנה על זיכרון (memory protection)

❖ יש להגן על:

- מערכת ההפעלה מפני תוכניות
- בקרים בעלי ממשק זכרון מפני תוכניות
- תוכניות אחת מרעותה

❖ לכן כל גישה לזיכרון דורשת אישור מערכת ההפעלה

❖ דורש מימוש יעיל (תקורה נמוכה)

❖ שיתוף פעולה בין מערכת ההפעלה והמעבד

❖ כל גישה לזיכרון נבדקת על ידי המעבד מול טבלאות שמערכת

ההפעלה מתחזקת – פרטים בהמשך

❖ שינוי הטבלאות דורש מצב מיוחד

# 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 (האלטרנטיבה, microkernel, לא הצליחה)
- ❖ הממשק לתוכניות בעיקר על ידי קריאות מערכת
- ❖ דוגמאות: התקני אכסון; כרטיסי רשת; כרטיסי מסך



# ממשקים משופרים לגישה לחומרה

❖ הממשק שהשגרות הללו מספקות יכול לספק שירותים שהחומרה אינה מסוגלת לספק בעצמה

❖ דוגמאות:

- אחסון או תקשורת נתונים אמינה בעזרת חומרה לא אמינה
- הצפנת נתונים לאחסון או תקשורת לא מאובטחת
- מערכות קבצים

❖ יכולות כאלה כלולות במערכות הפעלה משיקולי ביצועים או שווק

# ניהול יעיל והוגן של חומרה

## ❖ יעילות:

- מירב את הנצילות של משאבי חומרה לטובת תוכניות רגילות ומשתמשים.

דוגמה: חלוקת הזכרון, קווי תקשורת, תזמון גישה לדיסק

- מזעור תקורה – משאבים שמערכת ההפעלה עצמה צורכת או שמתבזבזים סתם בגלל ניהול כושל

## ❖ הגינות חלוקת המשאבים:

- בין משתמשים
- בין תוכניות שרצות בו-זמנית

## ❖ אכיפת מדיניות:

- לא שילמת לא קיבלת
- תוכניות אינטראקטיביות לעומת חישוב רקע

