I/O devices

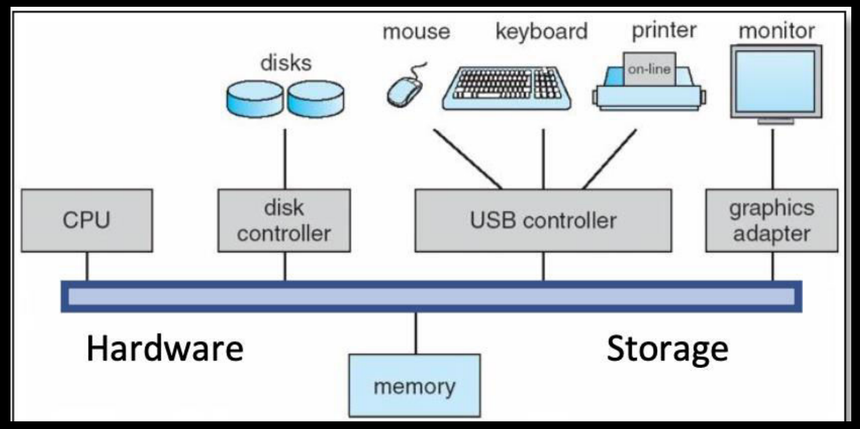
מהם התקני I/O?

- מכשירים המספקים קלט בלבד)מקלדת, עכבר, שלט(
- מכשירם המספקים רק פלט)מסך, רמקולים(
- מכשירים שמספקים גם וגם)הדיסק, כרטיס הרשת (

התממשקות התקנים

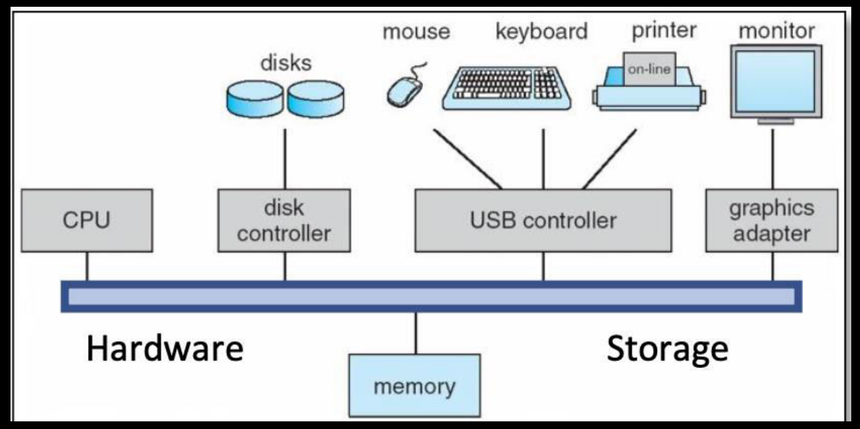
BUS

- קו התקשורת המחבר התקנים למערכת
- מחבר רכיבים שונים במחשב
- דרכו עובר מידע.



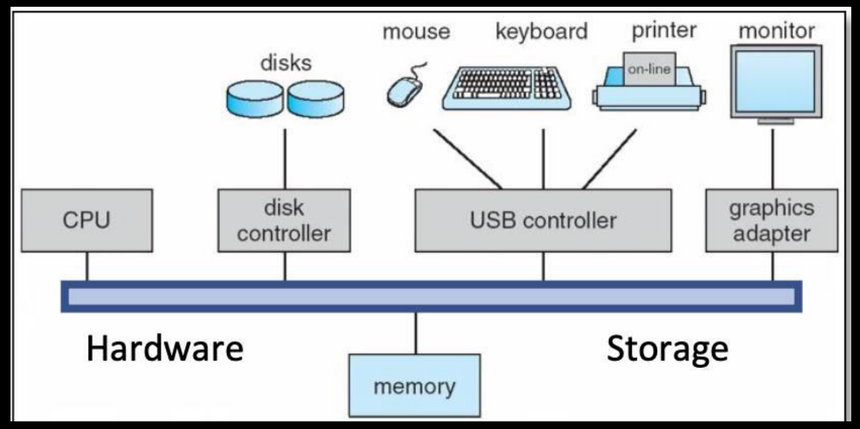
יתרונות בBUS

- קל להוסיף רכיבים חדשים
- מעבר בין מחשבים הוא קל אם לכולם משתמשים באתו סטנדרט BUS
- זול למימוש

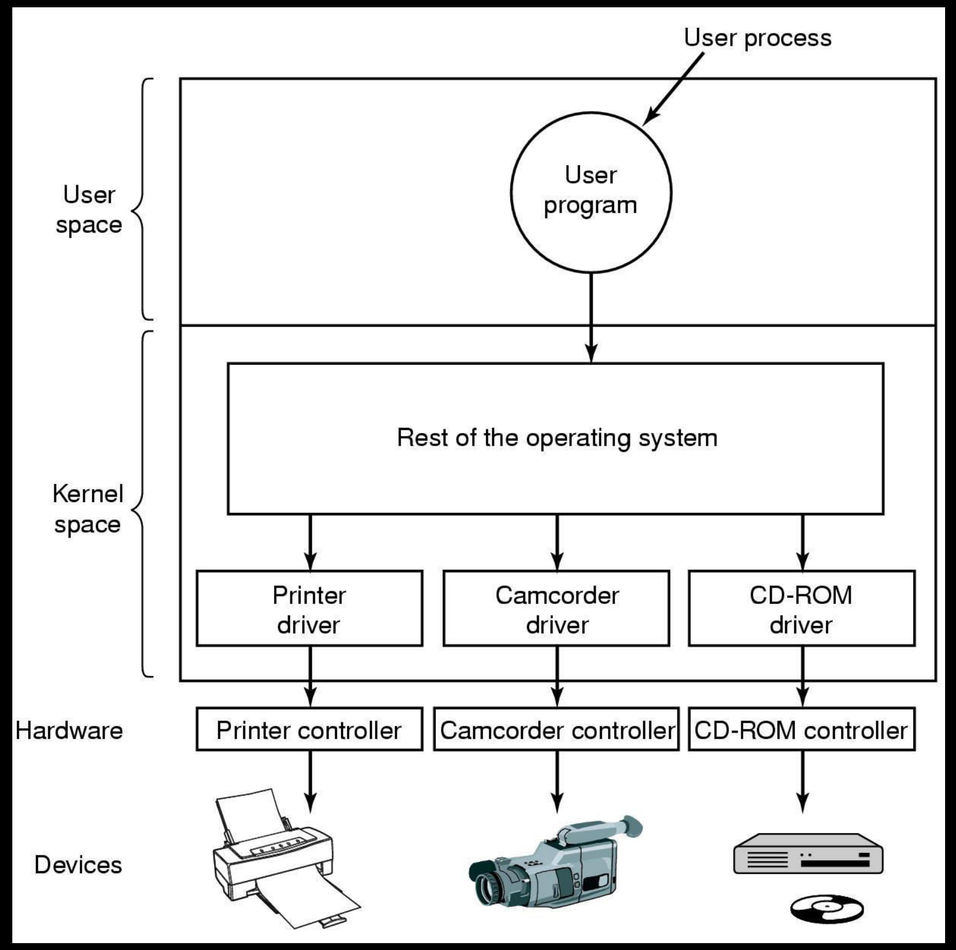


חסרונות בBUS

- צוואר בקבוק במערכת
- אמור להתאים לכל סוגי ההתקנים -גנרי



תפקיד מערכת ההפעלה ביחס לIO



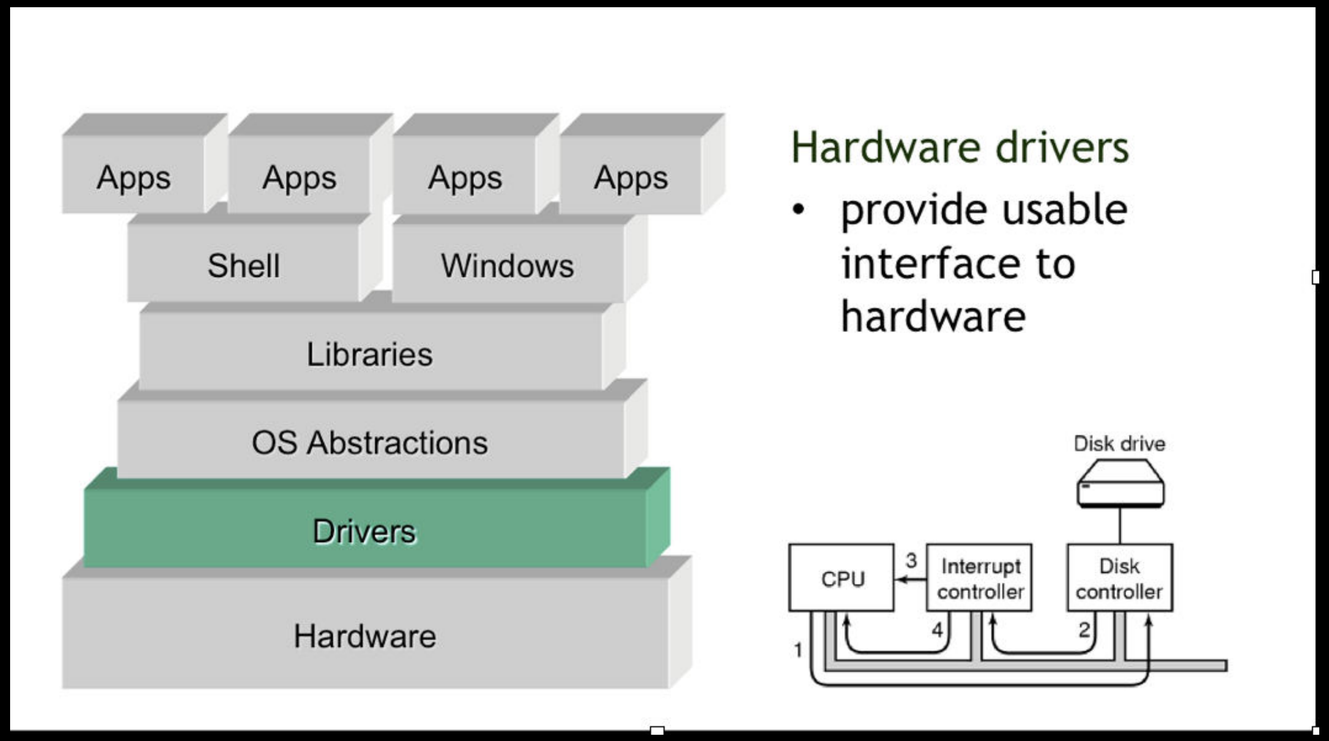
- להציג אבסטרקציה לוגית להתקנים
- החבאת תוכן ההתקן והמימוש שלו מהמשתמש
- החבאת טיפול בשגיאות
- לאפשר הפעלת IOותוך כדי לנצל את המעבד
- לאפשר גישה להתקן במקביל ולהגן על הגישה אליו

Controllers

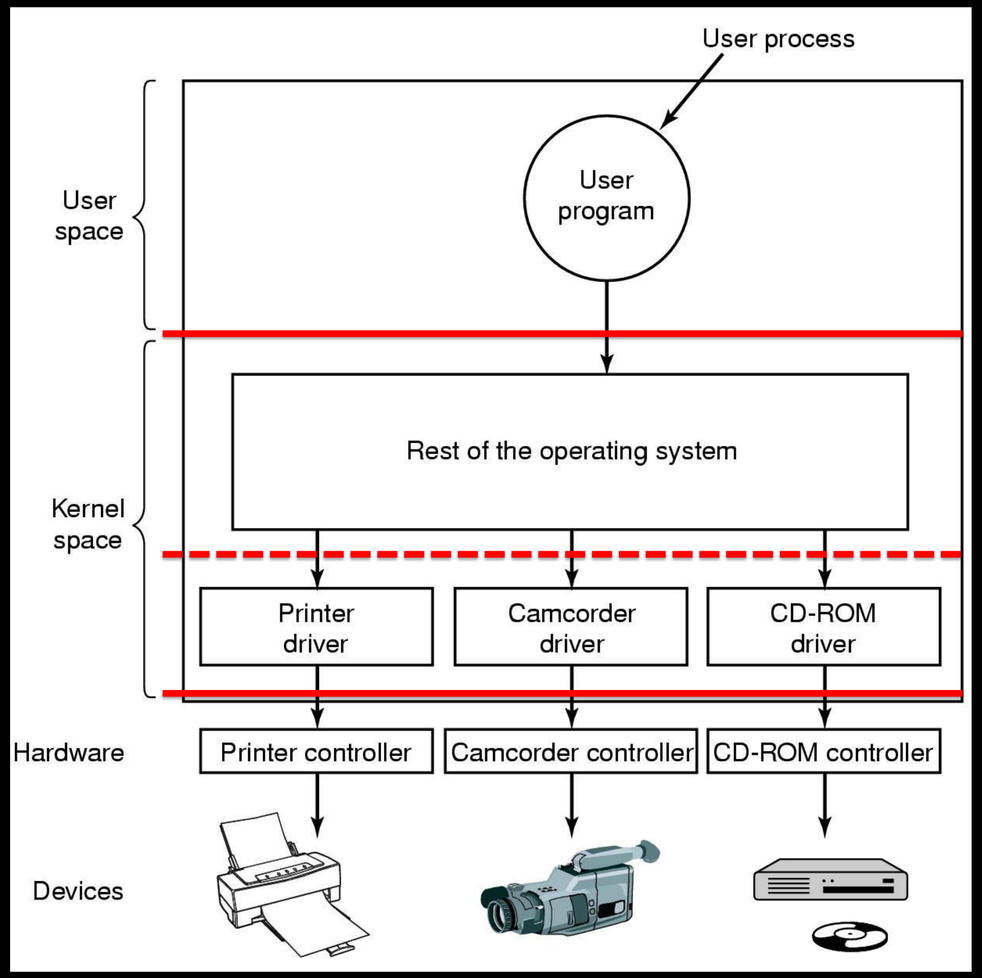
- התקן לרוב מורכב מרכיבים פיזיים מכאניים ,ואלקטרוניים
- בין ה controllerלהתקן יש ממשק low-level
- למשל בדיסק קשיח ה controllerצריך לדעת:
  - לקבל כתובת ,אוסף של בתים שייצג מידע ,ובתים לתיקון שגיאות
  - לתקן את השגיאות
  - להמיר את המידע לבלוק
  - להכניס בלוק לפי כתובת)מספר צילינדר ,וגזרה(

Drivers

- לאחר חיבור ההתקנים באופן חומרתי יש צורך לדעת לתקשר איתם
- ה driverמהווה את הדרך של מערכת ההפעלה ליצור קשר עם ההתקן

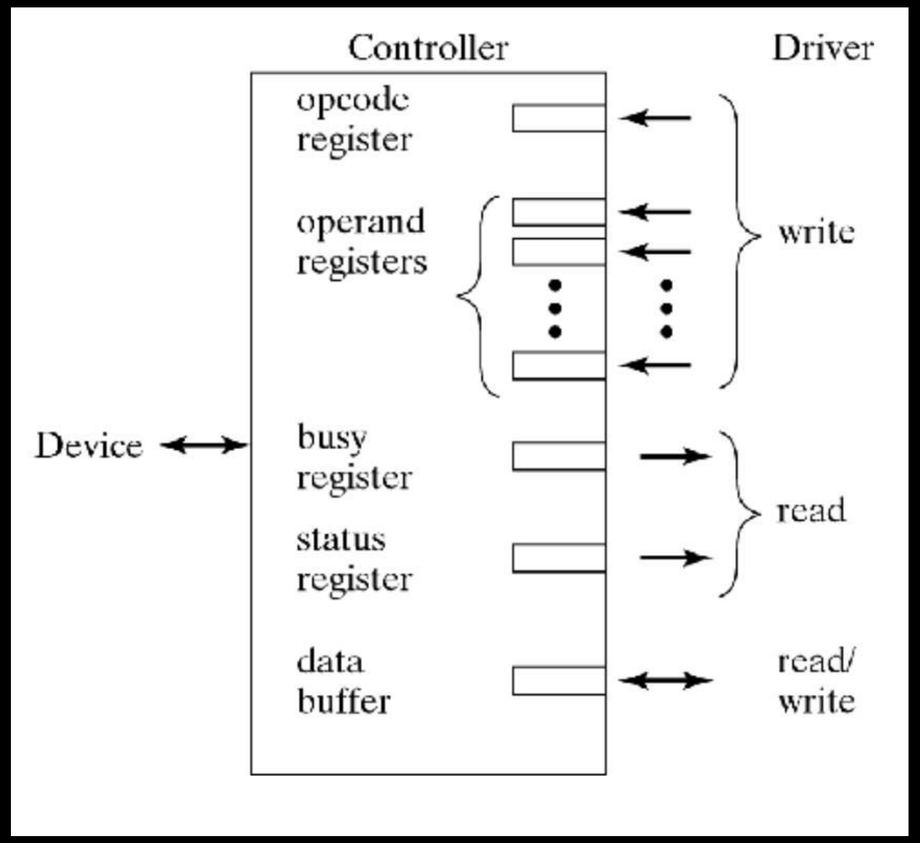


Drivers



- הקוד של ה Driverרץ על גבי הCPU ב kernel mode.
- לרוב נכתב על ידי הייצרן של ההתקן
- יש להם APIמוגדר וקבוע open, close, read, write, seek, flush :

Drivers



- התקשורת בין ה driverאל ההתקן עוברת דרך ה bus
- כשמתקבלת פקודה כמו read/ write block מתקשר עם ה controllerכדי לבצע
- כאשר ה CPUמריץ קוד של driverהוא יכול לבצע
  - פקודות רגילות כמו lw,sw, לכתובות שבפועל ימופו ל controllerולא ל.ram
  - פקודת IOמיוחדת

דוגמה Parallel Port

ממשק לשלוח או לקבל מידע על כבל

- לרוב משמש במדפסות
- ניתן להעביר מידע בכבל על ידי מספר קוים יעודים

- **Simple hardware has three control registers:**

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0

read/write data register (port 0x378)

$\overline{\text{BSY}}$	$\overline{\text{ACK}}$	PAP	OFON	$\overline{\text{ERR}}$	–	–	–

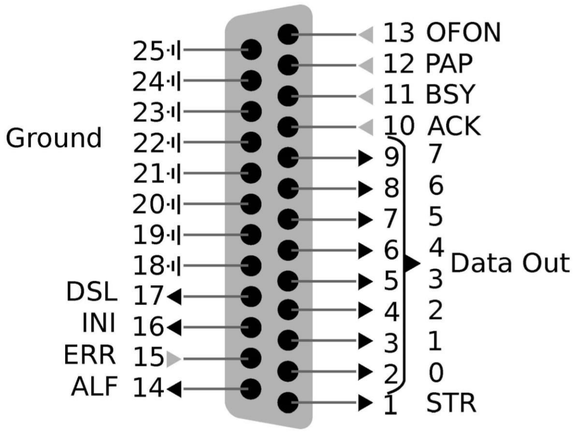
read-only status register (port 0x379)

–	–	–	IRQ	DSL	$\overline{\text{INI}}$	ALF	STR

read/write control register (port 0x37a)

[Messmer]

- **Every bit except IRQ corresponds to a pin on 25-pin connector:**



[image credits: Wikipedia]

דוגמה Parallel Port

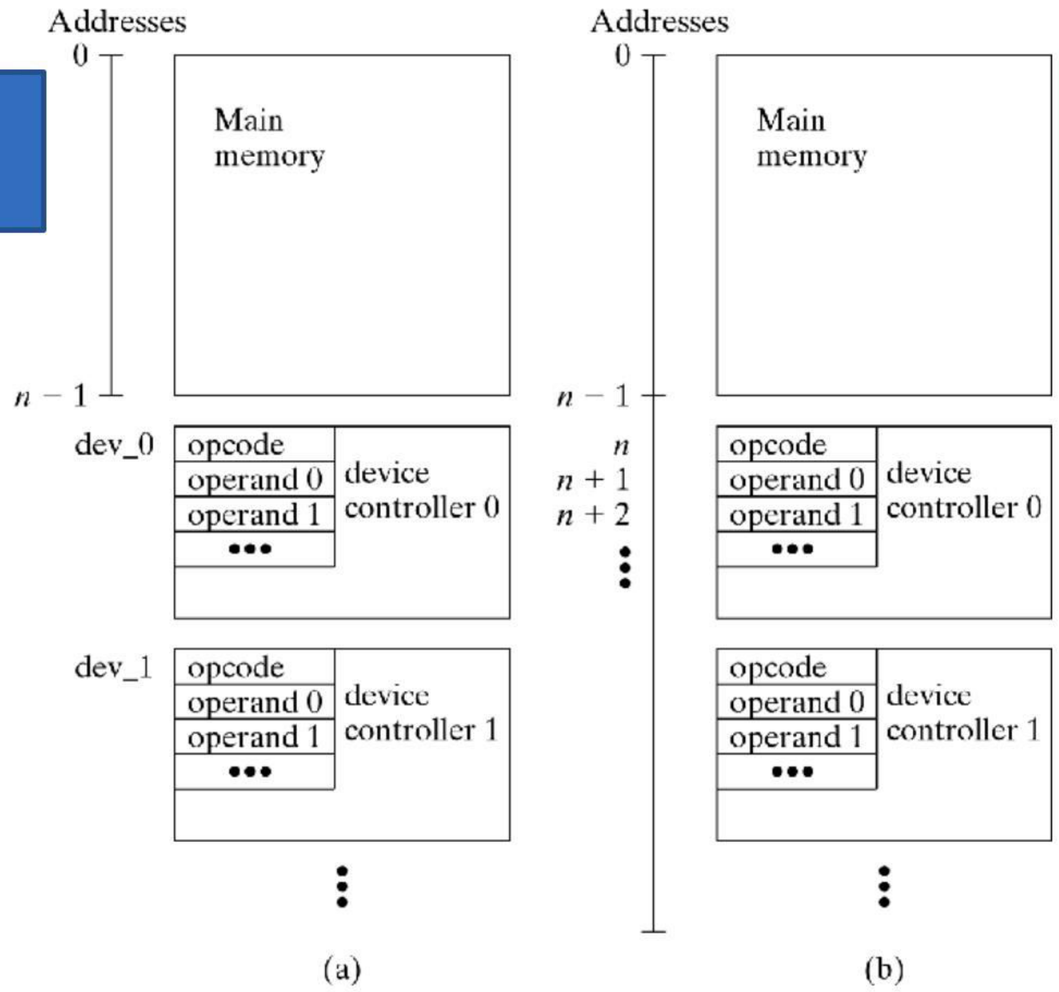
ממשק לשלוח או לקבל מידע על כבל

```
void
sendbyte(uint8_t byte)
{
  /* Wait until BSY bit is 1. */
  while ((inb (0x379) & 0x80) == 0)
    delay ();

  /* Put the byte we wish to send on pins D7-0. */
  outb (0x378, byte);

  /* Pulse STR (strobe) line to inform the printer
   * that a byte is available */
  uint8_t ctrlval = inb (0x37a);
  outb (0x37a, ctrlval | 0x01);
  delay ();
  outb (0x37a, ctrlval);
}
```

- כל עוד busy bitעמוס - המתן
- רשום ביט בעזרת d7עד d0
- שדר pulseעל 0x37a (strobe) שרשמת משהו חדש

מאינטרפייס לcontroller

Issue **explicit** I/O command



Main memory mapped to device controller

פעולות USB

- בעבר לכל התקן היה חיבור מיוחד
- USBהחליף את החיבורים השונים תחת סטנדרט אחיד
- זהו פרוטוקול לזיהוי ההתקן שמחובר
- 4סוגי תקשורת:
  - Interupt:מעט דאטה שנשלח ע״י ההתקן המחובר
  - Bulk: 64בתים עם תיקון שגיאות
  - Isochronous:סטרים של דאטה בלי תיקון שגיאות
  - Control:לקנפג ולשלוט בהתקן

Polling vs Interupts

- מי אחרי על העברת המידע?
- איך ה driverיודע שההתקן מוכן) ?ראינו גישה אחת לזה(

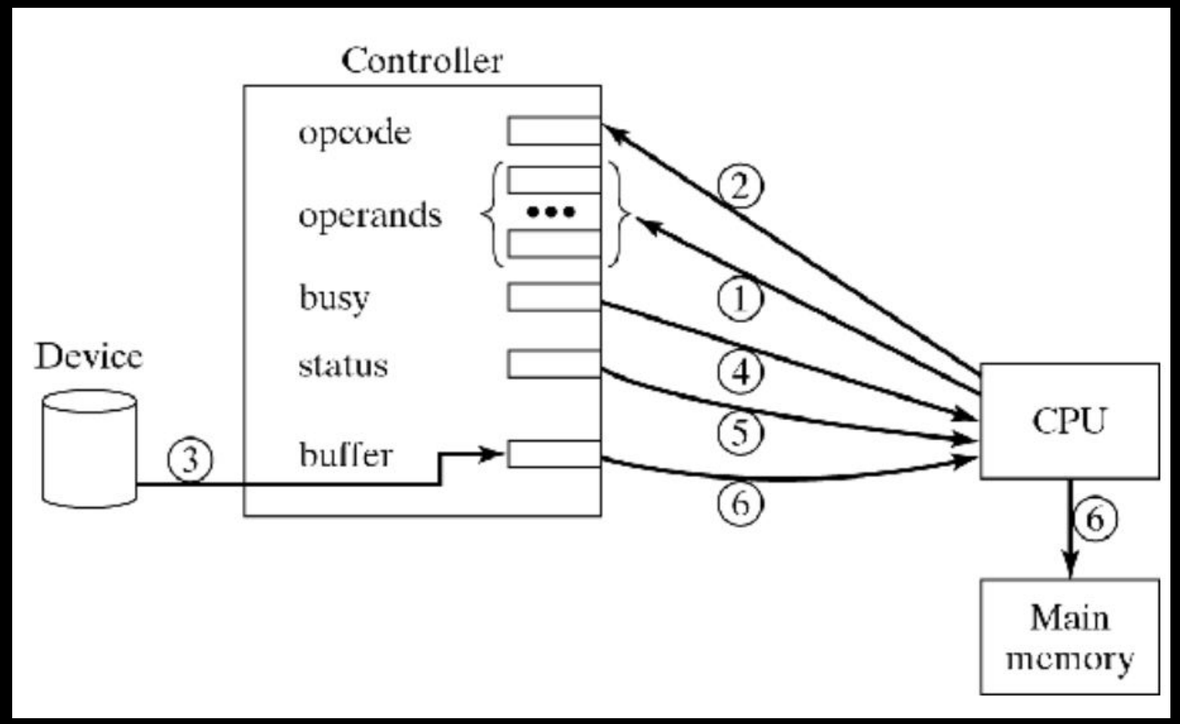
Polling vs Interupts

Polling

- ה CPUאחראי על
  - העברת כל תו ותו מ /אל הבאפר של ה controller
  - אחראי לזהות מתי פעולת I/Oהסתיימה
  - עושה זאת על ידי תשאול של הרגיסטרים המתאימים אצל ה) controllerכמו שראינו ב parallel ports)

Polling vs Interupts

Polling



1. מלא רגיסטרים מתאימים לבקשה בצד הקונטרולר.
2. שלח פקודה לקונטרולר
3. ההתקן מבצע את מה שצריך
4. המעבד בזמן הזה ממתין בלולאה ושואל האם הפעולה הסתיימה
5. בדוק שהכל תקין
6. שמור את התוצאה בזיכרון

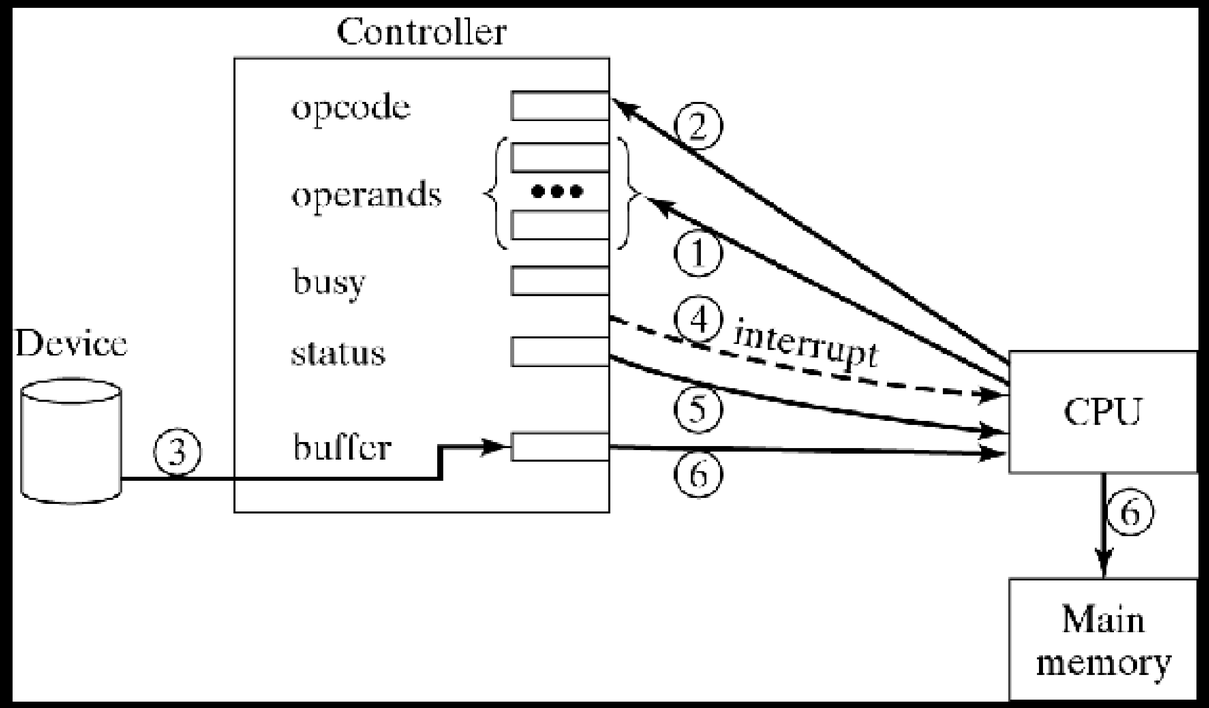
Polling vs Interupts

Interupts

- הגישה הנפוצה
- ה CPUעדיין אחראי על העברת המידע
- מקבל סיגנל אינטרפט כדי לדעת שהפעולת IOהסתיימה

Polling vs Interupts

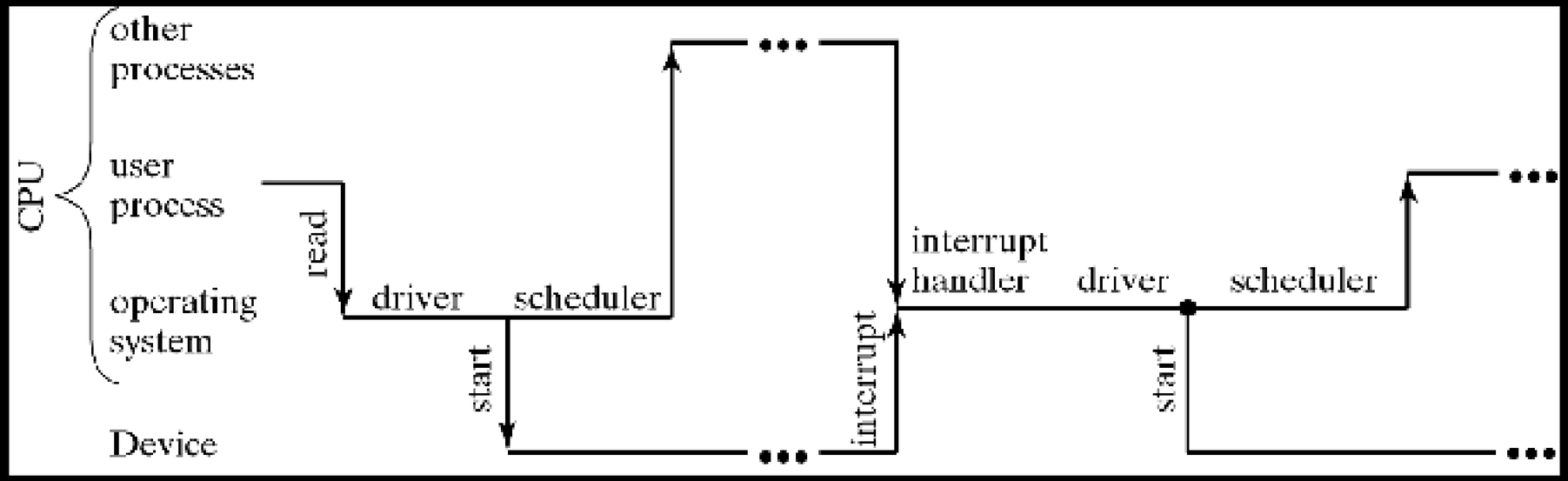
Interupts



1. מלא רגיסטרים מתאימים לבקשה בצד הקונטרולר.
2. שלח פקודה לקונטרולר
3. ההתקן מבצע את מה שצריך
4. המעבד ממשיך הלאה עד שמקבל סיגנל סיום מההתקן
5. בדוק שהכל תקין
6. שמור את התוצאה בזיכרון

Polling vs Interupts

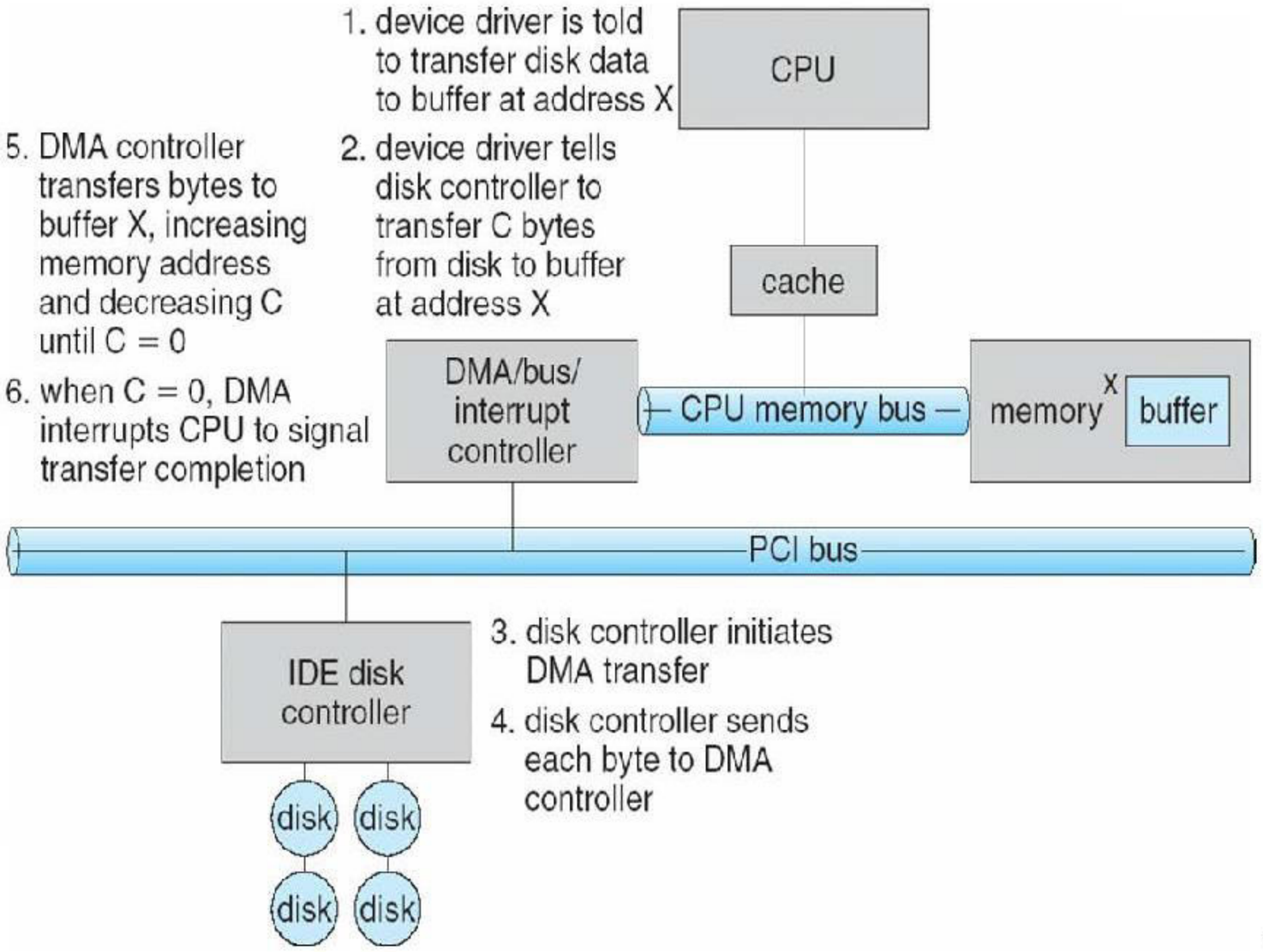
Interupts



DMA: Direct Memory Access

- רכיב המבצע בפועל את ההעברה של מידע בין הזיכרון לבין התקן

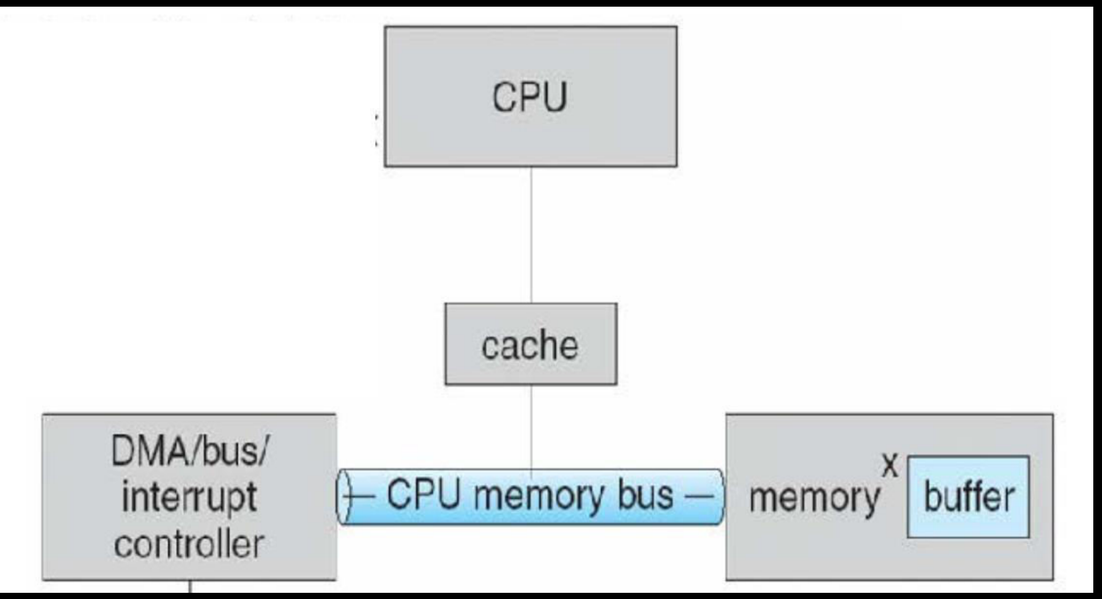
DMA: Direct Memory Access



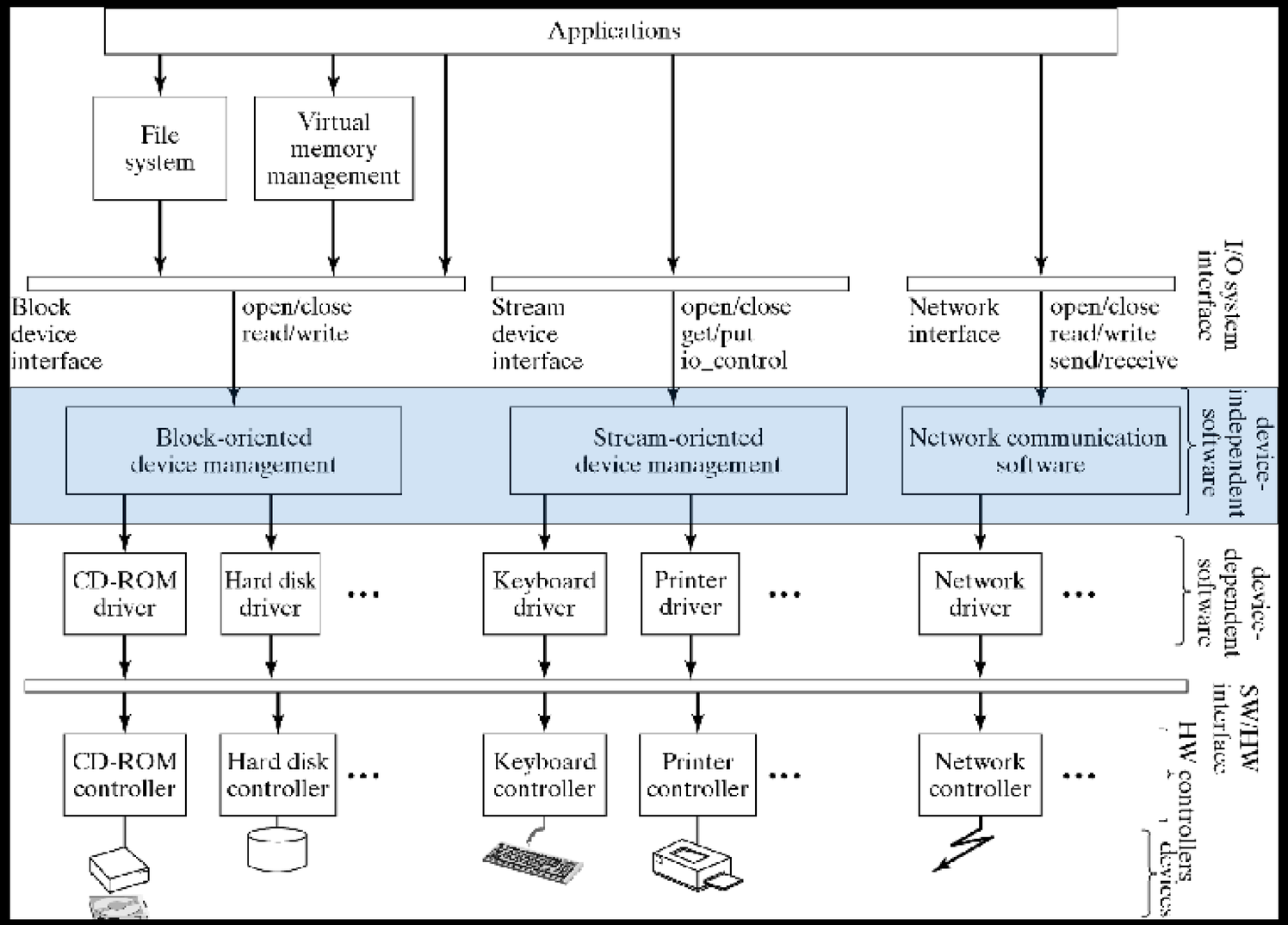
DMA: Direct Memory Access

Cycle stealing

- יש תחרות בין המעבד לבין ה DMAעל גישה לזיכרון
- אם ה CPUמנסה לגשת לזיכרון בזמן שה DMAמשתמש בזיכרון אז הCPU יצתטרך להמתין.
- זו לא בעיה בגלל שלרוב ה CPUינצל ויפנה ל.cache

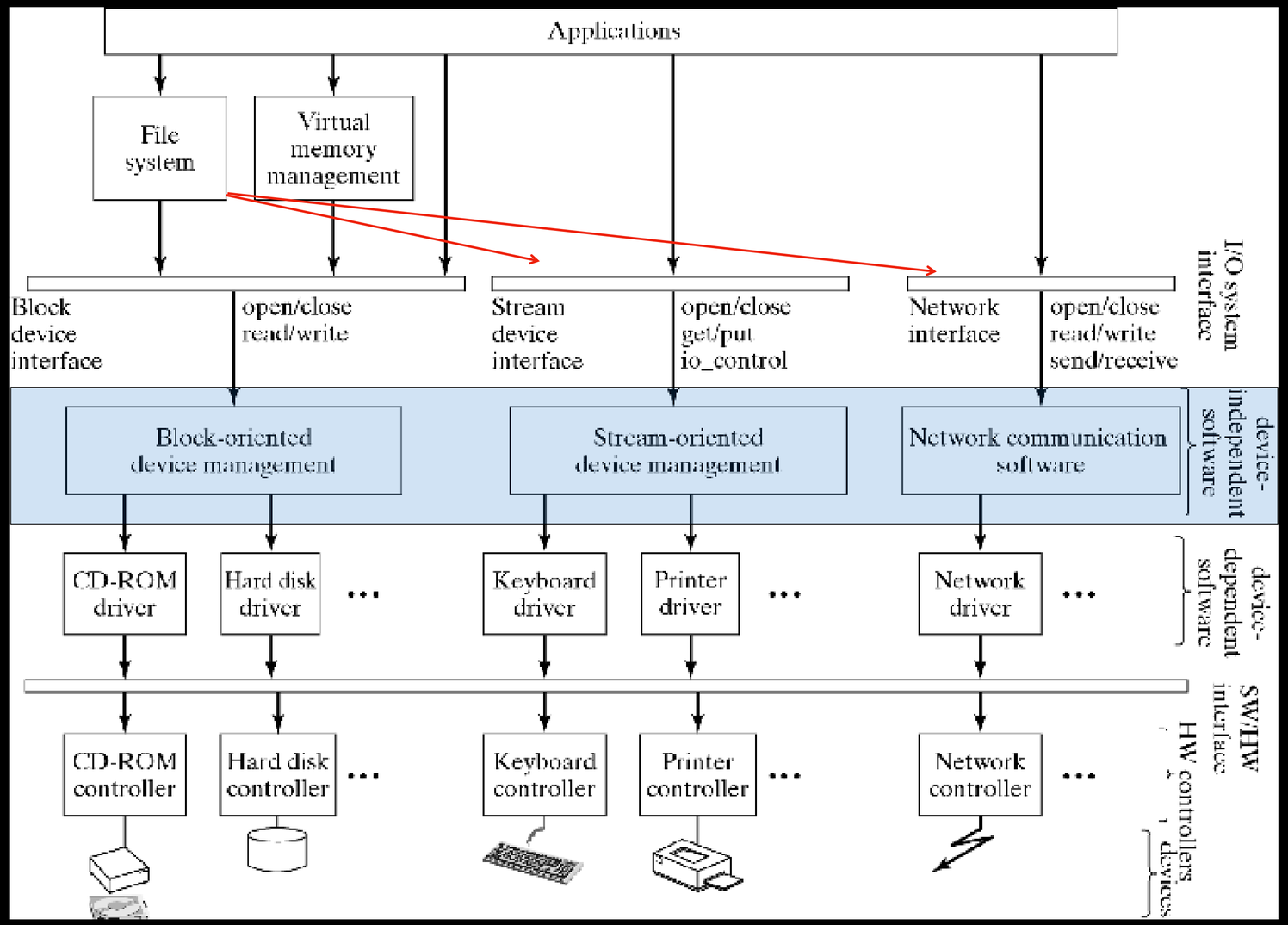


התייחסות להתקנים במערכת ההפעלה



- מערכת ההפעלה מחלקת דרייברים ל 3סוגים:
  - עובדים לפי בלוקים
  - עובדים לפי streams (בית אחרי בית)
  - עובדים לפי פקטות

מערכת הקבצים כאינטרפייס



- אינטרפייס אחיד הוא נוח
- מערכת הקבצים מהווה אינטרפייס לרוב ההתקנים שנשתמש בהם
- למשל ב sockets משתמשים בעזרת פעולות של כתיבה וקריאה מקובץ

ניהול התקנים

- למערכת ההפעלה מספר שיטות שונות להתנהל עם התקנים
- Buffering
- מיזעור שגיאות \טיפול בשגיאות
- תזמון גישה להתקן

ניהול התקנים

Buffering

- מאפשר פעילות אסנכרונית של producer ו consumer
- מאפשר חסכון בגישה למשאב יקר

ניהול התקנים

שגיאות

- מספר סוגי שגיאות :עקבית ,רגעית
- לרוב בעיה בהשגת משאב ,או תוכן שנפגע בזמן שינוע להתקן
  - ניתן לנסות שוב)לנסות לגשת שוב להתקן(
  - ניתן למנוע בעזרת קודים לתיקון שגיאות

ניהול התקנים

תזמון

- מערכת ההפעלה צריכה להבטיח גישה מסודרת להתקן
- יכולה לתזמן את הגישות להתקן כדי לנצל את מבנה ההתקן (למשל גישה לדיסק לפי המיקום שמבקשים לקרוא או לכתוב)