**מערכות הפעלה זמן אמת RTOS – Real Time Operating Systems**

**נועדו להיות תשתית ליישומי**

**Hard Real Time (HRT)**

**שפירושו "אי עמידה בלוחות זמנים, אפילו חד פעמית, פירושו אסון".**

**זאת בניגוד ל-**

**Soft Real Time**

**שפירושו "מספיק שאי עמידה בזמנים היא נדירה".**

**מערכות הפעלה כלליות (Windows, Linux)**

**הנחות:**

* **משרתות מספר גדול של יישומים במקביל**
* **משרתות מספר גדול של משתמשים במקביל**
* **תפקידן לשרת את כל המשתמשים והיישומים תוך זמן סביר**
* **מביאות בחשבון שיקולים סטטיסטיים**
* **יוצאות מתוך הנחה שעיכוב בביצוע קוד מועדף איננו אסון**

**יישומי זמן אמת:**

* **עשויות להיות יישומים רגישים שעיכוב פירושו אסון**
* **בדרך כלל המחשב רתום למטרה בודדת ולמשתמש בודד**
* **חשוב דטרמיניזם והמקרה הגרוע ביותר.**
* **לא ניתן ללמד מהניסיון ולתקן שגיאות בדיעבד.**
* **שאלה מרכזית בבדיקת היישום הינה "מתי אפשר לבדוק תוכנה במשך חודש ולהאמין שהוא ירוץ שנים".**

**מילת המפתח הוא דטרמיניזם שזה משהו כמו "לפעול תמיד תחת הוראות בטיחות".**

**דטרמיניזם – התנהגות הניתנת לחיזוי מדויק.**

**לאו דווקא היעיל ביותר.**

**היכן זה בא לידי ביטוי?**

* **מדיניות השיבוץ – נותן משקל מכריע לעדיפות. מדניות מסוג זה נקרא Event Driven.**
* **רמת הרשאות וקלט פלט לפעמים ניתנים ליישומים. הסיבה לכך הוא בדרך כלל חיבור לציוד לא סטנדרטי.**
* **חייבות לתמוך ב-multi-programming**
* **טיפול בפסיקות – מניעת עיכוב טיפול בפסיקות למינימום.**
* **שירותים צריכים להיות מתוכנתים דטרמיניסטית. האחד הראשונים זה ניהול זיכרון: בדרך כלל ביחידות זיכרון בגודל קבוע. אחד המקרים של עמידות לאורך זמן.**
* **זיכרון וירטואלי ודפדוף – תמיד קיים אבל רק לחלק מהמטרות שלמענם הוא קיים.**
* **אלגוריתמים מיוחדים בחסימות תהליכים: מדיניות השיבוץ המדגישה עדיפויות עושה בעיות.**