**חוטי משתמשUser End Threads**

**חוטים (Threads) הם תוכניות עצמאיות שרצות במסכרת תהליך אחד. התוכנית הראשית שלהם היא פרוצדורה (ולא קובץ בינארי).**

**חוטים מאפשרים multi-programming עם תקורה מופחתת.**

**רוב ההבדלים בין multi-processing ו-multi-threading נובעים מכך שה-multi-program הוא תהליך אחד (מרחב כתובות משותף, סביבה וטבלאות משותפות ...).**

**יש 2 קטגוריות של multi-threading:**

* **תמיכה בחוטים ברמת המערכת System Threads**
* **תמיכה בחוטים רק פנימית User-end Threads**

**ל-2 השיטות יתרונות וחסרונות.**

**ב-System Threads המערכת אחראית לתזמון, מדיניות השיבוץ והחלפת ההקשר.**

**ב- User-end Threads היישום דואג לכך, אם כי הוא צריך סיוע מהמערכת על מנת לחוש במעבר הזמן ולהעביר שליטה בהתאם.**

**דוגמא לתמיכה ב-User End Thread:**

**קיים ב-Unix סיגנל SIGALRM שניתן להזמין אותו שיתרחש חד פעמית או אינטרוולים:**

**useconds\_t ualarm(useconds\_t usecs, useconds\_t interval);**

**גורמת ל-SIGALRM חד פעמי usecs מילי שניות ובאינטרוולים של interval מילי שניות אחר כך:**

* **היישום תומך ב-thread יחיד.**
* **ה-thread נולד רץ**
* **היישום "מקצה" ל-thread אזור מחסנית משלו ע"י מערך לוקלי אוטומטי (למעשה, מקצה אותו ל-thread הראשי)**
* **ההקשר של 2 ה-thread-ים נשמר ע"י sigsetjmp**
* **כל SIGALRM מבצע החלפת הקשר ע"י siglongjmp.**