**מעבדה במערכות הפעלה: מפגש 12**

**נושא: פעילות אסינכרונית (פסיקות - interrupt) LINUX**

**מטרה:** להכיר את המושג 'פסיקה'(ופסיקות תוכנתיות בסביבת LINUX - SIGNALS)**.**

**הכרת מושג פסיקה ו-SIGNAL**

* תקראו בגוגל על פסיקות (תחפשו " פסיקות במחשב")
* ב-LINUX ניתן לשלוח הודעות מיוחדות (הנקראות **SIGNAL**) לתהליך, שמבחינת התהליך המקבל סימן להתחיל פעילות אסינכרונית (מן פסיקה).
  + יש כ-60 סוגי SIGNAL. ניתן להציג אותן ע''י פקודה:
    - Kill -l
    - למעשה פקודת KILL מייצרת פסיקה (SIGNAL) לתהליך המצוין בפקודה
* התהליך יכול להגדיר איזה פונקציה תתבצע עבור סוג מסוים של SIGNAL ע''י פקודה :
  + **signal(sig\_type, sig\_handler\_func)**
* הערה: לא עובדים עם חוטים (רק תהליכים)

**משימה 1: הדגמת פעולה אסינכרונית**

**התוכנית: signal\_ex1.c**

1. תריצו את התוכנית תנסו לסיים אותה עם

CNTRL-C - מה אתם רואים ?

1. תנסו (מחלון אחר ) את הפקודה הבאה:

Kill -USR1 PID

1. תנסו (מחלון אחר) את הפקודה: kill 9 PID
2. תבדקו התנהגות התוכנית אם שמים בהערה את פקודות ה-SIGNAL

להגשה:

* + תתארו ותסבירו את מה שראיתם ב 1-4.

**משימה 2:**

**התוכנית: signal\_ex2.c**

תסתכלו ותבינו את התוכנית.

* 1. פקודה kill(pid, SIGHUP) שולחת פסיקה (SIGNAL) לתהליךpid
  2. מה לדעתכם התוכנית צריכה להדפיס ?
  + תריצו את התוכנית מספר פעמים: האם אתם רואים הבדלים בין ההרצות ?
  + תנסו להבין את מקור הבעיה ותציעו פתרון (ותבדקו שבאמת פתר)

**להגשה:**

* + שתי הדפסות שונות: איזה מהן תקינה
  + הסבר קצר על מקור הבעיה
  + פתרון לבעיה

**משימה 3: לסנכרן את ההדפסות בעזרת מנגנון SIGNAL**

**התוכנית: fork\_char\_by\_char.c**

* + מכירים כבר את התוכנית: תריצו ותראו בעיה בהדפסות
  + עליכם לפתור את הבעיה (לסנכרן גישה למסך ) ע''י SIGNALS
  + הרעיון הכללי: שני תהליכים, כל אחד בתורו מדפיס שורה אחת ונותן זכות לתהליך השני להדפיס שורה הבאה שלו (כמובן צריך להבטיח שלא פוני ם ביחד באותו זמן).

**להגשה:**

* + תיאור כללי קצר של הפתרון המוצע
  + קוד מתוקן + הדפסה