**סיגנלים – מנגנון החריגות של שפת C**

**ב-Unix תהליך יכול לגרום לחריגה בתהליך אחר.**

**ב-Windows רק לעצמו.**

**ב-UNIX:**

**סיגנל – סוג של חריגה.**

**לכל סיגנל יש "אפקט ברירת מחדל":**

* **חיסול התהליך**
* **השהיית התהליך**
* **חידוש ריצה של תהליך מושהה**
* **התעלמות**

**לרוב הסיגנלים תהליך לשנות את התגובה שלו:**

* **להתעלם מהסיגנל.**
* **למנות רוטינה שתהיה "רוטינת טיפול בסיגנל" (לפעמים מחייב נוהל מיוחד).**
* **לחזור לברירת המחדל.**

**לסיגנלים יש מספרים, רובם סטנדרטיים.**

**רוטינת טיפול בסיגנל מקבלת כפרמטר את מספר הסיגנל.**

**3 קריאות מערכת חשובות:**

**Signal – משנה את תגובת התהליך**

**Kill – ששולח סיגנל**

**Raise – תהליך שולח סיגנל לעצמו**

**סיגנלים סטנדרטיים:**

**לא ניתנים להתגוננות:**

* **SIGKILL**
* **SIGSTOP**

**סיגנלים משהים:**

* **SIGSTOP**
* **SIGTSTP**

**סיגנלים של המקלדת:**

**Ctrl-C – SIGINT**

**Ctrl-\ - SIGQUIT**

**Ctrl-Z – SIGTSTP**

**שינוי התגובה לסיגנל X:**

**signal(X, SIG\_IGN);**

**signal(x, shandler);**

**signal(x, SIG\_DFL);**

**אם ממנים ב-signal "רוטינת טיפול בסיגנל" חזרה של הרוטינה פשוט מחדשת את התהליך מאותו מקום (למעט חריגות מערכת).**

**לפחות היסטורית, השפעה של הרוטינה signal היא לפעמים חד פעמית.**

**אם תהליך לא מעוניין שתהליכי הבן שלו יהפכו לתהליכי zombie עם סיומם עליו לבצע:**

**signal(SIGCHLD,SIG\_IGN);**

**קבוצות תהליכים**

**כל תהליך שייך לקבוצה.**

**ברירת המחדל – הקבוצה של תהליך האב.**

**תהליך יכול לפרוש לקבוצה של עצמו, או להצטרף לקבוצה אחרת.**

**השימוש העיקרי של קבוצות תהליכים: ניתן לשלוח סיגנל לכל חברי הקבוצה בבת אחת.**

**Ctrl-C, Ctrl-\, Ctrl-Z שולחים סיגנל לקבוצה של תהליך –prompt.**

**רוטינות:**

 **int setpgid(pid\_t pid, pid\_t pgid);**

 **pid\_t getpgid(pid\_t pid);**

 **int setpgrp(void);**

 **pid\_t getpgrp(void);**

**מימוש try,catch וחריגות מערכת**

* **אי אפשר לעשות SIG\_IGN לחריגת מערכת.**
* **ניתן לעשות הגדרת פונקציה מטפלת לחריגת מערכת, אבל זה לא מספיק, יווצר לולאה אינסופית או בעיה אחרת.**
* **ניתן לממש try,catch ע"י שילוב של signal והסתעפויות גלובליות.**