

## תרגיל בית 1 – חלק תאורטי

### שאלה 1:

בקונבציית GCC לקריאה לפונקציה שומרים בסדר הפוך במחסנית את הפרמטרים המועברים לפונקציה הנקראת כדי לאפשר גישה פשוטה ע"י שמירת הפרמטר הראשון במקום קבוע אחרי return address מאחר ולא ידוע כמה פרמטרים יש לה.

### שאלה 2:

בזמן ביצוע קריאת מערכת, כאשר עוברים מה-user mode ל-kernel mode לא ניתן להעביר פרמטרים דרך המחסנית מאחר וכל מצב משתמש במחסנית שונה.

### שאלה 3:

הפלטרים האפשריים למסך של קטע הקוד הם:

- במידה ו-fork נכשלה תקרא הפונקציה exit והמשך הקוד לא ייקרא, נקבל:

**פלט ריק**

- במידה ו-fork הצליחה ו-execv בתהליך הבן הצליחה, הקוד של הבן יוחלף בפקודת sleep והמשך הקוד המקורי לא ייקרא, לכן רק ההדפסה של האב תעבוד, נקבל:

**8**

- במידה ו-fork הצליחה ו-execv בתהליך הבן נכשלה, הקוד של הבן לא יוחלף ולכן ההדפסה שלו תעבוד גם (אחרי ההדפסה של האב), נקבל:

**8**

**13**

### שאלה 4:

הפלטרים האפשריים למסך של קטע הקוד הם:

- במידה ו-fork נכשלה ניכנס ל-if, הקריאה wait תחזיר (-1) (כי אין בן), לכן value יישאר 0, נוסיף לו 3 ונדפיס. אחרי עדכון הערך עוד פעם אין שום הדפסה לכן סהכ נקבל:

**3**

- במידה ו-fork הצליחה תהליך האב יכנס ל-if, ימתין לתהליך הבן (בגלל wait), בזמן הזה הבן ידפיס 0 (כי לא נכנס ל-if), ויחזיר לאב 4. האב יוסיף 3 וידען את value ל-7 וידפיס. סהכ נקבל:

**0**

**7**