**תרגיל יבש 2 – מערכות הפעלה**

**מגישים: להב פרידלנדר ת.ז 209403781**

**Email: lahavfri@campus.technion.ac.il**

**אלון פליסקוב ת.ז 315468116**

**Email:** **alonpliscov@campus.technion.ac.il**

**חלק 1:**

1. פקודת yes מקבלת string כארגומנט, ומדפיסה אותו (עם ירידת שורה) בלולאה אינסופית.

אם לא מצוין ארגומנט, אז yes מדפיסה את המחרוזת 'y'.

1. השתמשנו בפקודת yes עם מחרוזת ריקה, כדי לרוקן את הקובץ oldconfig עד סופו (הגודל המקסימלי של הקובץ). הצינור בפקודה מעביר את המחרוזות הריקות לקובץ oldconfig, ומוחק את השורות הקיימות. כך מתקבל קובץ oldconfig שמלא בשורות ריקות (כלומר מלא ב-'\n'). עבור המקרה שלנו, אנו יוצרים גרעין חדש, ולכן פעולה זו מונעת שימוש בפעולות הגרעין הישן לצורך התרגיל (ע"י הרצת make oldconfig).
2. GRUB\_TIMEOUT מונע מ-GRUB להריץ את מערכת ההפעלה הדיפולטיבית ישירות, בכך שהוא ממתין GRUB\_TIMEOUT שניות לפני שהוא מריץ את מערכת ההפעלה הדיפולטיבית. כך המשתמש מקבל מספר שניות להריץ מערכת הפעלה אחרת לבחירתו. היתרון בשינוי זה הוא האופציה לבחור מערכות הפעלה שונות, בעוד החיסרון הוא עיכוב במקרה של בחירה קבועה במערכת הדיפולטיבית, וסכנה שהמשתמש יבחר במערכת הפעלה שאינה מתאימה לו.
3. מכיוון שזהו קוד הגרעין, לכן משתמשים בפונקציית הגרעין do\_execve. שימוש בפונקציית גרעין גם מוודא שהמשתמש לא חורג מההרשאות שלו (כי execve חשופה למשתמש), וגם מונע טעויות של המשתמש בקוד קריטי לגרעין (כמו הפעולה הזו, שמריצה את התהליך המרכזי init). מכיוון שזהו קוד הגרעין, לא קיימת הפונקציה execve, לכן לא נוכל להריץ את הקוד לאחר החלפה ל-execve.



1. חתימת הפונקציה:

הפונקציה מקבלת את המספר *number* ואת הפרמטרים הנחוצים לקריאת המערכת (מספר הארגומנטים ותפקידם משתנה לכל קריאת מערכת), ומבצעת את קריאת המערכת המשוייכת למספר *number*.

הפונקציה ממומשת בספריה <unistd.h>.

1. syscall(57) קוראת לקריאת המערכת fork. המשתנה long r מקבל את ערך החזרה של קריאת המערכת. בתהליך האב, r שומר את ה-PID של תהליך הבן, במידה והקריאה לא נכשלה. בתהליך הבן, אם הקריאה הצליחה, r שווה ל-0.  
   במקרה של הצלחה, תהליך הבן ותהליך האב ימשיכו לבצע במקביל את פקודות הקוד, לכן אין סדר הדפסה ביניהם. בכל מקרה, תהליך האב ידפיס :

sys\_hello returned {child\_pid}

תהליך הבן ידפיס:

sys\_hello returned 0

ניתן להחליף את הקריאה ל-syscall בקריאה לקריאת המערכת fork(), כדי לקבל קוד ברור יותר.