

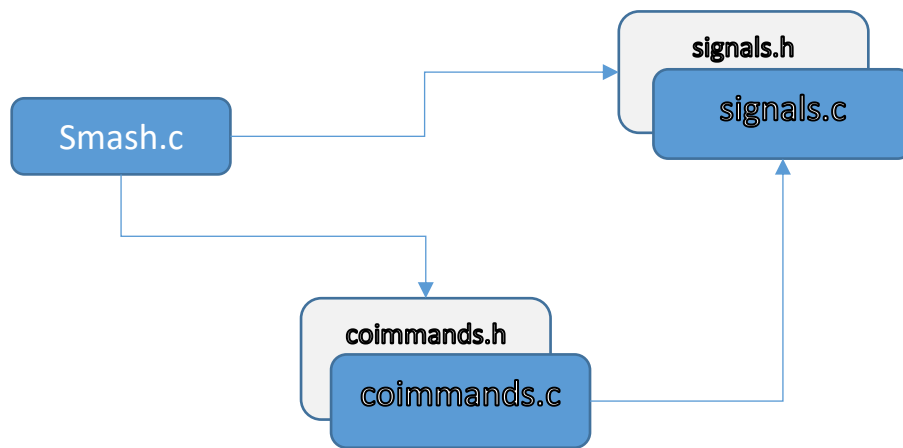
**מבנה מערכות הפעלה**

**שגיא גרש 204095313**

**יהב וינוקור 307853010**

**תאריך 30.11.19**

להלן דיאגרמה אשר מתארת את רכיבי התוכנית ומעבר הנתונים בניהם:



### • :Smash.c

- מימוש ה-main, אשר מקבל את קלט המשתמש ושולח אותו לפונקציות הרלוונטיות בתוכנית.
- אתחול משתנים- מבנה נתונים המכיל את ההיסטוריה, ומבנה נתונים המכיל את ה"עבודות".
- שיוך פונקציות טיפול בסיגנלים לסיגנלים שלהן.

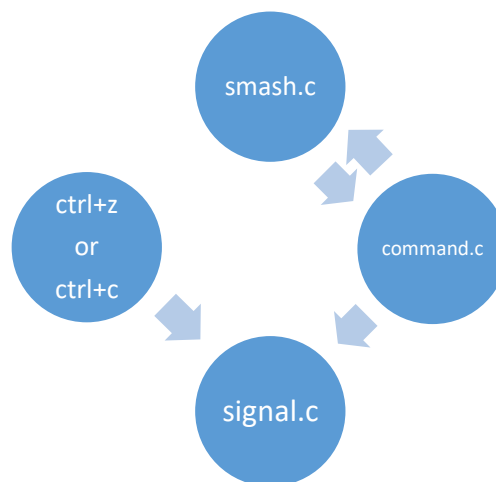
### • :Commands.c

- מימוש כלל הפקודות הפנימיות של התוכנית.
- מימוש הרצת תוכניות חיצונית.
- מימוש הרצה ברקע, ניהול ה"עבודות", עדכון ההיסטוריה.

### • :Signals.c

- מימוש כלל פונקציות הטיפול באותות.

המידע מועבר כך: העברת הפקודה מ-smash.c ל-command.c, ביצוע הפקודה ועדכון כלל מבני הנתונים, העברת signals בהתאם לצורך דרך signals.c לתהליכים השונים וחזרה דרך command.c ל-smash.c לקבלת פקודת קלט נוספת. כמו כן קבלת אות בצורת ctrl c או ctrl z יועברו לsignals.c אשר יטפל באות.



## מבנה הנתונים בו השתמשנו

מבנה הנתונים העיקרי בו השתמשנו היה מערך, מאחר והמערך אפשר לנו לגשת אל האיברים שבו, על פי אינדקס (כפי שמתייחסים לjob), בסיבוכיות של  $O(1)$ . וכן איפשר לנו מחיקה נוחה והשאת האיברים במקומם כפי שמתבצע בjobs, בshell.

כל איבר במערך הוא מצביע ל struct, אשר מייצג job יחיד. הוא שומר את כל השדות שלו -

- המספר הסידורי (אשר הוא גם המיקום המערך)
- השם של התהליך
- ה-process ID של התהליך
- הזמן שבו התהליך התחיל לפעול
- הסטטוס של התהליך

בנוסף, בשביל מבנה הנתונים של ההיסטוריה השתמשנו במערך בגודל 50, אשר מימשנו לו הוספה והדפסה באופן ציקלי.

פשטות המימוש והפעולות הן אלו שהנחנו אותנו בבחירת מבנה הנתונים, בנוסף לכך שגודלו המקסימלי של מבנה הנתונים נתון מראש ולכן אין צורך בדינאמיות גודלו של מבנה הנתונים.

הפעולות אשר הוספנו לצורך ממשק עם מבנה הנתונים של ה jobs הינן:

Add job - פעולה אשר מוסיפה תהליך למערך ה jobs. נקרא לה כאשר נפעיל תהליך ונעדכן אותה כל פעם שנקרא לפעולה.

rem job - פעולה אשר מסירה תהליך ממערך ה jobs. כאשר הוא נגמר או נעשית פעולה אשר גורמת להוצאתו.

UpdateJobs - עדכון מערך ה jobs, לפי שינוי סטטוס תהליכים. הפונקציה תיקרא בכל פעם שניכנס לפונקציה אשר מוציאה לפועל פקודה - כתיבה ולחיצת מקש אנטר על ידי המשתמש (execmd) וכן בעקבות סיגנלים. אלה גורמים בפועל לתהליך לעבור ממצב user mode ל kernel mode. כך תשמור על המערך מעודכן להדפסה נכונה.

בנוסף כדי לטפל בפעולות הקשורות להיסטוריה הוספנו את:

UpdateCmdHistory - פונקציה אשר דואגת לעדכן את מערך ההיסטוריה באופן ציקלי.