**קובץ אפיון- רטוב 1**

דיאגרמת בלוקים:

Smash

Signal

command

התוכנית הראשית Smash מעבירה את הפקודות מהמשתמש לקובץ command שמחזיר מידע לקובץ הראשי. בנוסף command מקבל סיגנלים בעזרת קובץ smash שעושה שימוש בפונקציות בקובץ signal.

**Smash**: הקובץ הראשי, מקבל את המידע מהמשתמש ומעביר אותו לטיפול בפונקציות המתאימות ומעביר סיגנלים לטיפול בקובץ signals.c. בקובץ מאותחלים משתנים גלובלים לשימוש שאר הקבצים.

**Command**: טיפול בפקודות המובנות והחיצוניות לפי ההנחיות שניתנו.

פונקציות לסיווג סוג הפקודה: הפונקציות מחזירות ערכים להמשך טיפול ב Smash לפי אופי הפקודה שהוזנה- חיצונית, פנימית או מסובכת.

פונקציה לטיפול בפקודה מסובכת: במקרה זה לא נעשה דבר כפי שהוגדר בהנחיות.

פונקציה לטיפול בפקודה מובנית: הפקודות המובנות מומשו באופן מלא בתוכנית והן מעדכנות את מבני הנתונים ומדפיסות מידע למסך לפי הצורך.

פונקציה לטיפול בפקודה חיצונית: לפקודות חיצוניות נוצר תהליך בן שמריץ את הפקודה, שנכתבה על ידי המשתמש או פקודות מערכת בעזרת הSHELL החיצוני שבו רצה התוכנית. במידה והפקודה היא פקודה שרצה ברקע יש עדכון של מבנה הנתונים המתאים.

**Signals**: מכתיב טיפול בסיגנלים וגם טיפול ספציפי ב- SIGINT SIGTSTP .

פונקציית שליחה סיגנל מקבלת את מספר הסיגנל לשלוח ואת מספר התהליך אליו נשלח הסיגנל ובודקת את הערכים שנשלחו ושולח את הסיגנל בעזרת KILL של מערכת ההפעלה. בנוסף, לפי חיווי בוליאני הפונקציה מדפיסה את המידע לגבי הסיגנל שנשלח.

טיפול ב SIGINT שמקורו ב ctrl-c שמגיע מהמשתמש תופס את הטיפול בסיגנל לsmash ושולח סיגנל לסיום התהליך שרץ ב-fg תחילה יש זיהוי של התהליך הרצוי ולאחר מכן שליחה של סיגנל תוך הדפסה של הפעולה שנעשתה.

טיפול בSIGTSTP שמקורו ב ctrl-z משהה את התהליך שרץ ב fg, מוודא שהפקודה נעצרה, מכניס לרשימת הפקודות שרצות ברקע ומעדכן שהפקודה נעצרה.

מבני נתונים:

history: להחזקת היסטוריית הפקודות שהוזנו על ידי המשתמש השתמשנו בתור מתוך ה-STL של C++. אפיון פעולת ההיסטוריה תואמת הגרת מבנה נתונים שמקיים FIFO ולכן השתמשנו בתור. כאשר מספר הפקודות בתור מגיע למקסימלי נוציא את הפקודות הישנות ונכניס חדשות.

Jobs: שמירת פקודות שרצות ברקע נעשה בעזרת וקטור מתוך ה-STL של C++. נעזרנו במבנה הנתונים הדינאמי על מנת להוסיף פקודות חדשות ולהחזיק מידע על הפקודות הקיימות שרצות ברקע. הוקטור החזיק מבנה struct job\_command שהכיל מידע על הפקודה שהוזנה- שם, מספר סידורי, מספר של התהליך, זמן הכנסה לוקטור, האם מושהה או לא. כתבנו מספר פונקציות שמאפשרות גישה למידע