

דו"ח מכין, מעבדה מס' 2 – Stack, Routine, Macro

מגישים: אסף קמבר (313390429) ואביב פרימור (208488973)

1. הסבר מהי מחסנית, את הצורך בה ואופן שימושה.

תשובה: מחסנית הינה אזור במרחב זיכרון ה-RAM שמיועדת לצרכים כמו שמירת כתובת חזרה מרוטינה ופסיקה או לאגירת מידע לצורך חישוב בנוסף לרגיסטרים של ה-CPU. על מנת להשתמש במחסנית עלינו להיעזר ברגיסטר Stack Pointer הנמצא ב-CPU המשמש כמצביע לראש המחסנית, תחילה נאתחל את ערכו לכתובת של תחילת המחסנית ובכל פעולה נוספת שדורשת מחסנית, ערכו ישתנה בהתאם. רגיסטר ה-SP מתקדם או הולך אחורה אוטומטית ע"י ה-CPU עבור פקודה המשתמשת במחסנית.

2. הסבר מהי רוטינה – את הצורך בה, אופן שימושה וכיצד היא משפיעה על המחסנית

תשובה: רוטינה מהווה דרך לביצוע פעולה שאותה נרצה לבצע מספר פעמים במהלך התוכנית, היא מוגדרת בקטע קוד שנוכל לקרוא לו במהלך הריצה בלי הצורך לרשום את אותן שורות הקוד פעם נוספת. בעזרת שימוש ברוטינות נוכל לחסוך בזיכרון הבקר, נוחות של תכנות הקוד, debug ותוכנית יותר מובנית. קטע הקוד מתחיל בכתובת שהתווית שלו היא שם הפונקציה והפקודה האחרונה היא פקודת return. הגדרת הרוטינה תעשה ע"י כתיבת קטע קוד המתחיל בתווית המשמשת יעד לפקודת call ומסתיימת בפקודת ret. בעת הפעלת הרוטינה – ה-CPU מבצע פקודת call רגיסטר SP שערכו קטן ב-2 לצורך אחסון כתובת לחזרה. לאחר מכן רגיסטר PC נטען בכתובת ההתחלתית של הרוטינה לצורך ביצוע. קטע הקוד של הרוטינה מתבצע עד להגעת לפקודת ret ולאחר הגעה לשורה זו, רגיסטר PC נטען בערך כתובת חזרה הנמצאת במחסנית בכתובת שערכה הוא ערך הרגיסטר SP, וכך רגיסטר PC נטען בכתובת הפקודה הבאה אחרי פקודת call.

3. הסבר מה היא פונקציית MACRO – את הצורך בה ואופן שימוש, רשום טבלת יתרונות וחסרונות בין פונקציית MACRO לבין רוטינה.

תשובה: פונקציית MACRO היא פונקציה המאפשרת למתכנת להגדיר סדרת פקודות הניתנת למחשב כפקודה אחת. שימוש בפונקציית המקרו מאפשר חיסכון בזמן ובמאמץ בשונה משימוש ברוטינות אשר כל קריאה וחזקה לקטע קוד שתגרום להשהייה בזמן ריצת התכנית. על מנת להשתמש בפונקציית המקרו, נגדיר קטע קוד באזור ה-pre-compile של הקוד ולאחר מכן נוכל לקרוא לקטע הקוד בעזרת ארגומנטים משתנים. כשנקרא לפונקציית המקרו, תוכנת סביבת הפיתוח תציב את ערכי המשתנים בגוף ה-MACRO ולאחר מכן מעתיקה את גוף ה-MACRO במקום שורת ההפעלה.

Macro	רוטינה	
1. נמנעת מהוספת מחזורי שעון לזמן ריצה. 2. קוד קריא וקל לתחזוקה.	1. קוד קריא וקל לתחזוקה הניתן לdebug. 2. אין שכפול מיותר של הקוד.	יתרונות
1. מגדיל את נפח הזיכרון. 2. לא ניתן לבצע debug לקטע הקוד.	1. הוספת מחזורי שעון בזמן הכניסה והיציאה מהרוטינה.	חסרונות

חלק מעשי

גודל התכנית: 18 bytes
זמן ריצה: 144.00 μsec