ממן 14 הערות – נתנאל חדד

ציון: 96

**תיעוד**

פרוייקט בסדר גודל הזה אמור לכלול תיעוד בשלוש רמות:

ברמת המאקרו – מה עושה הקוד וכיצד. הסבר כללי על משימות הבסיס עימן מתמודד הקוד והאופן בו הקוד שלכם עושה זאת. זה המקום גם להסביר את הלוגיקה מאחורי החלוקה לקבצים, ולהסביר מה נמצא בכל קובץ. זה גם המקום להתייחס לשיקולים שהנחו אתכם בבחירת דרך מימוש מסויימת, בפרט אם היו התלבטויות שליוו את התהליך.

ברמת המיקרו – הערות נקודתיות בתוך הקוד במקומות בהן זה נדרש. ההערות האלה נועדו להסב את תשומת הלב לדברים ש"בזמן אמת" או בחלוף הזמן עלולים להיות פחות מובנים. כמו כן, קריאה דרך ההערות האלה ברצף אמורה, פחות או יותר, לתת תמונה על מבנה הפונקציה ו/או כיצד היא עובדת.

ברמת הביניים – בכל קובץ כדאי לכתוב מה יש בו, ועבור כל פונקציה לתאר באופן כללי מה היא עושה וכיצד.

עד עכשיו, בכל הפרוייקטים שבדקתי באה כאן הערה שהתייחסה לאחת מרמות התיעוד שלא נעשתה כמצופה. במקרה שלך – שאפו! יופי של תיעוד בכל 3 הרמות!

**חלוקה לקבצים ולפונקציות**

יש חלוקה טובה לפונקציות משנה. יש חלוקה טובה גם לקבצים, וה- makefile מצליח לחבר ביניהם. ההסברים עבור כל קובץ מבהירים מה יש בו, ואכן זה נעשה בחלוקה הגיונית. קובץ MAIN קומפקטי המבהיר את האופן בו הקוד עובד – פתיחת קובץ, פרישת מקרואים, מעבר ראשון שמתוכו מתבצעת קריאה למעבר שני – עם הערות שמבהירות את המבנה. נכנס אצלי לשקף לטובת הסמסטר הבא... 😊

**קומפילציה** – קובץ makefile עושה את העבודה על פי הדרישה. הקימפול עצמו עובר ללא שגיאות וללא הערות עם כל הדגלים הדרושים.

**דוגמאות ריצה** – צירפת קבצי קלט ופלט, והסברת היטב מה מכילה כל תיקיה. נדרשתם לצרף גם צילומי מסך, אם כי זה קריטי יותר בקוד שיש בו בעיות ששווה להתייחס אליהן. אצלך זה עובד היטב, כך שמעבר להיבט הפורמאלי החוסר בצילומי מסך לא משמעותי.

**תקינות ריצה**

**קריאת שם הקובץ (או הקבצים) מהמשתמש**

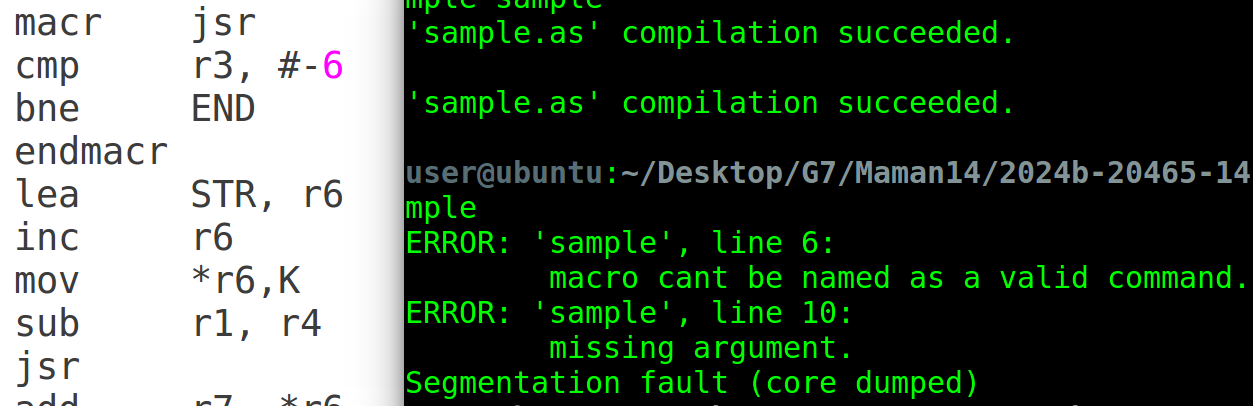
קריאת שם הקובץ, פתיחתו, יצירת קבצי הפלט – כל אלה עובדים מצויין כנדרש כולל עניין הסיומות. מטפל היטב גם במקרה של הרצה עם מספר קבצים.

**שלב הקדם מהדר**

עובד מצויין.

כאשר יש שגיאה בשלב פרישת המאקרו – למשל, כאשר שם המאקרו הוא שם של הוראה קיימת בשפה – הקוד שלך מייצר את הודעת השגיאה המתאימה וזה סבבה לגמרי.

על פי דרישות הפרוייקט, במצב כזה נקבע כי "אי אפשר לעבור לשלבים הבאים: יש לעצור להודיע על השגיאות ולעבור לקובץ המקור הבא" (עמ' 26). אצלך הוא כן ממשיך הלאה, ובדוגמה המצורפת אפילו ייצר הודעת שגיאה עבור שורה 10 שבה הוא טוען שחסר אופרנד – זו השורה שבה מופיע שם המאקרו שהוא גם שם ההוראה jsr (זו שורה 10 בקובץ ה-am.). זה מראה שהוא מזהה היטב את השגיאה עבור jsr, אבל על פי דרישות הפרוייקט לא היינו אמורים להגיע למעבר הראשון (שם מיוצרת הודעת השגיאה עבור שורה 10) והתוכנית הייתה אמורה להפסיק לרוץ לאחר הניסיון לפרוש את המאקרו הפגום.

אני מודע לקושי לעקוב אחרי כל דקדוקי הפרוייקט, מה גם שבמקום אחר בחוברת ניתן להבין אחרת לגבי מקרה של שגיאה בשלב פרישת המאקרו. מה שכן, הדבר גם גורם ל- segmentation fault, דבר שלא אמור לקרות בפרוייקט שעובד תקין.



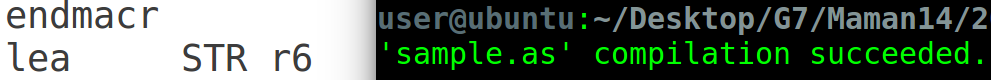
**מעבר ראשון ושני**

בסה"כ, עובד מצויין. קבצי הפלט מיוצרים כמצופה.

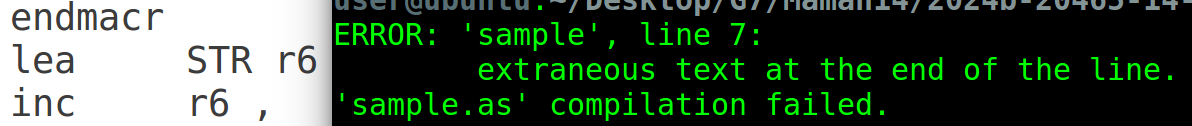
**טיפול בשגיאות** – עובד היטב בדוגמאות היפות שצירפת, וגם ברוב מה שאני בדקתי. הפיספוס היחיד שזיהיתי היה בנוגע לפסיקים – הוא לא מוטרד מעודף שלהם, וגם לא מהיעדרם. רק אם הפסיק מופרד ע"י רווח ומופיע בסוף השורה אז הוא מזוהה כ"טקסט עודף" (רווח בין שני הפסיקים ל- STR, למשל, לא "עזר" לקוד שלך לזהות שם בעיה).













השימוש בפסיקים הוא חלק מהתחביר של השפה, וגם בלי הודעת שגיאה קונקרטית ראוי היה שהוא יודיע על איזושהי שגיאה.

אבל הראית יכולת להתמודד עם שגיאות ולספק הודעות שגיאה אינפורמטיביות כך שזה נלקח בפרופורציות המתאימות...

**סיכום**

בסה"כ, יופי של עבודה – נהניתי לבדוק!

בהצלחה רבה בהמשך! 😊