מטלה 2

בתרגיל זה נעשה מספר ניסויים שמטרתם להכיר את הקומפיילר החכם של C++ ואת הדברים שהוא עושה עבורנו מאחרי הקלעים.

שימו לב: התשובות לרוב השאלות לא נלמדו בהרצאה. נסו לענות עליהן כמיטב יכולתכם ע"י (א) הגיון, (ב) אינטואיציה, (ג) ניסוי וטעיה, (ד) קריאה באינטרנט למשל
כאן: https://www3.nd.edu/~dthain/courses/cse40243/fall2015/intel-intro.html
או כאן: https://youtu.be/bSkpMdDe4g4 (סרטון מעולה, מומלץ מאד).
(ה) אם אתם מסתבכים תבואו לשאול אותנו, התרגיל לא אמור לקחת מעל חצי יום!

הגשה בזוגות. ניתן להוריד את התרגיל כקובץ וורד ולענות בגוף התרגיל.

א. הכנה

ודאו שיש לכם גישה למערכת לינוקס. אפשר להשתמש במחשבים בבניין 9 קומה 3; להתקין על המחשב שלכם: https://itsfoss.com/install-ubuntu-dual-boot-mode-windows/, או להתקין אובונטו על דיסק-און-קי: http://unetbootin.github.io , או להשתמש באתר כגון c9.io.

ודאו שיש לכם קומפיילר טוב ל-C++. למשל, אפשר להתקין clang++-5.0 ע"י הרצת הפקודה הבאה במסוף של אובונטו:

```
sudo apt install clang++-5.0
```

ב. תוכנית מינימלית

כיתבו תוכנית C++ מינימלית:

```
int main () { return 1234; }
```

1. קמפלו והריצו את התוכנית. בידקו את הערך המוחזר ממנה למערכת ההפעלה ע"י:

```
echo $?
```

מהו הערך המוחזר ומדוע?

2. קמפלו את התוכנית לאסמבלר ע"י:

```
clang++-5.0  --assemble  -fno-asynchronous-unwind-tables
        -fno-exceptions -fno-rtti <filename>.cpp
```

(הדגלים המתחילים ב-f אומרים לקומפיילר ליצור קוד אסמבלי נקי יותר - בלי הערות מיותרות).

קיראו את קוד האסמבלי שהתקבל. מה לדעתכם משמעות השורות שמתחילות בנקודה (.)?

3. איזה מהשורות הללו אפשר למחוק בלי לשנות את התוצאה?

(הערה: כדי לקמפל קובץ בשפת אסמבלי אפשר להשתמש ב-clang++ כרגיל בלי הדגל --assemble).

4. איזה פעולה באסמבלי משמשת להחזרת מספר שלם כלשהו מתוך פונקציה?

ג. משתנים מקומיים

שנו את הקוד בתוך main ל:

```
int i=2, j=3, k=i+j; return k;
```

5. קמפלו וקיראו את הקוד. איפה נמצאים המשתנים i, j, k?
6. לפי התשובה לסעיף א, מה לדעתכם המשמעות של %rbp?
7. לפי התשובה לסעיף ב והקוד, מה לדעתכם המשמעות של %rsp?
8. איזו פקודת אסמבלי משמשת לחיבור מספרים שלמים?
9. איזו פקודת אסמבלי משמשת לכפל מספרים שלמים?
10. איזו פקודת אסמבלי משמשת לחילוק מספרים שלמים?
11. איזו פקודת אסמבלי משמשת למציאת שארית בין מספרים שלמים? איך זה מסתדר עם סעיף ו?

ד. משתנים גלובליים

הוציאו את השורה "int i=2, j=3, k=i+j;" מתוך main והדביקו מעל main:

```
int i=2, j=3, k=i+j;
int main() { return k; }
```

12. קמפלו וקיראו את הקוד. איפה נמצאים המשתנים i, j, k עכשיו?
13. כמה ואיזה פונקציות נוספו לקוד עכשיו, מלבד הפונקציה main? מדוע הן נוספו?

ה. פונקציות

כיתבו את התוכנית הבאה:

```
int triple(int x) {   return 3*x;  }
int main() {  return triple(1111); }
```

14. קמפלו וקיראו את הקוד. איזו פקודת אסמבלי משמשת לקריאה לפונקציה?
15. מדוע הפונקציה נקראת triplei ולא רק triple?
16. מה צריך לעשות כדי ששם הפונקציה באסמבלי ישתנה ל-tripled (בלי לשנות את פעולתה)? triplec? triplel?
17. שנו את סיומת הקובץ מ-cpp ל-c, וקמפלו אותו שוב בתוספת הדגל **-ObjC** (בנוסף לכל הדגלים שהעברתם קודם). הדגל הזה אומר לקומפיילר להתייחס לקובץ כאל תוכנית בשפת Objective C. לצורך התוכנית שלנו, השפה הזאת שקולה לשפת סי שאתם מכירים. מדוע הפונקציה נקראת עכשיו רק triple בלי תוספת?

ו. אינליין

18. היפכו את הפונקציה triple ל-inline כפי שנלמד בשיעור, וקמפלו שוב. מה ההבדלים בקוד האסמבלי שהתקבל?

19. איך זה מסתדר עם מה שלמדנו בכיתה על inline?

ז. אופטימיזציות

בסעיף זה יש להוסיף לקימפול את הדגל **-O2** (בנוסף ל assemble--). הדגל אומר לקומפיילר לבצע אופטימיזציות שונות כדי להקטין את הקוד ולקצר את זמן הריצה שלו

20. הורידו את ה-inline מהפוקנציה triple וקמפלו את התוכנית. איך השפיע הדגל O2 על הפונקציה main?

21. איך השפיע הדגל O2 על הפונקציה triple? מדוע? כדי להבין טוב יותר, החליפו את המספר 3 במספרים אחרים - 6, 12, 24, 65537... מה עושה הקומפיילר ומדוע?

22. הוסיפו בחזרה את inline לפונקציה triple וקמפלו שוב. איך השפיעה הוספת inline על הפונקציה main?

23. איך השפיעה הוספת inline על הפונקציה triple?

24. נסו לשנות את הקוד של הפונקציה triple כך שיהיה מסובך יותר והקומפיילר לא יוכל "להעלים" אותו. מהו הקוד הכי פשוט שהצלחתם ליצור, שהוא מסובך מדי בשביל האופטימיזציה של הקומפיילר?