הרצאה 4- שיעורי בית 2

הגדרת תבנית ללולאה (חייבים להגדיר עבור profiling):

Every backward direct jump defines a loop

מה צריך עבור כל לולאה:

* #invocation
* average iterations #
* Total # of iterations- כמה פעמים ביקרתי בפקודת הקפיצה.

אפשר לעשות אינסטרומנטציה על כל פקודת קפיצה- לבדוק האם הקפיצה נלקחת או לא. ככה אין בעיה של קפיצה לאמצע הלולאה.

במקרה של jmp יותר מסובך כי הוא תמיד נלקח. לכן נעשה אינס' על הפקודה אחריו.

הבעיה היא כשיוצאים מלולאה- מה אם יש break באמצע?

כל ריצה של הלולאה לא בפעם הראשונה- בכל מקרה מטופלת אם עושים אינס' רק על פקודת הקפיצה. רק בפעם הראשונה כביכול מבצעים חלק מהלולאה, אבל לא סופרים אותה (אם יש break מאמצע הלולאה שנלקחת בלולאה הראשונה).

כביכול יש בעיה לא מובנת שהוא טוען שלא הגדרנו מספיק טוב. לכן מגדירים מחדש את התבנית:

Every backward direct jump defines a loop where loop iteration defined as a visit.

להשתמש בInstruction בפרמטר IARG\_BRANCH\_TAKEN- מאפשר לספור רק כאשר הקפיצה נלקחת.

קובץ שלו: loop-trip-count cpp

הוא העביר גם את הכתובת וגם את הפרמטר של "נלקח". כלומר הוסיף פרמטר IARG\_BRANCH\_TARGET\_ADDR. לבדוק האם הוא גדול או קטן מהכתובת שלי, וככה לדעת אם הקפיצה היא אחורה או קדימה.

אצלנו לולאה היא רק קפיצה מותנית אחורה (ולא וודאית כמו jmp).

* CountSeen is the sort key.
* Loop address- address of entry to loop
* Loop CountSeen- total number of iterations
* Loop CountLoopInvoked- no need I think. Ignore of possibility of jumping from the middle outward (like break).  
  How many times there was a fall through (not taken).
* MeanTaken-
* DiffCount- if there are 2 invocations right after the other- diffCount++  
  Meaning if a loop was invoked for 20 loops, then immediately invoked for a number different than 20, diffCount++.
* Routine invocations count- how many times was the routine called.

מה קורה אם יש 2 לולאות שונות (פקודות קפיצה שונות), שקופצות לאותה כתובת (אותו label- למשל L1)? איך מבדילים? לפי מספר הפעמים שכל jmp נקרא.