

מבנה מחשב

תרגיל 3

קומפילציה של switch

מועד אחרון להגשה: 11.1.2020

מטרות התרגיל: פקודות קפיצה ופקודות אריתמטיות

הנחיות הגשה:

- ההגשה ביחיד באמצעות מערכת ההגשה בכתובת:

<http://submit.cs.biu.ac.il>

- להוריד מאתר מודל את הקובץ test.c.
 - להוריד מאתר מודל את הקובץ switch.c.
 - לכתוב את הקובץ compile.c.
 - הקובץ compile.c קורא קובץ בשם switch.c ויוצר קובץ בשם switch.s.
 - לקמפל: gcc compile.c -o compile.
 - להריץ את compile ולבדוק שמתקבל קובץ switch.s תקין.
 - לקמפל: gcc test.c switch.c switch.s -fno-pie
 - להריץ ולבדוק שהפלט של הקומפיילר של C שווה לפלט של compile.
 - להגיש רק את הקובץ compile.c.
- אפשר לבצע את הבדיקות בכל מחשב יוניקס 64 ביט.

תיאור התרגיל

כפי שמוסבר בספר csapp, אפשר לקמפל switch באמצעות טבלת קפיצה.

יש לכתוב תכנית compile.c שמכילה פונקציה main() ללא ארגומנטים.

התכנית קוראת קובץ switch.c המכיל פונקציה שמבצעת פקודת switch, ויוצרת קובץ switch.s שמבצע את פקודת ה-switch.

בקובץ switch.c שהתכנית תקרא, שם הפונקציה הוא switch1, בקובץ switch.s שהתכנית תכתוב, שם הפונקציה יהיה switch2.

בקובץ switch.c שייבדק, מספר המקרים (case:) הוא 4 או יותר, ה-switch ימומש על ידי טבלת קפיצה.

בכל אחד מהמקרים (case:) תופיע אחת או יותר מהפקודות הבאות:

```
dst = src;  
dst += src;  
dst -= src;  
dst *= src;  
dst <<= count;  
dst >>= count;
```

השורות צמודות לשמאל או שיש רווח אחד.

בין dst ל-op ובין op ל-src מפריד רווח אחד, אחרי src יש נקודה-פסיק.
בכל אחת מהפקודות:

src הוא קבוע, *p1, *p2 או result.

dst הוא *p1, *p2 או result.

בפקודות ההזזה, count הוא קבוע, *p1, *p2 או result.

המקרים (case:) הם מספרים שלמים חיוביים ואינם בהכרח בסדר עולה.

מבנה הקובץ switch.c:

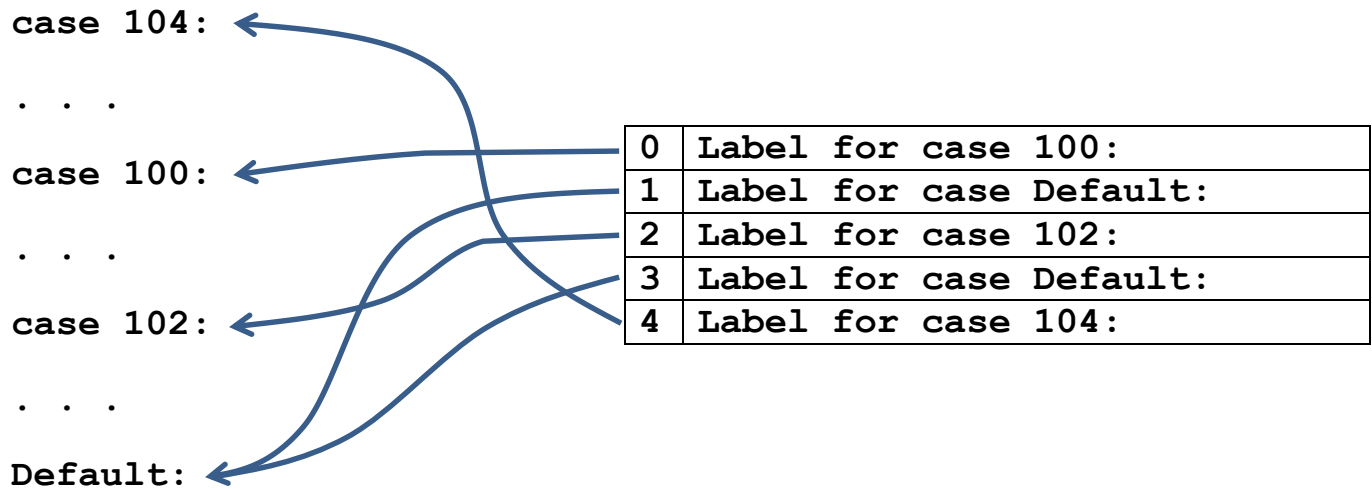
```
long switch1(long *p1, long *p2, long action)
{
    long result = 0;
    switch(action)
    {
        case . . . :
            . . . . .;
        default:
            . . . . .;
    }
    return result;
}
```

מבנה הקובץ switch.s: (כמו בספר ובמצגת):

```
.section .text
.globl switch2
switch2:
. . .
    jmp * . . .
. . .

.section .rodata
.align 8
.quad .L . . .
```

דוגמה לטבלת קפיצה



כשנבצע `switch(104)`, הפקודה `jmp *` תיגש לטבלה למקום 4 (104-100) ושם צריכה להימצא הכתובת של הקוד שמבצע `case 104:`.

כדי להכין את הטבלה, נוכל להגדיר בתכנית `compile.c` מערך של 5 מחרוזות (100-100 עד 104-100) ונוכל לאתחל את כל המחרוזות בתגית של `Default`.

נעבור על המקרים (`case:`), התגית של 104 תחליף את התגית `Default` במקום 4 (104-100) בטבלה, וכן הלאה.

נדפיס את הטבלה ל-`.rodata`.

פונקציות שימושיות בכתיבת התרגיל:

`fgets`

`fputs`

`strcpy`

`strcmp`

`if(strcmp(str1, str2)==0) // They are the same`

`or if(!strcmp(str1, str2) // They are the same`

strtok

```
char str[] = " Hello + World;";  
// split by space or ';'   
char* token = strtok(str, " ;");  
while (token != NULL) {  
    printf("%s\n", token);  
    token = strtok(NULL, " ;");  
}
```

בהגשה ייבדק הקובץ switch.c המצורף.

(בבדיקה הסופית יוחלף חלק מהמקרים)

בהצלחה