

תרגיל 1

הנחיות הגשה

1. העבודה היא ביחידים.
2. ההגשה היא עד ליום חמישי, בתאריך 1.11.2018 בשעה 23:30.
3. כל התוכנית תוגש בקובץ אחד. הגשת התרגיל תיעשה ע"י submit. עזרה ניתן למצוא באתר: <http://help.cs.biu.ac.il/submit.htm>
- כדי למנוע בעיות, עדיף להגיש את הקובץ ישירות מחשבון הלינוקס שלכם.
- שימו לב: העברת הקובץ דרך Windows יכולה לגרום לכך שלא יעבור קומפילציה. במקרה זה **הציון יהיה 0** ללא זכות לערעור.
4. יש לוודא שהתרגיל מתקמפל ורץ ללא שגיאות על גבי שרת ה-u2.
5. בשורה הראשונה (!) של הקובץ אותו אתם מגישים, יש לציין בהערה מס' ת.ז. ושם מלא. לדוג':
`// 123456789 Avi Mahshev`
6. בהצלחה ☺

רקע כללי

בתרגיל זה נתרגל את האופן בו מידע מיוצג במחשב ואת נושא ה-Endianness.

מה צריך להגיש

בכל אחת מהשאלות הבאות עליכם לממש פונקציה (תוכלו להוסיף פונקציות עזר כרצונכם). מצורף קובץ **ex1.h** עם ה-headers לפונקציות הללו ו-makefile. **עליכם להגיש קובץ בשם ex1.c** עם המימושים של הפונקציות. הקובץ `main.c`, שמצורף, כולל קריאות לפונקציות שתממשו ב-**ex1.c**. תוכלו לערוך קובץ זה כראות עיניכם כדי לבדוק את הקוד שלכם עם קלטים שונים (אינכם מגישים את `main.c`).

שאלה 1

כתבו פונקציה בשפת C בשם `int is_little_endian()`. הפונקציה מחזירה 1 אם היא קומפלה והורצה על מכונה שפועלת לפי `little endian` ומחזירה 0 אם היא קומפלה והורצה על מכונה שפועלת לפי `big endian`. הניחו שמילה היא בגודל של `long`.

שאלה 2

כתבו פונקציה בשפת C בשם `unsigned long merge_bytes(unsigned long x, unsigned long y)` הפונקציה מחזירה מילה (הניחו שהיא בגודל של `long`) שבנויה מה-**byte** least significant של `y` ומשאר ה-**bytes** של `x`. למשל, עבור הארגומנטים `x = 0x89ABCDEF12893456` ו-`y = 0x76543210ABCDEF19`, הפונקציה תחזיר `0x89ABCDEF12893419`.

שאלה 3

עבור מילה בגודל `w` ביטים, נמספר את ה-`bytes` שלה מ-0 (ה-`least significant`) עד `w/8 - 1` (ה-`most significant`).

כתבו פונקציה בשפת C בשם `unsigned long put_byte(unsigned long x, unsigned char b, int i)` הפונקציה מחזירה את `x` לאחר שה-`byte` ה-`i` שלו הוחלף בערך `b`. למשל:

```
put_byte(0x12345678CDEF3456, 0xAB, 2) --> 0x12345678CDAB3456
```

```
put_byte(0x12345678CDEF3456, 0xAB, 0) --> 0x12345678CDEF34AB
```

הערות

1. שימו לב שמיקום ה-`least significant byte` במילה תלוי ב-`endianness` של המכונה. תוכלו להיעזר בפונקציה משאלה 1 בפתרון שאלות 2,3.
2. בזמן הבדיקה יוכנסו מספר קלטים שונים ויבדקו גם מקרי קצה, אך ניתן להניח כי כל הקלטים שיוכנסו יהיו בהתאם למה שהוגדר בתרגיל.
3. אין להשתמש בספריות חיצוניות.
4. תוכנית שלא תתקמפל ו/או, תיכשל בבדיקה תקבל ציון סופי 0 – לא תינתן זכות לערעורים על כך.
5. יש צורך בהערה **משמעותית** בתחילת הפתרון לכל משימה, ורצוי להוסיף הערות גם לקוד עצמו. מתכנת חיצוני (למשל, הבודק של הקורס) שמסתכל על הקוד שלכם צריך להבין בקלות את ההיגיון של הפתרון שלכם.

בהצלחה!

