

## תרגיל מספר 5

כתבו תוכנית המקבלת מהמשתמש מחרוזת עם הצגתה על המסך בגודל מקסימלי של 18 תווים. (סיום מחרוזת הינו בלחיצת המשתמש על ENTER – המחרוזת נשמרת בזיכרון בסגמנט הנתונים)

- (1) על התוכנית להציג את המחרוזת בהיפוך אות גדולה <-- אות קטנה  
אות קטנה <-- אות גדולה (רמז: ההפרש ביניהם הינו 20h)
- (2) על התוכנית להציג את המחרוזת מהסוף להתחלה.
- (3) על התוכנית להציג כמה פעמים האות 'a' מופיעה במחרוזת.
- (4) על התוכנית להציג כמה פעמים הצירוף 'ab' מופיע במחרוזת.
- (5) על התוכנית להציג כמה פעמים הצירוף 'abc' מופיע במחרוזת.

הסבר להקדמה לתרגיל:

עבדנו עם syscall 8 כלומר הקצינו מקום של 19 בתים בסגמנט הנתונים (data) החל ממקום buf לקלט מחרוזת מהמשתמש.

.data  
buf: .space 19

מיקום  
buf

כתוצאה מההקצאה יש 19 בתים המאותחלים לערך 0 החל ממקום buf

\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18

אינדקס במחרוזת

הפעלת syscall 8 בקוד (text):

.text  
li \$a1,19  
la \$a0,buf  
li \$v0,8  
syscall

שימו לב ש syscall 8 מסיימת קלט ב null (ערך אסקי 0), אולם במידה שלא הוכנסו כל התווים יופיע לפני ה null קוד האסקי 0xa (\n) המסמן ירידת שורה.

בכל סעיפי התרגיל עלינו **לפנות לזיכרון** בלולאה לכל תו בנפרד במחרוזת התווים המתחילה ממקום buf. קיימות שתי אפשרויות בסיסיות לגשת למחרוזת בזיכרון.

(1) ניתן בעזרת הפקודה

lb \$rt,0(\$rs)

לאחר שדאגנו לאתחל את אוגר \$rs לכתובת buf בעזרת הפסיאודו

la \$rs,buf

(2) לחלופין ניתן לגשת למחרוזת buf בעזרת הפסיאודו פקודה:

lb \$rt,buf(\$rs)

במקרה זה המשמעות של ערך \$rs תהיה האינדקס בתוך המערך buf.

שימו לב שכל סעיפי התרגיל מתייחסים למחרוזת המקורית ללא שינוי, כלומר **אין בתרגיל שמירה על המחרוזת בעזרת sb**.