

תרגיל בית 7 – ADT ו-Shared Objects		
alonrs@campus.technion.ac.il	אחראי התרגיל: אלון רשלבר	תאריך הגשה: 12 בינואר 2021

בתרגיל זה אנחנו נבנה מערכת לניהול ציונים של סטודנטים. לצורך בניית המערכת, נשתמש ב-Shared Library של Linked-List מסוג ADT.

### חלק א' – דרישות המערכת

נתונה לכם תכנית בשם prog.exe. בעזרת תוכנה זו, מזכירות הפקולטה יכולות להכניס פרטים של סטודנטים, להוסיף לכל סטודנט/ית את הקורסים אותם הוא/היא עשה/תה, ולהדפיס מידע למסך. התוכנית קוראת פקודות מהמשתמש, לפי הפורמט הבא, ומבצעת אותן אחת אחת (אינטרפטר). להלן הפקודות הנתמכות:

Command: add-student NAME ID

Description: Adds a new student with NAME and ID to the system.

Command: add-grade STUDENT-ID COURSE-NAME GRADE

Description: Adds a new course COURSE-NAME with GRADE to the student with id ID.

Command: calc-avg STUDENT-ID

Description: Calculates the average of a student with id ID.

Command: print-student STUDENT-ID

Description: Prints the student's information.

Command: print-all

Description: Prints information of all students.

Command: echo DATA [DATA...]

Description: Similar to Bash's echo, echos the input text to the screen.

ניתן להכניס קלט ל-prog.exe בעזרת file-redirection (ראו סדנה 1).

את כל הפקודות הנ"ל התוכנית מבצעת בעזרת קריאה לפונקציות בספרייה libgrades.so – אותה אתם תצטרכו לממש ולקמפל. שימו לב לקובץ הממשק grades.h – יש שם את מפרט כל הפונקציות בהן יש לתמוך. שימו לב במיוחד למקרי קצה וערכי חזרה רצויים. הקוד אותו תממשו צריך להיות כתוב בקובץ בשם grades.c.

לצורך בניית libgrades.so, תוכלו להיעזר בספרייה liblinked-list.so הנתונה לכם, ובה ממומשת רשימה מקושרת בצורת ADT. חשבו על אילו רשימות תצטרכו, ואילו איברים נמצאים בכל רשימה.

### חלק ב' – liblinked-list.so

נתונה לכם ספרייה עם מימוש של רשימה מקושרת. עברו על הפונקציות בקובץ הממשק linked-list.h ושימו לב לפרטים הבאים:

- מהן פונקציות המשתמש של ה-ADT? מה המבנה שלהן, ומה צריך לעשות בהן?
- אילו פונקציות נתמכות ע"י liblinked-list? בפרט, איפה בא לידי ביטוי העובדה שמדובר ב-ADT?
- איך ניצור רשימה מקושרת? איך נוסף אליה איברים? איך נעבור על האיברים ברשימה?

4. שימו לב למונח איטרטור (Iterator) המופיע בממשק – זהו מבנה נתונים ש"מצביע" על איברים ברשימה מקושרת. אנחנו לא יודעים ממה הוא מורכב, אבל אנחנו כן יודעים שיש פונקציות מעטפת שבעזרתן ניתן להזיז איטרטור קדימה ואחורה, ניתן לאתחל איטרטור לראש או סוף הרשימה, וניתן להוסיף איברים במקום שרירותי ברשימה בעזרת האיטרטור.

### חלק ג' – בניית MakeFile

עליכם ליצור Makefile אשר מקמפל את grades.c ל-grades.o, ומלנקג' את ה-Object לספרייה דינאמית בשם libgrades.so. שימו לב:

- עליכם להצהיר במהלך הלינקוג' על השימוש בספרייה liblinked-list.so – ראו סדנה 7 ותרגול 8.
- עליכם לקמפל את האובייקט עם דגלים מיוחדים – ראו סדנה 7.
- שימו לב שבקריאה ל-prog.exe יתכן כי ה-Loader לא יכיר את liblinked-list.so ו-libgrades.so (שכן הספריות לא מותקנות במערכת). ראו סדנה 7 על איך להתגבר על קושי זה.

### חלק ד' – דיבוג ומדידת יעילות

- וודאו כי התוכנית רצה כמו שצריך, שאין דליפות זיכרון, ושאין בעיות כלשהן. במידה ויש, תוכלו למצוא אותן כפי שלמדנו בעזרת GDB.
- שימו לב כי בסיום הריצה התכנית פולטת ל-stderr את סך זמן ריצת התוכנית במילי שניות. ההגשה הנכונה והיעילה ביותר תזכה את בעליה ב-0.5 נקודות בונס לציון הסופי. (אופן המדידה שלנו: הרצת התוכנית שלכם 20 פעמים על אותו הקלט, וביצוע ממוצע).

### דגשים מיוחדים

- על ה-Makefile שלכם לקמפל את grades.c ל-libgrades.so. אנחנו נשתמש ב-Makefile שלכם – באחריותכם לוודא שהוא עובד ופועל כנדרש!
- יש לוודא שהתוכנית פועלת ללא דליפות זיכרון בעזרת Valgrind (ראו סדנה 6). לתוכניות עם דליפות זיכרון יורדו נקודות. ב-prog.exe וב-liblinked-list.so אין דליפות זיכרון. אם יש לכם דליפות, כנראה שיש לכם באג בקוד.
- אתם יכולים להשתמש ב-GDB. לא תוכל לדבג את הפונקציות ב-prog.exe וב-liblinked-list.so (כי אין לכם את הקוד מקור), אבל כן תוכלו לדבג את הקוד שלכם בספרייה הדינאמית (GDB יודע לקרוא קבצי ELF של ספריות דינאמיות ולקשר אותם לקוד שלכם בזמן ריצה).
- נתון לכם קובץ בדיקה (קלט ופלט רצוי). כתבו עוד קבצי קלט, וודאו שהקוד שלכם פועל כראוי.
- הקוד שלכם חייב לעמוד בקונבנציות הקוד כפי שראינו בתרגולים 1 ו-2. ירדו נקודות למי שלא יעבוד לפי הקונבנציות.

### הוראות הגשה:

- עברו היטב על הוראות ההגשה של תרגילי הבית המופיעים באתר טרם ההגשה! ודאו כי התכנית שלכם עומדת בדרישות הבאות:
  - התכנית קריאה וברורה
  - התכנית מתועדת היטב לפי דרישות התייעוד המופיעות באתר.
- יש להגיש לינק ל-Repository המכיל את הקבצים (שימו לב לשמות הקבצים עם lower case). על ה-Repository להיות בעל הרשאות Public. בעת בדיקת התרגיל, אנו נבצע clone ל-Repository שלכם, נבצע קומפילציה בעזרת ה-makefile שתגישו ונבדוק את התוכנית. שימו לב לשם הספרייה הנדרש מכם!
- אין הגשות חוזרות לתרגיל! בבקשה, שימו לב שהגשתם הכל כנדרש! בפרט, יש להגיש לפי הפורמט הבא:

<https://github.com/your-username/repository-name>

0123456789 student\_1\_mail@campus.technion.ac.il first\_name\_1 last\_name\_1

0123456789 student\_2\_mail@campus.technion.ac.il first\_name\_2 last\_name\_2

4. שאלות בנוגע לתרגיל יש להפנות לפורום התרגיל ב-moodle בלבד. ניתן לשלוח שאלות במייל למתרגל האחראי על התרגיל בלבד, ורק במידה והשאלה מכילה פתרון חלקי.

#### סיכום מפרט התרגיל:

prog.exe liblinked-list.so linked-list.h grades.h test.in test.out	קבצים נתונים
grades.c Makefile	קבצים להגשה
libgrades.so	שם הספרייה שיש ליצור
ע"י יצירת Public Repository ב-GitHub	אופן ההגשה
אלון רשלבך alonrs@campus.technion.ac.il	אחראי התרגיל

**בהצלחה!!**