מבוא לתכנות מערכות תרגיל בית מספר 0

2014/15 סמסטר חורף

23/03/2015 : מאריך פרסום

31/03/2015 : תאריך הגשה

234122cs@gmail.com מתרגל אחראי לתרגיל: דני רסין

1 הערות כלליות

- תרגיל זה מהווה 1% מהציון הסופי
- בכל השאלות בנוגע לתרגיל, יש לפנות לדני (ולא למתרגל אחר). בכל פנייה בקשר לתרגיל, נא mtm0 : (Subject). כתוב בשורת הנושא

2 הקדמה

מטרת תרגיל זה היא ביצוע מספר צעדים ראשונים בעבודה מרחוק על שרתים ובסביבת UNIX. התרגיל מורכב משלושה חלקים:

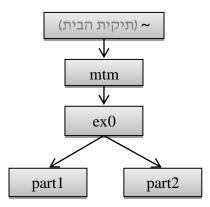
- 1. התחברות ל-t2 וביצוע פעולות בסיסיות.
- t2. כתיבת תכנית ראשונה ב-C, הידורה ובדיקתה על שרת ה-2
 - 3. מציאת באגים בתכנית לדוגמה עייי שימוש בדיבאגר.

:הערות

- .t2 באתר הקורס מופיעים מדריכים המסבירים כיצד להתחבר ל-
 - התרגיל להגשה ביחידים.
- יש להגיש את חלק ב׳ וג׳ של התרגיל כך שייבדק על ידי הבודק האוטומטי אשר בשימוש בקורס.

3 חלק א' - התחברות ופעולות בסיסיות

- 1. התחברו לחשבונכם בשרת ה- t2 הוראות מפורטות על הדרכים השונות לעשות זאת נמצאות במדריך ההתחברות ל- t2 שבאתר תחת במדריך ההתחברות ל- t2 שבאתר תחת במדריך ההתחברות ל- t2
- http://webcourse.cs.technion.ac.il/234122/Spring2015/ho/WCFiles/Connecting_to_t2.pdf
- 2. לאחר ההתחברות צרו את התיקיות הבאות תחת תיקית הבית שלכם: תיקיה בשם mtm, בתוכה תיקיה בשם part2 ובתוכה תיקיות בשמות part2 ו-part2. כך שיווצר המבנה הבא:



כעת נרצה להעתיק את הקבצים הדרושים לפתרון שאר חלקי התרגיל מחשבון הקורס לחשבונכם
 -mtm/public/1415b/ex0/part1 האישי. לשם כך העתיקו את כל הקבצים הנמצאים תחת התיקיה

teat1.in, יבצים שונים: 9 אותה אכן הועברו מקודם. ודאו שלאחר ההעתקה אכן מקודם. mtm_sol , test1.out, test2.in, test3.in, test3.out, test4.in, test4.out

5 באותה בורה העתיקו מ-mtm/public/1415b/ex0/part2. ודאו שאכן הועתקו - .mtm/public/1415b/ex0/part2. מ- .mtm_buggy.c, test1.in, test2.in, test2.out קבצים שונים:

t2-1 חלק ב' - תכנית ראשונה ב-4

בחלק זה נכתוב ונבדוק תכנית המקבלת רשימת מספרים אשר המשתמש מכניס כקלט, וסוכמת את המעריכים של המספרים המהווים חזקה שלמה של 2.

4.1 מפרט התכנית

התכנית אשר תיקרא mtm_tot תופעל משורת הפקודה ותפעל כלהלן:

- 1. רושמת "Enter size of input:" ומקבלת מהמשתמש כקלט מספר שלם.
- 2. אם המספר שהתקבל אינו גדול ממש מ-0 התכנית תדפיס "Invalid size" ותסתיים.
- 3. כעת התכנית מדפיסה "Enter numbers:" ומקבלת מהמשתמש כקלט מספרים שלמים בהתאם למספר שנקבע בשלב 1. אם יש בעיה באחד המספרים בקלט (למשל מוכנסים תווים שאינם למספר שנקבע בשלב 1. אם יש בעיה "Invalid number" ומסתיימת.
 - 4. התכנית מדפיסה את המספרים המהווים חזקה שלמה של 2 שהוכנסו בקלט ואת סכומם המעריכים בפורמט הבא:

שורות מהמבנה לכל חזקה שלמה מ is a power of 2: $a=2^j$ לכל חזקה שלמה שורות שורות מהמבנה לפי הסדר בו התקבלו בקלט.

. במבנה b הוא סכום המעריכים Total exponent sum is b הוא סכום המעריכים.

4.2 דגשים והמלצות

- כל הקוד בתרגילי הבית בקורס צריך לציית למוסכמות הקוד (code conventions) המפורסמות באתר תחת Course Material.
 - כדי להימנע מבעיות עם הבודק האוטומטי על התכנית להחזיר 0 בכל מקרה.

4.3 הידור ובדיקה

: כדי להדר את התכנית ולהריצה עליכם להשתמש בשורת הפקודה gcc כדי להדר את ולהריצה עליכם להשתמש בשורת -std=c99 -Wall -pedantic-errors -Werror -DNDEBUG

.mtm_tot ודאו ששם קובץ ההרצה הוא אכן

:כלומר

 $\verb|> gcc -std=c99 - Wall - pedantic-errors - Werror - DNDEBUG part 1.c - o mtm_tot|\\$

כדי לבדוק את התכנית מסופקים לכם קבצי בדיקה. הקבצים מכילים קלט לתכנית ופלט צפוי לכל קלט. את קבצי הבדיקה ניתן למצוא תחת תיקית הקורס ב-t2 בכתובת: mtm/public/1415b/ex0/part1~ הקבצים test1.in - test4.in הם קבצי הקלט ואילו הקבצים test1.out - test4.out הם קבצי הפלט (בהתאמה). אלו אותם קבצים אשר העתקתם בחלק הקודם של התרגיל. כדי לבדוק את התכנית בעזרת הקבצים בצעו את הפעולות הבאות:

- 1. הדרו את הקוד
- והפלט הסטנדרטי הוא לקובץ test#.in הריצו את התכנית כך שהקלט הסטנדרטי הוא מהקובץ. והפלט הסטנדרטי הוא לקובץ ומני כלשהו. למשל כך:

> ./mtm_tot< test1.in > tmpout

עליכם לוודא שקובץ הפלט הזמני זהה לגמרי לקובץ הפלט הצפוי. ניתן לעשות זאת ע"י שימוש מליכם לוודא שקובץ הפלט הזמני למוד על הפקודה diff השתמשו בפקודה מנת ללמוד על הפקודה לדוגמה - שימוש ב-diff לבדיקת הקובץ הקודם:

> diff test1.out tmpout

אם הקבצים זהים לא יודפס כלום, אם יש הבדל יודפסו ההבדלים בין הקבצים.

בנוסף, מסופק לכם קובץ בשם mtm_sol, המהווה גרסה מקומפלת של התכנית אותה אתם צריכים לכתוב. ניתן להשתמש בו כדי לבדוק מקרים נוספים ולייצר טסטים נוספים.

הלק ג' - דיבוג 5 n - 5

נעבור כעת לתיקיה part2 שיצרנו בחלק אי. תיקיה זו מכילה קובץ קוד בשם part2 וקבצי בדיקה ופלט צפויים.

- התכנית buggy אמורה לקלוט מהמשתמש מספר מחרוזות (בדומה לתכנית בחלק הקודם)
 ולאחר מכן להדפיס את המחרוזת הארוכה ביותר, המחרוזת המינימלית לפי סדר לקסיקוגרפי ואת
 המחרוזת המקסימלית לפי סדר זה.
 - הדרו את התכנית (לא לשכוח את כל הדגלים) ונסו להריץ את התכנית עם קובץ הבדיקה הראשון.
- 2. התכנית מתרסקת בשגיאת "Segmentation fault", משמעות השגיאה היא שהתכנית מנסה לקרוא ערכים מתאי זיכרון שאינם מוקצים לה. בדייכ שגיאות אלה נובעות ישירות משימוש לא נכון NULL במצביעים או פשוט נסיון לקרוא מצביע שערכו
- אמנם הקוד בתכנית $mtm_buggy.c$ אינו גדול במיוחד, אך כבר בכמות כזו של קוד יש להשקיע זמן מה למציאת הנקודה בה מתרחשת השגיאה. כדי למצוא את השגיאה הזו בקלות ניתן להשתמש בדיבאגר gdb אשר מותקן על שרת ה-t2.
- הריצו את הפקודה הבאה אשר מתחילה את הדיבאגר עם התכנית mtm_buggy (כאשר mtm_buggy הריצה) הוא שם קובץ ההרצה)

> qdb mtm buggy

- gdb הוא דיבאגר העובד בטרמינל. כדי להשתמש בו יש להכניס פקודות בדומה לשימוש הרגיל בטרמינל. כדי להריץ את התכנית ניתן להשתמש בפקודה run (כדי להפנות קלט ופלט פשוט מוסיפים את ההפניות כמו בדייכ). נסו להריץ את התכנית עם קובץ הקלט מתוך gdb.
- התכנית רצה תחת הדיבאגר כמו בריצה רגילה, אך הפעם כאשר נגיע לגישה הלא חוקית הדיבאגר יעצור את התכנית ויודיע על השגיאה. בשלב זה נוכל להשתמש למשל בפקודה bt כדי להדפיס את מצב מחסנית הקריאות. פקודות נוספות ניתן ללמוד פשוט עייי שימוש בפקודה help.
- 3. עדיין קיימת בעיה המקשה עלינו: המידע במחסנית הקריאות אינו מפורט מספיק. כדי לאפשר ל-gdb להדפיס מידע מדויק יותר יש להדר מחדש את התכנית ולהוסיף את הדגל g-g. דגל זה שומר מידע עבור דיבאגרים בתכנית ומאפשר להם להתייחס לקוד המקור.
- .4. צאו מהדיבאגר (ע"יי הפקודה quit), הדרו מחדש את התכנית והריצו אותה תחת gdb. הפעם כאשר תדפיסו את מצב המחסנית תקבלו פירוט של השורות בקוד מהן התבצעו הקריאות לכל פונקציה. למעשה, תקבלו את השורה המדויקת בה קרתה השגיאה. שימו לב ששורה זו היא חלק מפונקציה שמימושה לא נתון ולכן עליכם לחפש את הבאג בפונקציות הקוראות לה.
 - .5 תקנו את השגיאה.
- . לאחר תיקון השגיאה הריצו שוב את הקוד ותיווכחו לדעת שקיימת עוד שגיאה בקוד. מצאו ותקנו גם אותה.
- 7. כעת התכנית עובדת נכון עם הדוגמה הראשונה אך זה אינו מבטיח את נכונותה. הריצו את התכנית עם קובץ הבדיקה השני ומצאו את השגיאה הנוספת שהוא חושף לולאה אינסופית. (הערה: כדי לעצור תכנית שנתקעה בלולאה אינסופית ניתן ללחוץ על Ctrl+C.)
 - 8. לאחר תיקון כל שלושת השגיאות ודאו שהתכנית מוציאה פלט זהה לזה שבקבצי הפלט.

6 הגשה

יש להגיש את חלק ב׳ וג׳ בהגשה אלקטרונית. תוכלו לנצל הזדמנות זו להיכרות עם הבודק האוטומטי ולחסוך אי-נעימויות בתרגילים הבאים.

Electronic ,Exercise 0 ,Assignments את ההגשה האלקטרונית יש לבצע דרך אתר הקורס. תחת zip (ו- zip , submission פובץ ההגשה צריך להיות קובץ zip למתוקו).

- אין צורך לצרף קבצים שסופקו לכם, על קובץ ה-zip להכיל רק את קבצי ה-C שכתבתם בעצמכם.
- על הקובץ להיות מכווץ כ-zip (לא rar או כל דבר אחר) כאשר קבצי הקוד נמצאים בתיקיה .zip הראשית בקובץ ה-zip.

- סטודנטים המעוניינים לנצל ימי איחור, צריכים להגיש קובץ טקסט בודד תחת HW0_LATE המכיל מספר יחיד שבו רשומים מספר ימי האיחור אותם אתם מעוניינים לנצל.

7 שינויים עדכונים והודעות בנוגע לתרגיל

כל ההודעות הנוגעות בתרגיל ימצאו באתר של הקורס באתר של הקורס בתרגיל ימצאו באתר ליכם לעקוב בדואר. עליכם לעקוב בדף התרגילים. דף זה יכיל שאלות ותשובות נפוצות. רק הודעות דחופות תשלחנה בדואר. עליכם לעקוב אחר האתר והעדכונים שיפורסמו בו.

בהצלחה!