

ב"ה

**תרגיל מס' 1 – תהליכים וקבצים**

**הוראות הגשה**

- שאלות בנוגע לתרגיל נא לכתוב בפורום הפיאצה תחת ex1.
- מועד אחרון להגשה: 11/05/17 23:59.
- יש לשלוח את הקבצים באמצעות האתר:  
<https://submit.cs.biu.ac.il/cgi-bin/welcome.cgi>  
לפני חלוף התאריך הנקוב לעיל.
- שם ההגשה של תרגיל ex1: 1.
- יש להקפיד מאוד על כל הוראות עיצוב הקלט והפלט כפי שלמדתם בקורס C (בדיוק אותו coding style).
- חובה לבדוק כל פונקציה האם היא הצליחה או לא, אם היא לא הצליחה יש לתת הודעה מתאימה ל STDERR ולסיים את התכנית (בצורה נקייה כמובן).
- להזכירכם, העבודה היא אישית. "עבודה משותפת" דינה כהעתקה.
- אין להדפיס שום דבר מעבר למה שנתבקש בתרגיל.
- שימו לב להערות בסוף התרגיל.
- יש לוודא שהתרגיל מתקמפל ורץ על ה U2 ללא שגיאות/אזהרות.

## השוואת קבצים

### הנחיות עבור ex11

- שם התרגיל: ex1
- שם קובץ מקור (source file) שיש לשלוח: ex11.c

כתבו תכנית המקבלת שמות ו path של 2 קבצים כפרמטרים ל main ובודקת האם 2 הקבצים זהים.  
אם הקבצים דומים (הסבר בהמשך) התכנית תחזיר 2, אם הקבצים זהים (מכילים בדיוק אותו תוכן) התכנית תחזיר 1, אחרת תחזיר 3 (שימו לב, לא מדובר כאן בתכנית שמדפיסה למסך).  
שני קבצים יהיו דומים אם הם מכילים את אותו טקסט אך קיים הבדל בשימוש באותיות קטנות או גדולות, ריווח ו/או ירידות שורה.

לדוגמא, ששת השורות הבאות דומות בטקסט:

1. 12ab23
2. 12Ab23
3. 12aB23
4. 12AB23
5. 12 aB 23
6. 12  
ab2  
3

כאשר אתם מקמפלים את התכנית תנו לה את השם comp.out במקום a.out  
דוגמא להרצת התכנית:

```
[os2017@localhost ~]$ ./comp.out /home/os2017/code/1.txt /home/os2017/2.txt
```

לאחר הרצת התכנית אם הקבצים זהים לחלוטין וארשום את הפקודה \$? echo אקבל את הערך 1, אם הם רק דומים אקבל את הערך 2, אחרת אקבל את הערך 3.

לדוגמא (בהנחה שקבצים 1.txt ו 2.txt דומים):

```
[os2017@localhost ~]$ ./comp.out /home/os2017/code/1.txt /home/os2017/2.txt
[os2017@localhost ~]$ echo $?
2
[os2017@localhost ~]$
```

דוגמא שניה:

```
[os2017@localhost ~]$ ./comp.out /home/os2017/code/1.txt /home/os2017/1.txt
[os2017@localhost ~]$ echo $?
1
[os2017@localhost ~]$
```

## הרצה + השוואת קבצים

### הנחיות עבור ex12

- שם התרגיל: ex1
- שם קובץ מקור (source file) שיש לשלוח: ex12.c

כתבו תכנית המקבלת שם ו path של קובץ קונפיגורציה כפרמטר ל main (בשורת הרצת התכנית). קובץ הקונפיגורציה מכיל 3 שורות:

שורה 1: מיקום של תיקייה המכילה תתי תיקיות (כל תתי תיקייה זה בעצם שם משתמש) המכילות קבצי C.

שורה 2: מיקום ושם של קובץ המכיל קלט

שורה 3: מיקום ושם של קובץ המכיל את הפלט הנכון עבור הקלט משורה 2.

על התכנית שלכם להיכנס לכל תתי התיקיות שבתוך התיקיה משורה 1, לחפש בתקיה (ובתתי התיקיות שלה) קובץ C, לקמפל את קובץ ה C שמופיע בתיקיה ולהריץ את קובץ הריצה שנוצר כתוצאה מהקימפל עם הקלט שמוכל במיקום משורה 2 (התכנית שתריצו קולטת מ stdin ומדפיסה ל stdout).

הניחו שלכל היותר יהיה קובץ C אחד בתיקיה (יכול להיות שלא יהיה קובץ C בכלל).

את הפלט של התכנית יש להשוות עם הפלט הרצוי שמיקומו מגיע משורה 3 בעזרת התכנית comp.out שממשתם בסעיף הקודם (פשוט תריצו את התכנית comp.out ותתנו לה כפרמטרים ל main את המיקום של הפלט הנכון [נמצא בקובץ הקונפיגורציה בשורה 3] ואת הפלט של התכנית של הסטודנט אותה הרצתם).

התכנית שלכם צריכה לייצר קובץ (בתיקיה שממנה הורצה התכנית שלכם) בשם results.csv שמכיל עבור כל שם משתמש (שם תתי תיקייה) את ציונו בהתאם לתשובה ש comp.out החזירה (אפס עד מאה) ואת הסיבה. יש לרשום את התו " ", בין שם המשתמש לבין ציונו ובין ציונו לבין הסיבה.

### סיבות אפשריות:

1. NO\_C\_FILE - אין בתקיה קובץ עם סיומת c. הציון שינתן, 0.
2. COMPILATION\_ERROR - בעיית קומפילציה (קובץ לא מתקמפל). הציון שינתן, 0.
3. TIMEOUT - התרגיל רץ יותר מ 5 שניות. הציון שינתן, 0.
4. BAD\_OUTPUT - הפלט שונה מהפלט הרצוי. הציון שינתן, 0.
5. SIMILAR\_OUTPUT - הפלט שונה מהפלט הרצוי אך דומה. הציון שינתן, 70.
6. WRONG\_DIRECTORY - קובץ הקוד לא נמצא בתתי התיקיה בעלת שם הסטודנט. הציון שינתן, 10-100. כאשר x הינו עומק התיקיות שבו נמצא הקובץ.
7. MULTIPLE\_DIRECTORIES - קיימות מספר תיקיות בתוך תתי התיקיה בעלת שם הסטודנט. הציון שינתן יהיה 0.
- לדוגמא אם הקובץ נמצא בתקיה home/os2017/ex1/israel/1/2/3/4 יהיה 60. במקרה שהציון המתקבל הינו שלילי יש לעדכן ל-0.
8. GREAT\_JOB - התרגיל נתן את התוצאה הנכונה. הציון שינתן, 100.

home/os2017/ex1/Israel/1  
home/os2017/ex1/Israel/2  
home/os2017/ex1/Israel/3

כלומר פקודת ls תדפיס:

דוגמא לתוכן קובץ results.csv :

```
chen,100,GREAT_JOB
dan,0,NO_C_FILE
moshe,0,TIMEOUT
dolev,0,COMPILATION_ERROR
igor,0,BAD_OUTPUT
guy,70,SIMILLAR_OUTPUT
sam, 60, WRONG_DIRECTORY
```

דוגמא לתוכן קובץ הקונפיגורציה :

```
/home/os2017/ex1
/home/os2017/code/input.txt
/home/os2017/code/output.txt
```

דוגמא לתוכן קובץ קלט (שורה 2) :

```
1
3 4
4
```

דוגמא לתוכן קובץ פלט (שורה 3) :

```
Please enter operation
Please enter two numbers
The sum is 7
Please enter operation
Bye
```

דוגמא להפעלת קובץ הריצה של התכנית שלכם :

```
[os2017@localhost ~]$ ./a.out /home/os2017/confFile.txt
```

דוגמא לתנאי התקיות שמופיעות תחת התיקיה שהכתובת אליה מופיע בשורה 1 בקובץ הקונפיגורציה :

```
u2 levypri1 141 : cd ex1
~/ex1
u2 levypri1 142 : ls
chen dan dolev guy igor moshe sam
u2 levypri1 143 : cd dan
u2 levypri1 144 : ls
1.c
u2 levypri1 145 : █
```

הערות:

1. התכניות יכולות להדפיס למסך מה שהן רוצות, לא צריך להסתיר את ההדפסות.
2. הניחו שה gcc מוגדר ב path ולכן אין בעיה להשתמש ב execvp
3. השתמשו בקריאות המערכת fork,dup,dup2,exec,open,write,read. **אסור להשתמש בפונקציות ספרייה אלטרנטיביות לפונקציות הנ"ל.**
4. הניחו שהתכנית comp.out תהיה בתקיה הנוכחית (./) שממנה יריץ הבודק את קובץ הריצה של חלק ב' של התרגיל (הבודק יקמפל את התרגיל הראשון שלכם ורק אח"כ את התרגיל השני).
5. במצב שקריאת מערכת (SYSCALL) נכשלה יש להדפיס הודעת שגיאה בעזרת הפונקציה write ל file descriptor מספר 2 (stderr)
6. כל שורה בקובץ הקונפיגורציה לא תעלה על 160 תווים
7. אין חשיבות לסדר של הסטודנטים הרשומים בקובץ results.csv

8. חובה למחוק את כל הקבצים שהתכנית שלכם יצרה (חוץ מהקובץ results.csv) בסוף הריצה. מחיקה תעשה בעזרת קריאת המערכת unlink
9. אם תכנית של סטודנט לא מסיימת את עבודתה תוך 5 שניות אתם לא צריכים לדאוג לסיים אותה, אך אתם צריכים להתייחס לזה בקובץ results.csv ע"י ציון 0 וסיבה תואמת.
10. בכל תיקיה בלינוקס יש "." ו ".." (קישור לתיקיה הנוכחית ולתיקית האב), שימו לב לכך כאשר אתם משתמשים ב readdir.

בהצלחה!

