

ב"ה

תרגיל מס' 3 – תהליכים וקבצים

הוראות הגשה

- שאלות בנוגע לתרגיל יש לשלוח בפורום הקורס במודל.
- מועד אחרון להגשה: 23:59 29/04/19.
- **לא תינתן הארכה נוספת משום סיבה, אנא תכננו בהתאם את הזמן!**
- 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 3 חלק א': ex31 ושל חלק ב': ex32
- **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.
- יש לוודא שהתרגיל מתקמפל ורץ על ה U2 ללא שגיאות/אזהרות.
- **שימו לב להערות בסוף התרגיל**

השוואת קבצים – חלק א'

הנחיות עבור ex31

- שם התרגיל: ex31
- שם קובץ מקור (source file) שיש לשלוח: ex31.c
- יש לכתוב שם מלא ות.ז. בראש הקובץ.

כתבו תכנית המקבלת שמות ו path של שני קבצים כפרמטרים ל main ובודקת האם שני הקבצים זהים.

אם הקבצים זהים (מכילים בדיוק אותו תוכן) התכנית תחזיר 1, אם הקבצים דומים (הסבר בהמשך) התכנית תחזיר 3, אחרת תחזיר 2 (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:

```
[os2019@localhost ~]$ ./comp.out /home/os2019/code/1.txt /home/os2019/code/2.txt
לאחר הרצת התוכנית אם הקבצים זהים לחלוטין וארשום את הפקודה $? echo אקבל את הערך 1, אם הם רק דומים אקבל את הערך 3, אחרת אקבל את הערך 2.
```

כלומר בהנחה שקבצים 1.txt ו 2.txt דומים:

```
[os2019@localhost ~]$ ./comp.out /home/os2019/code/1.txt /home/os2019/code/2.txt
[os2019@localhost ~]$ echo $?
3
[os2019@localhost ~]$
```

דוגמא נוספת:

```
[os2019@localhost ~]$ ./comp.out /home/os2019/code/1.txt /home/os2019/code/1.txt
[os2019@localhost ~]$ echo $?
1
[os2019@localhost ~]$
```

השוואת קבצים – חלק ב'

הנחיות עבור ex32

- שם התרגיל: ex32
- שם קובץ מקור (source file) שיש לשלוח: ex32.c
- יש לכתוב שם מלא ות.ז. בראש הקובץ.

כתבו תכנית המקבלת שם ו path של קובץ קונפיגורציה כפרמטר ל main (בשורת הרצת התכנית). קובץ הקונפיגורציה הזה 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.
(ראו דוגמאות בהמשך).

על התכנית שלכם להכנס לכל תתי התקיות שבתוך התיקייה משורה 1, לחפש בתיקיה (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 ולכן תשתמשו ב i/o redirection).
הניחו שיהיה לכל היותר 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 בחלק א' של התרגיל (פשוט הריצו את התכנית comp.out ותתנו לה כפרמטרים ל main את המיקום של הפלט הנכון [שנמצא בקובץ הקונפיגורציה בשורה 3] ואת הפלט של התכנית של המשתמש אותה הרצתם).

התכנית שלכם צריכה לייצר קובץ (בתיקייה 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 בתיקיה (או בתתי התיקיות) קובץ עם סיומת c. **הציון שינתן, 0.**
2. COMPILATION_ERROR - בעיית קומפילציה (קובץ לא מתקמפל). **הציון שינתן, 20.**
3. TIMEOUT - התרגיל רץ יותר מ 5 שניות. **הציון שינתן, 40.**
4. BAD_OUTPUT - הפלט **שונה** מהפלט הרצוי. **הציון שינתן, 60.**
5. SIMILAR_OUTPUT - הפלט **שונה** מהפלט הרצוי **אך דומה**. **הציון שינתן, 80.**
6. GREAT_JOB - הפלט **זהה** לפלט הרצוי. כלומר, התרגיל נתן את התוצאה הנכונה. **הציון שינתן, 100.**

שימו לב, כל סיבה עומדת בפני עצמה. כלומר, אין מצב שידרוש שרשור של סיבה אחת עם אחרת.

דוגמא לתוכן קובץ results.csv :

```
Chen,100,GREAT_JOB
Dan,0,NO_C_FILE
Moshe,40,TIMEOUT
Dolev,20,COMPILATION_ERROR
Igor,60,BAD_OUTPUT
Guy,80,SIMILAR_OUTPUT
```

דוגמא לתוכן קובץ הקונפיגורציה :

```
/home/os2019/ex3
/home/os2019/code/input.txt
/home/ os2019/code/output.txt
```

דוגמא לתוכן קובץ קלט (שורה 2) :

```
1
3 4
4
```

דוגמא לתוכן קובץ פלט (שורה 3) :

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

דוגמא להפעלת קובץ הריצה של התכנית שלכם :

```
[os2019@localhost ~]$ ./a.out /home/os2019/confFile.txt
```

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

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

לנוחיותכם, מצ"ב לתרגיל זה תיקייה המכילה דוגמאות לבדיקה. תיקייה זו מכילה:

1. תיקייה students המכילה תיקיות של שמות משתמשים
2. קובץ קונפיגורציה
3. קובץ input
4. קובץ output
5. קובץ results.csv

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. אתם רשאים להשתמש בכל קריאות המערכת שנלמדו בתרגולים עד היום.
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 החלק הראשון ורק אח"כ את החלק השני).
5. במצב שקריאת מערכת (`SYSCALL`) נכשלה יש להדפיס את הודעת השגיאה
"Error in system call" בעזרת הפונקציה `write` ל `file descriptor` מספר 2 (`stderr`)
6. כל שורה בקובץ הקונפיגורציה לא תעלה על 150 תווים. אתם יכולים גם להניח שאורך ה `path` המקסימלי המוביל לקובץ C גם לא יעלה על 150 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 40 וסיבה תואמת.
10. בכל תיקיה בלינוקס יש "." ו ".." (קישור לתיקיה הנוכחית ולתיקית האב), שימו לב לכך כאשר אתם משתמשים ב `readdir`.

בהצלחה !