

תרגיל מס' 1 – קבצים ותהליכים

הנחיות עבור חלק א':

כתבו תכנית המקבלת path של 2 קבצים כפרמטרים ל main ובודקת האם 2 הקבצים זהים. אם הקבצים זהים התכנית תחזיר 2, אחרת תחזיר 1. כאשר אתם מקמפלים את התכנית תנו לה את השם comp.out במקום a.out

דוגמא להרצת התכנית (בהנחה שהקבצים 1.txt, 2.txt שונים):

```
shani@shani-VirtualBox:~$ ./comp.out /home/shani/Dropbox/Operating
Systems/Homework/code/1.txt /home/shani/Dropbox/Operating Systems/Homework/2.txt
```

לאחר הרצת התכנית אם הקבצים זהים וארשום את הפקודה `echo $?` אקבל את הערך 2, אחרת אקבל את הערך 1.

לדוגמא:

```
shani@shani-VirtualBox:~$ ./comp.out /home/shani/Dropbox/Operating
Systems/Homework/code/1.txt /home/shani/Dropbox/Operating Systems/Homework/2.txt
shani@shani-VirtualBox:~$ echo $?
1
```

דוגמא שנייה:

```
shani@shani-VirtualBox:~$ ./comp.out /home/shani/Dropbox/Operating
Systems/Homework/code/1.txt /home/shani/Dropbox/Operating Systems/Homework
/code/1.txt
shani@shani-VirtualBox:~$ echo $?
2
```

הנחיות עבור חלק ב' (אפשר כקובץ נפרד או לשלב אותה בחלק הראשון)

מוטיבציה: תכנית המריצה תכניות אחרות, בודקת ונותנת להם ציון כתבו תכנית המקבלת שם ו path של קובץ קונפיגורציה כפרמטר ל main. קובץ הקונפיגורציה מכיל 3 שורות: שורה 1: מיקום של תיקייה המכילה תתי תקיות (כל תתי תיקייה זה בעצם שם משתמש) המכילים 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 להכנס לכל תתי התקיות שבתוך התיקייה משורה 1, לקמפל את קובץ ה C שמופיע בתיקייה ולהריץ את קובץ הריצה שנוצר כתוצאה מהקימפול עם הקלט שמוכל במיקום משורה 2 (התכנית שתריצו קולטת מ stdin ומדפיסה ל stdout ולכן תשתמשו ב i/o redirection).

את הפלט של התכנית יש להשוות עם הפלט הרצוי שמיקומו מגיע משורה 3 בעזרת התכנית comp.out שממשתם בתרגיל הקודם (פשוט תריצו את הקובץ comp.out ותתנו לו כפרמטרים לmain את המיקום של הקובץ משורה 3 ואת הפלט של התכנית שהרצתם).

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 או מאה). יש לרשום את התו " " בין שם המשתמש לבין ציונו.

דוגמא לתוכן קובץ results.csv :

elmalea,100
Dan,0
Moshe,100
dolev,100

דוגמא לתוכן קובץ הקונפיגורציה :

/home/ alkobys /ex3/
/home/ alkobys /code/input.txt
/home/ alkobys /code/output.txt

דוגמא לתוכן קובץ קלט (שורה 2) :

1
3 4
4

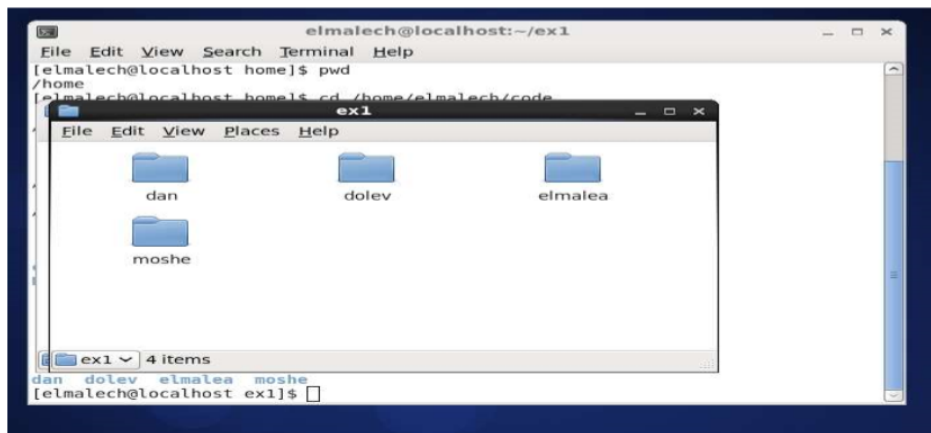
דוגמא לתוכן קובץ פלט (שורה 3) :

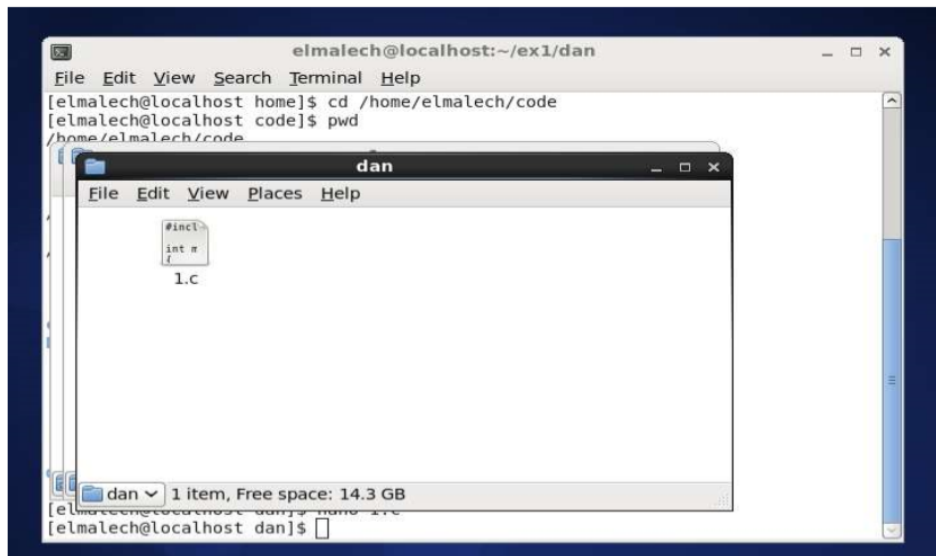
Please enter operation
Please enter two numbers
The sum is 7
Please enter operation
Bye

דוגמא להפעלת קובץ הריצה של התכנית שלכם :

shani@shani-VirtualBox:~\$./a.out /home/alkobys/confFile.txt

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





הערות:

- לא כל התכניות (שהתכנית שלכם צריכה לבדוק) חייבות להתקמפל, אם תכנית לא מתקמפלת צריך לרשום בקובץ ה results.csv את שם המשתמש ולתת לו ציון 0.
- הניחו שאם תכנית מתקמפלת היא רצה וחייבת להסתיים, רק לא בטוח אם הפלט שלה נכון.

חובה להקפיד על ההוראות שרשומות במודל בנוגע לאופן הגשת המטלות

אין להשתמש בפקודות system אלא רק ב-fork, execvp, wait.

בחלק א' חובה להשתמש רק ב-system calls. בחלק ב' אין חובה כזאת.

ניתן להניח ששם התכניות המקומיות של הסטודנטים שהגישו (אותם אתם בודקים) הוא: main.exe

שמות הסטודנטים יהיה זהה לשם התקיה של כל סטודנט.

ניתן להניח שכל שם סטודנט וכל שם תקיה לכל היותר 50 תווים.

ניתן להניח שכל שורה בקובץ הקונפיגורציה היא לכל היותר 50 תווים.

ניתן להניח ששמות תקיות הסטודנטים ושם של הסטודנטים עצמם זהה (עד רמת האותיות הקטנות-גדולות).

ניתן להניח שכל סטודנט קימפל את הקובץ שלו ולכן בתקיה האישית שלו יש כע רק קובץ אחד ושמו main.exe.

ניתן להניח שכל התכניות (של כל היוזרים) כבר מקומפלות ותקינות ויש רק לבדוק את הפלט מול המצופה (לתת 0 או 100).

הקבצים של הסטודנטים כבר מקומפלים וניתן להשתמש בקובץ מחלק א' כקובץ שכבר מקומפל.

יש למחוק כל הקצאות דינמיות, לשחרר את כל קבצי העזר (למעט גליון הציונים).

עבור חלק ב': במסגרת צילומי המסך יש לצרף גם את צילום המסך של ציוני הסטודנטים (יש להראות לפחות 3 סטודנטים) וצילום קובץ הקונפיגורציה. כמו כן, יש לצרף את קוד המקור.

עבור חלק א': לצרף את קוד המקור, ושתי הרצות של התכנית: פעם אחת עם הקבצים זחים ופעם אחת עם קבצים שונים (יש לצלם את הטרימינל אחרי פקודת echo \$?)

את שני החלקים יש לאחד לקובץ RAR או ZIP יחיד.

Yaniv Hajaj פתרון במחשבי האוניברסיטה LINUX

דוגמה לקבצים שהם כן זהים

```

37 write(2, "wrong path number two\n", 23); /* wrong path*/
38 close(fdin1);
39 exit(1); /* return 1 not the same files*/
40 }
41
42 int file_1_size;
43 int file_2_size;
44 do{
45
46     file_1_size = read(fdin1, ch1, 1); /*-l=error @=end file l=single char was read*/
47     file_2_size = read(fdin2, ch2, 1); /*-l=error @=end file l=single char was read*/
48     if(file_1_size==0 || file_2_size==0){ /* error cannot read from file 1 or 2*/
49         close(fdin1);
50         close(fdin2);
51         write(2, "cannot read the from one of the files file\n", 22);
52         exit(1);
53     }
54     /* read 1 char*/
55     //printf("file 1 size:%d\n", file_1_size);
56     //printf("file 1 char:%c\n", ch1[0]);
57     /* read 2 char*/
58     //printf("file 2 size:%d\n", file_2_size);
59     //printf("file 2 char:%c\n", ch2[0]);
60     if(ch1[0]==ch2[0]){
61         close(fdin1);
62         close(fdin2);
63         write(2, "files dont match !\n", 21); /*char dont match*/
64         exit(1); /* return 1 not the same files*/
65     }
66     if(file_1_size==0 || file_2_size==0){
67         break;
68     }
69     while(ch1[0]==ch2[0]);
70     close(fdin1);
71     close(fdin2);
72     write(2, "files ok !!\n", 13); /*char dont match*/
73     exit(2); /* return 1 not the same files*/
74 }
75
76

```

```

bash-4.2$ ./comp.out /u/e2020/hajajya/Desktop/OS/1.txt /u/e2020/hajajya/Desktop/OS/2.txt
files ok !!
bash-4.2$ echo $?
2
bash-4.2$

```

דוגמה לקבצים שהם לא זהים

```

37 write(2, "wrong path number two\n", 23); /* wrong path*/
38 close(fdin1);
39 exit(1); /* return 1 not the same files*/
40 }
41
42 int file_1_size;
43 int file_2_size;
44 do{
45
46     file_1_size = read(fdin1, ch1, 1); /*-l=error @=end file l=single char was read*/
47     file_2_size = read(fdin2, ch2, 1); /*-l=error @=end file l=single char was read*/
48     if(file_1_size==0 || file_2_size==0){ /* error cannot read from file 1 or 2*/
49         close(fdin1);
50         close(fdin2);
51         write(2, "cannot read the from one of the files file\n", 22);
52         exit(1);
53     }
54     /* read 1 char*/
55     //printf("file 1 size:%d\n", file_1_size);
56     //printf("file 1 char:%c\n", ch1[0]);
57     /* read 2 char*/
58     //printf("file 2 size:%d\n", file_2_size);
59     //printf("file 2 char:%c\n", ch2[0]);
60     if(ch1[0]==ch2[0]){
61         close(fdin1);
62         close(fdin2);
63         write(2, "files dont match !\n", 21); /*char dont match*/
64         exit(1); /* return 1 not the same files*/
65     }
66     if(file_1_size==0 || file_2_size==0){
67         break;
68     }
69     while(ch1[0]==ch2[0]);
70     close(fdin1);
71     close(fdin2);
72     write(2, "files ok !!\n", 13); /*char dont match*/
73     exit(2); /* return 1 not the same files*/
74 }
75
76

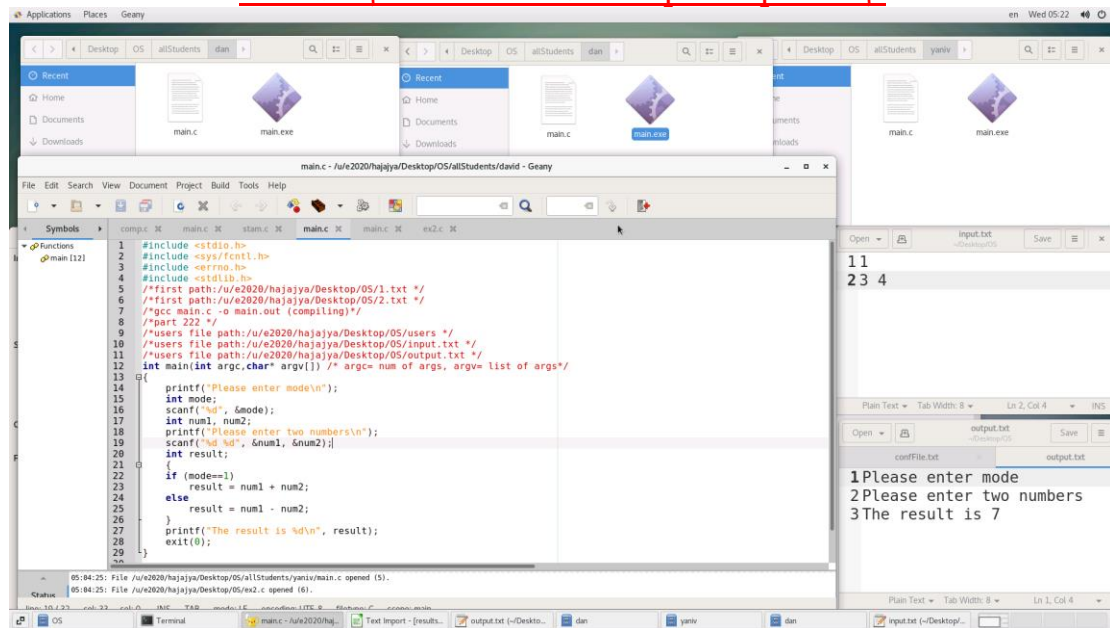
```

```

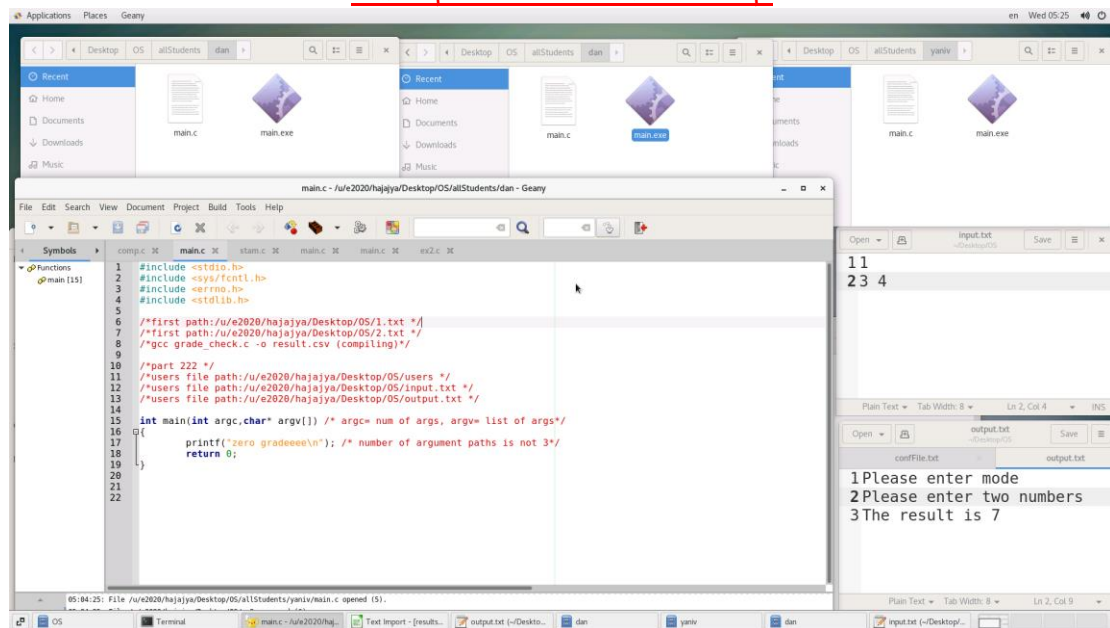
bash-4.2$ ./comp.out /u/e2020/hajajya/Desktop/OS/1.txt /u/e2020/hajajya/Desktop/OS/2.txt
files ok !!
bash-4.2$ echo $?
2
bash-4.2$ ./comp.out /u/e2020/hajajya/Desktop/OS/1.txt /u/e2020/hajajya/Desktop/OS/2.txt
files dont match !
bash-4.2$ echo $?
1
bash-4.2$

```

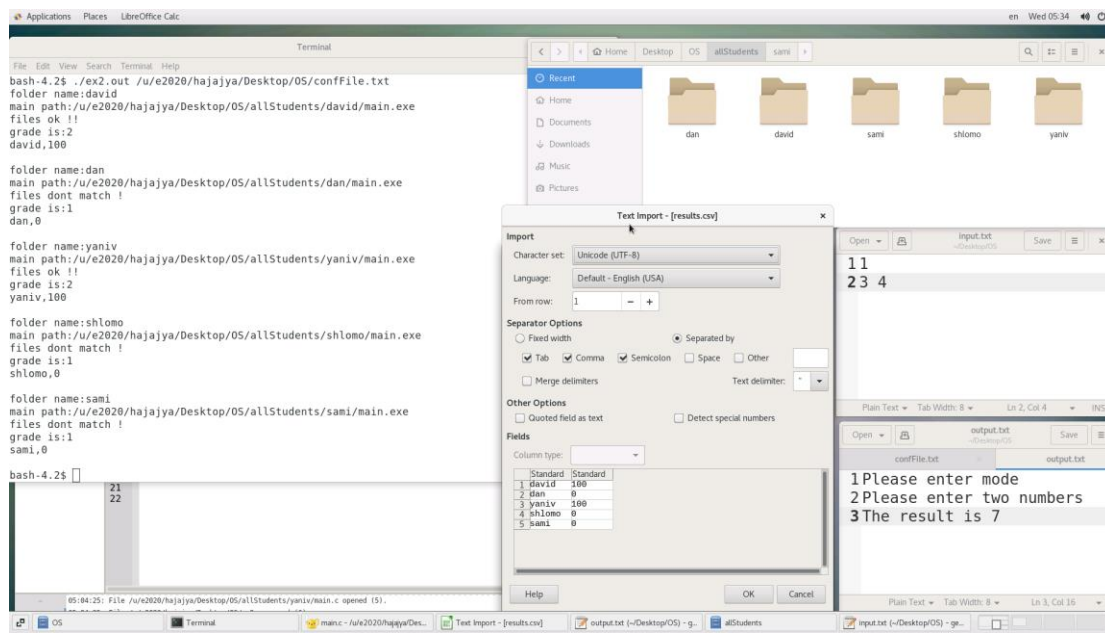
קבצי input וoutput של סטודנטים שקיבלו מאה



קבצים של סטודנטים שקיבלו 0



תוצאת טבלה למטה מראה איזה סטודנטים קיבלו 100 ואיזה קיבלו 0 כתלות בפונקציית הC שהם כתבו (דוד ויניב עם 100, דן שלמה וסמי עם 0)



עוד הרצה כזו

