

תרגיל בית מס. 2

מטרת התרגיל

כתבו תכנית המפעילה פונקציה נתונה על השמות של כל הקבצים המצויים בעץ התיקיות תחת תיקיה נתונה. יש לבחון את התיקיה הנתונה, לקרוא לפונקציה הנתונה עם כל אחד מהשמות של הקבצים שבה, ולהמשיך באופן רקורסיבי על כל התיקיות שבה ולסרוק גם את אלה, עד שכל הקבצים שבעץ (תיקיות וקבצים אחרים) טופלו באמצעות הפונקציה.

התכנית מקבלת כפרמטר את המסלול (path) אל התיקיה ממנה יש להתחיל את הסריקה. בהעדר פרמטר, הכוונה לסריקה של התיקיה הנוכחית.

מה צריך לעשות?

יש לכתוב פונקציה בשם `scantree`, שחתימתה מופיעה ב-`header file` ששמו `scantree.h` שבאתר הקורס. הפונקציה מקבלת שני פרמטרים:

1. שם של תיקיה (path)
2. `מָחָוּן` (pointer) לפונקציה שנקרא לה כאן `Myfunc`. (מחווין לפונקציה אינו נושא את שמה של הפונקציה, רק את מקום הקוד שלה בזכרון, כך ששמה האמיתי לא משנה.) המימוש של `Myfunc` נתון אף הוא (ראו להלן).

תפקיד הפונקציה `scantree` הוא לסרוק את עץ התיקיות תחת `path`. עליה לשלוף את שמות הקבצים שבתיקיה אחד אחד, ולקרוא ל-`Myfunc` עם שם הקובץ בכל פעם. (שימו לב ששם הקובץ צריך להיות מסלול אבסולוטי, הכולל את שמות כל התיקיות מהשורש עד הקובץ האמור.) אם אחד הקבצים הללו הוא בעצמו תיקיה, אזי חוץ מאשר לקרוא ל-`Myfunc` עם שם התיקיה, יש לעבור על כל הקבצים שבתוך התיקיה ולקרוא ל-`Myfunc` גם עליהם, ושוב אם גם ביניהם יש תיקיות, וכך הלאה באופן רקורסיבי, עד שכל הקבצים תחת ה-`path` טופלו באמצעות `Myfunc`.

כדי להמנע מלולאות בסריקה, אין לעקוב אחרי קישורים סימבוליים: יש להתייחס אליהם כקבצים בפני עצמם, ולא לקבצים אליהם הם מכוונים. כמו כן, אין להתייחס לתיקיות ששמן '...' ו-'...' כי גם הן יוצרות לולאות בסריקה.

ל-`Myfunc` יש פרמטר יחיד מסוג `char*`, דרכו מעבירים את שם הקובץ שצריך טיפול.

איך להריץ ולבדוק?

הפונקציה `Myfunc` עצמה מוגדרת בקובץ שמגדיר גם את ה-`main` של המערכת, ושגרסה מקומפלט שלו נמצאת באתר אף היא תחת השם `main.o`. כדי להריץ את מה שכתבתם, יש לקמפל את הקוד שלכם ולחבר אותו עם `main.o` כך:

```
prompt $ cc -o scantree scantree.c main.o
```

כדי לבדוק את התכנית יש להריץ אותה, כך:

```
prompt $ ./scantree <ריק או תיקיה או ריק>
```

(כל זאת בהנחות שהקובץ שכתבתם נקרא `scantree.c`, שהטרמינל שלכם עונה `prompt $` אחריו כל פקודה.) מובן שאפשר לבדוק את התכנית על כל תיקיה (ראו דוגמה בעמוד הבא), אבל יהיה מעניין להריץ על התיקיות `/bin` ו-`/dev`.

הפלט שמתקבל יכול להיראות כך :

```
prompt $ ./scantree
/home/rsivan/Dropbox/AdvancedProg/HW/1-Scantree/main.c
/home/rsivan/Dropbox/AdvancedProg/HW/1-Scantree/scantree.c
/home/rsivan/Dropbox/AdvancedProg/HW/1-Scantree/main.o
/home/rsivan/Dropbox/AdvancedProg/HW/1-Scantree/scantree
/home/rsivan/Dropbox/AdvancedProg/HW/1-Scantree/scantree.h
/home/rsivan/Dropbox/AdvancedProg/HW/1-Scantree
regular files    =      5, 83.33 %
directories      =      1, 16.67 %
block special   =      0,  0.00 %
char special    =      0,  0.00 %
FIFOs           =      0,  0.00 %
symbolic links  =      0,  0.00 %
sockets         =      0,  0.00 %
total bytes     =    24442
total blocks    =      112
prompt $
```

להלן התוכן של `scantree.h`, הקובץ שעליכם לכלול בקוד שלכם (באמצעות `#include`), שמכיל את החתימות של `scantree` ושל `Myfunc` :

```
#ifndef _SCANTREE_H
#define _SCANTREE_H
/*
 * function type that is called for each filename
 */
typedef int Myfunc( const char * );
/*
 * function for scanning a directory tree
 */
int scantree( const char *path, Myfunc *func );
#endif
```

את התרגיל יש לבצע בבית על מחשב אישי או על מחשב במעבדה במכללה. יש לקבץ את קוד התכנית ואת הפלט שלה באמצעות ZIP, RAR או תוכנה דומה, ולהעלות את המקבץ באמצעות moodle לאתר הקורס. נא ציינו את שמכם (בעברית) ואת מספר הזהות שלכם בגוף ההגשה.

את התרגיל יש להגיש ביחידים (לא בזוגות!) עד יום ששי, 29 בנובמבר.

בהצלחה!