

תרגיל בית מס. 1

מטרת התרגיל: כתבו תכנית המייצרת פלט הדומה ככל האפשר לזה של `wc`. הפקודה, ששמה הוא קיצור של `word count`, למעשה עושה קצת יותר מאשר רק לספור מילים: היא סופרת את מספר האותיות, המילים והשורות שיש בקובצי הקלט שלה. (הריצו את הפקודה על כמה קבצים, אפשר גם כמה ביחד, וראו מה היא יודעת לעשות).

התכנית יכולה לקבל **אפס** או יותר פרמטרים, שכל אחת מהן היא `path` (יחסי או מוחלט) של קובץ. אם התכנית נקראת ללא פרמטרים, היא מפיקה את הסטטיסטיקות האמורות על תוכנו של **הקלט הסטנדרטי**.

עבור **כל פרמטר**, על התכנית להדפיס את הדברים הבאים:

- מספר **התווים** הכולל שבקובץ.
- מספר **המילים** שבקובץ (ראו להלן מהי ההגדרה של מילה).
- מספר **השורות** שבקובץ (ראו להלן מהי הגדרת שורה).

פורמט הפלט הוא פשוט:

```
rsivan@ubuntu:~$ wc memtest0.s examples.desktop try try.c
 47  136 1097 memtest0.s
240  569 8980 examples.desktop
 15  125 8752 try
  7   17   97 try.c
309  847 18926 total
rsivan@ubuntu:~$
```

כל שורה בפלט מייצגת סטטיסטיקה של אחד מקובצי הקלט. בכל שורה שלושה מספרים, שאחריהם שם הקובץ בו מדובר. המספר הראשון הוא מספר השורות, השני הוא מספר המילים, והאחרון הוא מספר התווים. אם יש יותר מפרמטר אחד, נוספת שורה בסוף המתוויגת `total`, ומכילה את סכום כל הסטטיסטיקות של כל הפרמטרים. שימו לב שקובצי הקלט יכולים להיות קובצי טקסט (כמו `try.c`) וגם קבצים בינאריים (כמו `try`).

אם פרמטר מציין תיקיה, ולא קובץ, אזי הסטטיסטיקות כולן 0 ונוספת הודעה המציינת שהפרמטר הוא תיקיה:

```
rsivan@ubuntu:~$ wc Desktop Dropbox
wc: Desktop: Is a directory
 0    0    0 Desktop
wc: Dropbox: Is a directory
 0    0    0 Dropbox
 0    0    0 total
rsivan@ubuntu:~$
```

ספירת **התוים** היא קלה: כל byte בקובץ נחשב תו, גם אם הקובץ הוא בינארי (ו-byte עשוי להיות, למשל, 8 הביטים האמצעיים של מספר בפורמט int, שתופס 4 בתים סך הכל).

הגדרת **שורה** פשוטה אף היא: שורה היא רצף של אפס או יותר תוים שונים מ-'\\n' שמסתיים בתו '\\n'. שני תו '\\n' רצופים מכילים שורה (ריקה) ביניהם שיש לספור. כמו כן, אם התו האחרון בקובץ אינו '\\n', אזי התוים האחרונים בקובץ (מה- '\\n' האחרון ועד סוף הקובץ) מהווים שורה גם הם.

הכי מסובכת היא הגדרת **מילה**: כל רצף של תוים שאיננו אחד מהתוים ה-"לבנים", אלה שאינם משאירים דיו על הנייר. התוים ה-"לבנים" כוללים רווח (' '), טאב ('\\t'), ועוד כמה control characters שתפקידם לנהל את המדפסת (במקום להשאיר סימן מודפס). הפונקציה isspace() מחזירה ערך שונה מאפס עבור כל תו "לבן", ואתם מומרים לחפש אותה בתיעוד (הפקודה man) ולבדוק כיצד משתמשים בה. (אצלי היא מחייבת הכללת הקובץ ctype.h, אבל אצלכם זה עשוי להיות אחרת.)

הערות:

1. נצלו את התיעוד המובנה בתוך Unix: הפקודה man (קיצור של Manual) מספקת הסברים על כל פקודה, קריאת מערכת, פונקציה ספריה או סתם נושא שקשור בבצורה מטלות ב-Unix. פשוט קראו ל-"<שם> man" על מנת לקבל הסבר על המטרה ודרך השימוש ב-"<שם>".

man כולל דפי הסבר גם לפקודות (כאלו שניתן להפעיל מה-shell, כמו ls), קריאות מערכת (כמו open או write) ופונקציות של C (כמו printf או isspace). קורה שלכם מהן שם זהה, למשל chmod (שהיא גם פקודה וגם קריאת מערכת) או dialog, שהיא גם פקודה וגם פונקציה של C. במקרה זה אפשר לכוון את man לפרק המתאים ע"י הדגל -S:

```
$ man -S1 chmod - chmod הפקודה
$ man -S2 chmod - chmod קריאת המערכת

$ man -S1 dialog - dialog הפקודה
$ man -S3 dialog - dialog פונקציה C
```

2. במעבדות ראיתי שאנשים לא תמיד מעבירים פרמטרים ל-main שהם כותבים, ומעדיפים לקבוע בקוד פרמטרים (לפעמים באמצעות #define). לעיתים רחוקות זו ההתנהגות המבוקשת, ולעולם לא בקורס הזה. אנא העבירו את הפרמטרים לתכניות שאתם כותבים משורת הפקודה, תוך ניצול המנגנון הבנוי של (int argc, char *argv[]) של main לשם כך. את התרגיל יש לבצע ביחידים (לא בזוגות!) בבית על מחשב אישי או על מחשב במעבדה במכללה. יש לקבץ את קוד התכנית ואת הפלט שלה באמצעות ZIP, RAR או תוכנה דומה. שם הקובץ שתיצרו צריך להיות בפורמט הבא:

Israel_Israeli_123456789_Ex1.zip

כלומר: השם באנגלית, מספר הזהות, והמחרוזת "Ex1" עם קו תחתון ביניהם. את הקובץ (היחיד) יש להעלות באמצעות moodle לאתר הקורס. נא ציינו, בנוסף, את שמכם (בעברית) ואת מספר הזהות שלכם בגוף ההגשה בקובץ ReadMe.txt. את התרגיל יש להגיש עד יום ששי, 15 בנובמבר.

בהצלחה!