

מערכות הפעלה – תרגיל 2

(חלק א להגשה בהתאם לתיבת ההגשה עבור חלק א)
(חלק ב להגשה בהתאם לתיבת ההגשה עבור חלק ב)

מצורף הקוד עבור התוכנית primecount. התוכנית מקבלת טווח של מספרים (ברירת מחדל מ-1 עד 100) ומדפיסה כמה מספרים ראשוניים נמצאים בטווח וככל שישנם כאלה, היא גם מדפיסה את המספרים הראשוניים (מהקטן לגדול).

בתרגיל הנכם נדרשים לכתוב קוד מעודכן עבור התוכנית primecount:

- בחלק א בדיקת הראשוניות תעשה על ידי מספר תהליכים בו זמנית (או במקביל) תוך שימוש בזיכרון משותף (mmap). בחלק זה חלוקת העבודה בין התהליכים תהיה קבועה מראש.
- בחלק ב בדיקת הראשוניות תעשה שימוש במספר חוטי חישוב (threads) בו זמנית. בחלק זה חלוקת העבודה בין חוטי החישוב תהיה דינמית.

הכנה

- קראו בעיון את התוכנית primecount.c.
- הריצו את התוכנית.
- שנו את הקוד כך שהתוכנית תאפשר את האופציה '-n' (בנוסף לאופציות המקוריות '-l' ו'-u'). ברירת המחדל לאופציה זו הינה 10. האופציה '-n' קובעת כמה מספרים, לכל היותר, יש להדפיס בפלט. למשל, אם הטווח הינו מ-1 עד 100 וערכו של n הינו חמש אזי יש להדפיס שנמצאו 25 מספרים ראשוניים, אך יש להדפיס רק חמישה מספרים ראשוניים (2,3,5,7,11) ולא את כל 25 המספרים הראשוניים שנמצאו.

חלק א

בחלק זה עליכם לערוך את התוכנית primecount.c לאחר שלב ההכנה (הוספת האופציה -n).

שימו לב כי בחלק זה בדיקת הראשוניות תעשה על ידי מספר תהליכים בו זמנית (או במקביל) תוך שימוש בזיכרון משותף (mmap). בחלק זה, כאמור, חלוקת העבודה בין התהליכים תהיה קבועה מראש.

- ראשית, הוסיפו לתוכנית את האופציה '-p' (בנוסף לאופציות המקוריות '-l' ו'-u' ולאופציה '-n' משלב ההכנה). ברירת המחדל לאופציה זו הינה 4. האופציה '-p' קובעת כמה תהליכים ('בנים') יש ליצור לצורך בדיקת הראשוניות.
- הוסיפו לתוכנית יצירה של p בנים, כך שהאבא והבנים משתפים את המערך flagarr.
- לאחר יצירת הבנים:
 - על האבא להמתין (wait) עד שכל הבנים 'יצאו'.
 - על הבן הראשון לעדכן את flagarr[0], flagarr[p], flagarr[2p], flagarr[3p], ...
 - על הבן השני לעדכן את flagarr[1], flagarr[p+1], flagarr[2p+1], ...
 - על הבן השלישי (ככל שישנו) לעדכן את flagarr[2], flagarr[p+2], ...
 - וכן הלאה
- לאחר שהבנים סיימו את הבדיקה המוטלת עליהם, עליהם 'לצאת'.
- לאבא נותר 'לספור' כמה ראשוניים נמצאו ולהדפיס את הפלט הנדרש.

חלק ב

בחלק זה עליכם לערוך את התוכנית `primecount.c` לאחר שלב ההכנה (הוספת האופציה `-n`) ולא את התוכנית שכתבתם בחלק א.

שימו לב כי בחלק זה בדיקת הראשוניות תעשה על ידי מספר חוטי חישוב בו זמנית. כאמור, בעוד שבחלק א חלוקת האחריות לעדכון התאים במערך `flagarr` בין הבנים השונים הייתה 'קשיחה', בחלק זה חלוקת העבודה בין חוטי החישוב לא תהיה קבועה מראש.

- ראשית, הוסיפו לתוכנית את האופציה `'-t'` (בנוסף לאופציות המקוריות `'-l'` ו-`'-u'` ולאופציה `'-n'` משלב ההכנה). ברירת המחדל לאופציה זו הינה 4. האופציה `'-t'` קובעת כמה חוטי חישוב יש ליצור לצורך בדיקת הראשוניות.
- הוסיפו לתוכנית יצירה של `t` חוטי חישוב.
- כל אחד מחוטי החישוב שנוצרו יפעל כדלהלן:
 - יאתר מספר פנוי (`num`) מהטווח שטרם נבדק (או נמצא בבדיקה) על ידי אחד מחוטי החישוב. איתור המספר הפנוי מבוצע על ידי "תפיסת" המספר הבא לחישוב. שימו לב כי עליכם לוודא שמספר לא ייתפס מספר פעמים על ידי מספר חוטי חישוב.
 - יבדוק את ראשוניות המספר שאותר ויעדכן את התא הרלוונטי במערך `flagarr`.
- יש לחזור על הפעולה לעיל שוב ושוב עד אשר כל המספרים בטווח נבדקו (או נמצאים בבדיקה על ידי חוט חישוב אחר).
- לאחר שכל חוטי החישוב שנוצרו יסיימו, על חוט החישוב המקורי להדפיס את הפלט הנדרש