

תרגיל תכנות מערכות הפעלה סמסטר 2019 ב -- תהליכים.

התוכנית תכתב בשפת C ותרוץ על Linux. הגשה דרך moodle.

יש לכתוב שתי גרסאות של תוכנית הבודקת אם פתרון לחידת Sudoku הוא חוקי. בפתרון זה יעשה שימוש במספר תהליכים -- processes. (להבדיל מ- threads).

חידת Sudoku

בחידה מסוג זה נתונה מטריצה בת 9 שורות ו- 9 עמודות כאשר בחלק מהכניסות רשומות ספרות (בין 1 ל- 9) וחלק מהכניסות ריקות. יש לרשום ספרות בכניסות הריקות כך שיתקיימו התנאים הבאים.

בכל אחת מהשורות יופיעו כל הספרות בין 1 ל- 9

בכל אחת מהעמודות יופיעו כל הספרות

בכל אחת מתשעת -"תתי המטריצה" בגודל 3×3 יופיעו כל הספרות

ראו דוגמא ב- <https://en.wikipedia.org/wiki/Sudoku>

הממשק לתוכנית

התוכניות שיש לכתוב קוראות מקובץ קלט (אחד או יותר) מטריצה של ספרות (מטריצה אחת בכל קובץ) ובודקות לכל מטריצה אם היא מהווה פתרון חוקי לחידת Sudoku. אם המטריצה מהווה פתרון חוקי אזי, התוכנית מדפיסה ל- standard output את ההודעה "FILENAME is legal" ואחרת היא מדפיסה "FILENAME is not legal" (כאשר FILENAME הינו שם הקובץ בו הייתה שמורה המטריצה).

שמות קבצי הקלט ימסרו לתוכנית כ- command line argument (ב- argv). אם אין command line argument אז תקרא מטריצה בודדת מה- standard input.

קובץ קלט כולל 81 ספרות מופרדות ע"י white space (רווחים, newlines או tabs). בקלט מופיעות הספרות של השורה הראשונה (משמאל לימין), לאחריון הספרות של השורה השנייה וכן הלאה עד לספרות של השורה התשיעית.

לנוחיותכם מצורף קובץ קלט לדוגמא הכולל את הפתרון המופיע בקישור הנ"ל.

הנחיות לכתובת התוכנית

התהליך הראשי יצור שלושה תהליכים ילדים ע"י קריאה ל- fork(). לאחר יצירת שלושת הילדים, התהליך הראשי יקרא את קבצי הקלט (אחד אחד) ויעביר את התוכן (המטריצה) לתהליכים הילדים. בדיקת תקינות כל מטריצה תתחלק בין התהליכים הילדים. אחד משלושת התהליכים הילדים יהיה אחראי

על בדיקת תקינות כל השורות (שורה היא תקינה אם היא מכילה את כל הספרות בין 1 ל-9). תהליך ילד שני יהיה אחראי לבדיקת תקינות כל העמודות ותהליך הילד השלישי יבדוק את תקינות כל "תתי המטריצות". שלושת הילדים ידווחו לתהליך הראשי על תוצאות הבדיקות שעשו והתהליך הראשי ידפיס לפלט הודעה מתאימה.

תקשורת בין התהליכים.

התהליך הראשי צריך לתקשר עם הילדים שלו: הילדים צריכים לדעת מה הפתרון המוצע לחידת ה-Sudoku (81 ספרות) כדי שיוכלו לבדוק את תקינותו. התהליך הראשי יודע מה הפתרון כי הוא קורא אותו מהקלט. כל אחד מהילדים צריך לדווח להורה (התהליך הראשי) את תוצאת הבדיקה שעשה ("תקין" או לא "תקין"). ישנן דרכים רבות לממש תקשורת בין תהליכים. בתרגיל זה נעשה שימוש ב-pipes ושימוש בזכרון משותף ע"י קריאת המערכת mmap.

גרסה ראשונה: שימוש ב-execve ו-pipes.

כל אחד מהילדים ישתמש בקריאת המערכת execve כדי להריץ קובץ הרצה עם תוכנית שעושה את הבדיקה המוטלת ע"י הילד (בדיקת תקינות השורות, תקינות העמודות או תקינות תתי המטריצות). כל אחד מהילדים יכול להריץ קובץ הרצה שונה, או לחילופין שלושתם יכולים להריץ את אותו קובץ הרצה כאשר משתמשים ב-command line argument כדי לציין את סוג הבדיקה שיש לבצע (בדיקת השורות, העמודות או תתי המטריצות).

(במקום לקרוא ל-execve ישירות ניתן להשתמש באחת מפונקציות הספרייה שמשתמשות ב-execve, ראו <http://man7.org/linux/man-pages/man3/execve.3.html>).

התהליך הראשי יתקשר עם הילדים תוך שימוש ב-pipes. עבור כל ילד, התהליך הראשי (ההורה) יצור pipe. ההורה יכתוב את המטריצה (הפתרון המוצע לחידת ה-Sudoku) ל-pipe והילד יקרא מה-pipe. לקבלת תוצאות הבדיקה ניתן להסתפק ב-pipe נוסף אחד בלבד שלתוכו יכתוב כל אחד מהילדים את תוצאת הבדיקה שעשה. כל ילד יכתוב בית בודד ל-pipe (שערכו 1 אם "תקין" ו-0 אם "לא תקין") והתהליך הראשי יקרא מתוכו את שלושת הבתים. לחילופין ניתן שהתהליך הראשי יצור pipe נוסף עבור כל אחד מהילדים שישמש את הילד לכתובת תוצאת הבדיקה שעשה.

בכל מקרה, רצוי לתת לכל ילד לקרוא מ-0 file descriptor (STDIN) ולכתוב ל-1 file descriptor (STDOUT).

גרסה שנייה : שימוש ב-mmap.

התהליך הראשי יתקשר עם הילדים ע"י שימוש בזיכרון משותף לו ולילדיו. התהליך הראשי ישתמש בקריאת המערכת mmap כדי ליצור את אזור הזיכרון המשותף. הקריאה ל-mmap תיעשה לפני הקריאות ל-fork, כך שהתהליכים הילדים יקבלו בירושה את המיפוי. מכיוון שקריאה ל-execve מבטל את השיתוף שנוצר בקריאה ל-mmap אין צורך לעשות שימוש בקריאת המערכת execve.

תיעוד של mmap נמצא ב-URL:

<http://man7.org/linux/man-pages/man2/mmap.2.html>

אזור הזכרון המשותף יכול להכיל מבנה נתונים דומה לזה:

```
struct shared {  
    int solution[81];  
    int status[3];  
};
```

כאשר status הוא מערך שלתוכו יכתבו שלושת הילדים את תוצאות הבדיקה שלהם (כל ילד כותב רק לכניסה במערך המיועדת לו. אם יותר מתהליך אחד היה כותב לאותו מקום בזכרון היה צורך באמצעי סנכרון כדי לתאם ביניהם).

ניתן לעשות שימוש ב-busy waiting לסנכרון בין התהליך הראשי ובין התהליכים הילדים.

פרק 10