

מכללה אקדמית הדסה

החוג למדעי המחשב

תרגיל #7: תכנות מערכת ומבוא לתכנות מקבילי-- תהליכונים

תכנית a: מילוי מערך של ראשונים

נחזור למשימה המוכרת של בעל מערך שמטרתו למלא מערך ממזין, ויצרנים המייצרים מספרים ראשוניים עבורו. הפעם נשתמש בתהליכונים.

תהליכון ראשי: בעל המערך

הגדירו משתנה גלובלי: מערך של מספרים שלמים בן `ARR_SIZE` (קבעוהו להיות 101) תאים. התהליכון הראשי יאפס את המערך. תא #0 במערך ישמש 'כמנעול' (כמו בתרגיל הז"מ). לתוכו התהליכון יכניס את הערך אפס [הוא נעול].

התהליכון הראשי ייצר שלושה תהליכונים משנה שיגרילו ערכים ראשוניים, ויוסיפו אותם למערך (כך שהוא יהיה ממזין).

אחרי יצירת שלושת התהליכונים הוא יפתח להם את המנעול: יכניס לתוכו את הערך אחד. הוא ימתין להם, ועתה הם יסיימו הדבר יעיד שהמערך התמלא (או שהם לא הצליחו למלאו, בדומה למה שתיארנו בעבר). הוא יציג אותו פלט כמו בתרגילים הקודמים, ויסיים.

שלושה תהליכונים משנה: יצרני מספרים ראשוניים

כל תהליכון יתנהל בלולאה. בכל סיבוב בלולאה התהליכון מייצר מספר ראשוני, מנסה להוסיף אותו למערך, וסופר כמה ערכים הוא הוסיף למערך. (כמו בעבר נקבע שאם התהליכון נכשל מאה פעמים ברציפות בהוספת ערך חדש למערך הוא מסיים)

כדי לבצע את הבדיקה+ההוספה: התהליכון פונה לתא #0 (המנעול) כל עוד ערכו 0 (כלומר הוא 'נעול') הוא ממתיך. עת ערכו הוא אחד (הוא 'פותח') הוא משנה את ערכו לאפס (נועל אותו), ובכך זוכה בזכות לסרוק את המערך לבדו, ולהוסיף את המספר שלו. לבסוף הוא 'פותח' את המנעול: מכניס לו את הערך 1.

אם התהליכון מצא שכל תאי המערך מלאים (או הוא נכשל מאה פעמים ברציפות בהוספה) הוא מציג אותו פלט כמו בעבר (כמה ערכים הוא הוסיף), ובכך הוא מסיים.

את `srand(17)` זמנו רק פעם יחידה, בתהליכון הראשי.

הערה: הסבירו בקובץ ה: `README` אילו מצבי מרוץ קיימים בתכנית.

תכנית b שרת מ.מ.מ ושרת פרוק לגורמים

כתבו את התכנית הבאה: תהליכון הורה מוליד שלושה ילדים, שומר במערך גלובלי את המזהים שלהם, שלחם, ומסיים.

שלושת הילדים הם:

א. שרת ראשוניות: רץ בלולאה. בכל סיבוב בלולאה הוא:

(א) הולך לשון עד קבלת הסיגנל `SIGUSR1`.

(ב) עת מתעורר בשל קבלת הסיגנל הנ"ל, שולף את הנתונים שמחכים לו במערך גלובלי

ייעודי לו (בן מאה תאים), בודק את ראשוניותם, שם את תוצאת הבדיקה במערך שני, כך

שכך תא במערך השני יכיל את הערך 1 או 0, אם הנתון במערך הראשון היה ראשוני/פריק,

ושולח את הסיגנל `SIGUSR1` לתהליך ה: `front end` (שקרא מהמשתמש את המספר

שאת הראשוניות שלו יש לחשב).

ב. שרת פלינדרומיות. מתנהל באופן דומה (משתמש ב: SIGUSR2). מכיוון שעליו להחזיר רק ערך יחיד: האם הסדרה מהווה פלינדרום או לא, ניתן לקבוע שערך זה הוא יחזיר באמצעות התא #0 במערך שהועבר לו (תא שיושאר ריק ע"י ה: front end)

ג. תהליך end front: בלולאה אינסופית:
(א) קורא מהמשתמש את הערך q ואחריו מספרים שאת ראשונותם יש לבדוק, או את הערך q ואחריו סדרת מספרים שאת פלינדרומיותה יש לבדוק.
(ב) מאחסן את הקלט במערך הדרוש.
(ג) שולח סיגנל לשרת הדרוש.
(ד) הולך לשון עד קבלת הסיגנל SIGUSR1/2.
(ה) עת מתעורר, שולף את הפלט, ומציג אותו למשתמש.

בתכנית עלולה להיווצר תקלה אם השרת ישלח את הסיגנל לתהליך ה: front end לפני שהאחרון עושה pause. בתור 'טלאי' אפשר שהשרת ימתין מעט (עם sleep או usleep לפני שהוא שולח את התשובה).
התכנית תסתיים ע"י קבלת הסיגנל SIGINT.

הערה: הסבירו בקובץ ה: README אילו מצבי מרוץ קיימים בתכנית.