

מכללה אקדמית הדסה

החוג למדעי המחשב

תרגיל #5: תכנות מערכת ומבוא לתכנות מקבילי---

זיכרון משותף

תכנית a: בינוגו הפור

נחזור למשימה המוכרת של תהליך אחד שהינו בעל לוח בינוגו, ותהליכים אחרים המייצרים מספרים ע"מ לרוקן את הלוח. הפעם הכלי לתקשורת בין תהליכים בו נשתמש יהיה הז"מ.

תכנית א': בעל המערך

מגדירה בזיכרון משותף מערך של מספרים שלמים בן `ARR_SIZE` (קבעוהו להיות אלף) תאים. לתא הראשון בז"מ מכניס התהליך את מספרו, לשלושת התאים הבאים הוא מכניס אפס (שיוסבר מיד), גם לתא החמישי שישמש כמעין 'מנעול' פרימיטיבי, הוא מכניס אפס (הסבר על כך בהמשך). לתוך יתר התאים (995) הוא מגדיל ערכים כמו בתרגילים הקודמים. עתה הוא הולך לשון עד קבלת סיגנל. עתה הוא מקבל סיגנל הוא מתעורר. השרת מציג כמה זמן לקח לו להשלים את המשימה. הוא משחרר את הזיכרון, ומסיים.

זמנו את ('5', '.'). `ftok`

תכנית ב' (ממנה תריצו שלושה עותקים): יצרני מספרים

תכנית זאת תתחבר לז"מ שהקצתה תכנית א'. התכנית תתנהל בלולאה. בכל סיבוב בלולאה התכנית מייצרת מספר, ובודקת האם הוא מופיע בז"מ וניתן למוחקו. אם כן, היא מוחקת אותו.

כדי לבצע את הבדיקה: התכנית פונה לתא החמישי (המנעול) כל עוד ערכו 1 (כלומר הוא 'נעול') היא ממתינה. עת ערכו הוא אפס (הוא 'פתוח') היא משנה את ערכו לאחד (נועלת אותו), ובכך זוכה בזכות לסרוק את המערך לבדה. כאמור: אם היא מוצאת במערך את הערך שהיא הגדילה אזי היא מוחקת אותו. לבסוף היא 'פותחת' את המנעול: מכניסה לו את הערך 0.

אם התכנית מצאה שכל תאי המערך מחוקים היא מציגה כמה ערכים היא הגדילה, וכמה מתוכם זכו למחוק ערכים במערך, היא שולחת סיגנל לבעל המערך, ואז היא מסיימת.

שלושת תאי המערך הראשונים ישמשו אותנו כדי לגרום ליצרנים להתחיל למלא את המערך (פחות או יותר) ביחד: כל יצרן עת מתעורר מכניס 'לתא שלו' מבין השלושה את הערך 1. רק אחרי שהוא מגלה שבכל שלושת התאים של שלושת היצרנים מצוי הערך 1, הוא נכנס ללולאה המרכזית של מחיקת הערכים.

כמו בתרגיל הקודם, נקבע שהיצרנים יקבלו דרך וקטור הארגומנטים את המספר שלהם\ שיהיה גם ה: `seed` שלהם: 1, 2, 3.

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

תכנית b: שרת מ.מ.מ. ושרת פרוק לגורמים ראשוניים

כתבו את שלוש התכניות הבאות:

שרת מ.מ.מ. (מחלק משותף מקסימלי)

עם תחילת ריצתו מקצה ז"מ בן חמישה תאים.

לתוך התא #0 הוא מכניס את ה pid שלו.

דרך התאים #1, #2 הוא יקבל שני מספרים טבעיים שאת המ.מ.מ. שלהם עליו לחשב.

לתא #3 הוא יכניס את תוצאת החישוב.

בתא #4 הוא ימצא את מספר התהליך שלו יש להודיע מהו המ.מ.מ., וישלח לו את הסיגנל SIGUSR1

כדי לסמן לתהליך שהמ.מ.מ. חושב, וממתין בתא #3.

השרת	רץ	בלולאה.	בכל	סיבוב	בלולאה	הוא:
(א)	הולך	לשון	עד	קבלת	הסיגנל	SIGUSR1.
(ב)	עת מתעורר בשל קבלת הסיגנל הנ"ל,	שולף את שני הנתונים שמחכים לו בז",	מחשב את המ.מ.מ.			
	שלהם, שם אותו אחריהם בז"מ,	ושולח את הסיגנל SIGUSR1 לתהליך שמספרו מצוי בתא #4.				

שרת פרוק לגורמים ראשוניים.

מתנהל באופן דומה. נקבע שתאי הז"מ שלו יהיו:

#0 = ה: pid שלו.

#1 = המספר שיש לפרק

#2 עד #20 = תוצאת הפרוק

#21 = מספר התהליך שממתין לתוצאה.

שרת זה יעורר באמצעות הסיגנל SIGUSR2, וזה יהיה גם הסיגנל שהוא יישלח.

תהליך front end

בלולאה

אינסופית:

(א) קורא מהמשתמש את הערך g ואחריו שני מספרים שאת המ.מ.מ. שלהם יש לחשב, או את הערך

d ואחריו מספר טבעי שיש לפרק לגורמים ראשוניים.

(ב) מאחסן את הקלט בז"מ הדרוש.

(ג) שולח סיגנל המתאים לשרת הדרוש.

(ד) הולך לשון עד קבלת הסיגנל SIGUSR2/SIGUSR1.

(ה) עת מתעורר, שולף את הפלט, ומציג אותו למשתמש.

כל תכנית תסתיים ע"י קבלת הסיגנל SIGINT. ב: signal handler ישוחררו קטעי הז"מ שהוקצו.

נקבע של: ftok נעביר את ".", וכן את התווים: בשרת המ.מ.מ. g, בשרת הפרוק: d.