מכללה אקדמית הדסה
החוג למדעי המחשב

תרגיל 4#: תכנות מערכת ומבוא לתכנות מקבילי-- צינור משוים ותור הודעות

בחלק מתרגיל זה נחזור למשימה מתרגיל 3b, כדי להתרשם מההבדלים בין pipe, named pipe, message queue. נקווה שתוכלו להתרשם עד כמה תור הודעות מקל על ביצוע המשימה, בעוד עת התהליכים אינם הורה וילדיו, צינור משוים נעשה מסורבל.

תכנית a – חזרה על תכנית b מתרגיל 3# בינגו הפוך – עם named pipe

כתבו את התכניות הבאות:

תכנית א': בעל לוח הבינגו

מגדירה מערך של מספרים שלמים בן חמשים אלף תאים, ומגרילה לתוכו מספרים טבעיים בתחום אפס עד מאתיים אלף (לא כולל מאתיים אלף). מטרת התכנית היא 'למחוק' את המספרים במערך, באמצעות מספרים שהיא תקבל מתכניות אחרות (שיממשו את מה שבתרגיל הקודם מימשו הילדים). התכנית תקבל באמצעות וקטור הארגומנטים את שמו של ה: named pipe בו היא תעשה שימוש, ואת שמות אלה של הילדים.

נקבע ששמות התורים יהיו: fifo1, fifo2, fifo3, fifo0. (המספרים עבור הלוח יועברו לתכנית זאת ע"י fifo0) את התורים יש להכין טרם שמורצות התכניות. הן רק תפתחנה את התורים הקיימים. (התכניות לא תייצרנה את התורים, ולא תמחקנה אותם).

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

תכנית זאת תקבל באמצעות וקטור הארגומנטים את שמו של התור המשוים של תכנית א' (fifo0), ואת המספר של התהליך הנוכחי (1, 2, 3,...) שיקבע גם את שם קובץ התור דרכו התהליך הנוכחי יקבל את הפידבק האם המספר שהוא שלח לתכנית א' התקבל\נדחה (כל עותק מתכנית זאת שיורץ יקבל שם שונה).

תכנית זאת, בלולאה אינסופית מגרילה מספרים בתחום אפס עד מאתיים אלף (לא כולל מאתיים אלף), את המספרים היא שולחת יחד עם המזהה של השולח (באמצעות הצינור המשוים של תכנית א').

תכנית א', כאמור, מסירה את המספרים מהמערך שלה. תכנית א', בכל סיבוב בלולאה הראשית שלה, קוראת נתון מהצינור המשוים שלה. בודקת האם המספר קיים במערך, ושולחת לתהליך ששלח לה את המספר, על הצינור המשוים של תכנית זאת: 1 = המספר היה במערך (ונמחק), 0 = המספר לא היה במערך.

מכיוון שיצרני המספרים בתכנית זאת אינם ילדיו של ממלא המערך, יותר מורכב יהיה ל'ממלא המערך' להרגם. (הדבר אינו בלתי אפשרי. אייך?). על כן נקבע שסיומם של התהליכים מסוג ב' יהיה עת הם קוראים מהתור שלהם את הערך 1- המסמן שעליהם לסיים. באחריות תהליך א', כמובן, לדאוג לשלוח לכל אחד מהם את הערך.

עת ממלא המערך גומר למלא את המערך הוא מסמן ליצרני המספרים לסיים. כל יצרן מספרים, עת מתבקש לסיים מציג כמה ערכים הוא שלח לבעל המערך, וכמה מתוכם התקבלו ע"י בעל המערך.

בעל המערך מציג כמה זמן לקח לו להשלים את המשימה. כמה מספרים הוא קיבל, וכמה הוא מחק (כלומר מה גודל המערך שלו).

הערות

א. כל יצרן מספרים יקבל דרך וקטור הארגומנטים את שם התור של בעל המערך, ואת המספר שלו (1, 2, 3).
ב. התור של כל יצרן ייקרא fifoX אם X הוא המספר שלו.

ג. כל יצרן יוכל לשלוח לממלא המערך לצד המספר שהוא הגריל גם את X, והממלא יחזיר לו תשובה דרך התור המתאים.
ד. X גם ישמש כ: seed של כל יצרן (אצל בעל הלוח הוא יהיה אפס).
ה. שמות הקבצים: ex4a1.c, ex4a2.c

נגדיר בצורה ברורה את הארגומנטים.

תכנית א' תורץ:

./ex4a1 fifo0 fifo1 fifo2 fifo3

כאשר תכנית א' קוראת מ: fifo0

תכנית ב' תורץ:

./ex4a2 fifo0 1

(ובהתאמה 2 או 3 לעותק השני או השלישי)

<u>הערה נוספת</u>:
כדי למנוע מצב בו היצרן הראשון בלבד 'זוכה' לשלוח מספרים לממלא המערך (טרם ששני היצרנים האחרים בכלל התחילו לרוץ) נקבע כי: כל יצרן בתחילת ריצתו שולח לממלא המערך את מספרו (1, 2, או 3). הממלא אחרי שקרא שלושה מספרים משלושה יצרנים מסיק שכל היצרנים רצים. הוא יישלח להם חזרה מספר שמורה להם: 'עתה אתם רשאים להתחיל להעביר לי מספרים'. כל יצרן, כאמור, שולח לממלא את מספרו, קורא מהממלא מספר, ורק אז פונה ללולאה המרכזית שלו, בה הוא מגריל מספרים, שולחם, ומקבל פידבק.

<u>דוגמא לפלט אפשרי:</u>

התהליך הראשון מבין יצרני המספרים ידפיס

50001 40265

התהליך השני מבין יצרני המספרים ידפיס

49082 4817

התהליך השלישי מבין יצרני המספרים ידפיס

49184 4918

התהליך שמחזיק את המערך ידפיס

18 148267 50000

משמעות:

התהליך הראשון מבין יצרני המספרים הגריל 50001 מספרים, אך מתוכם רק 40265 היו במערך.
התהליך השני מבין יצרני המספרים הגריל 49082 מספרים, אך מתוכם רק 4817 היו במערך.
התהליך השלישי מבין יצרני המספרים הגריל 49184 מספרים, אך מתוכם רק 4918 היו במערך.
לכן, בסה"כ התהליך שמחזיק את המערך קיבל 148267 מספרים, מתוכם 50,000 גרמו למחיקת איבר במערך. הזמן שלקח למחיקת האיברים מהמערך הוא 18 שניות.

תכנית b – חזרה על תכנית b מתרגיל #3 – עם message queue

תכנית זאת חוזרת על המשימה של תכנית a: יש בה תהליך 'בעל לוח בינגו', ושלושה תהליכים (שאינם ילדיו) שהינם 'יצרני מספרים'. אולם בתכנית זאת נעשה שימוש בתור הודעות אחד בלבד שיקצה וישחרר ממלא המערך.

הערות

א. לצורך ייצור המספרים האקראיים השתמשו ב: seed : שיתקבל בוקטור הארגומנטים, יהיה 1, ליצרן המספרים האקראיים הראשון, 2 לשני, 3 לשלישי, 0 לבעל הלוח.

ב. זמנו את ftok("."," '4')

ג. כדי שכל היצרנים יתחילו להעביר מספרים לממלא פחות או יותר במקביל, נקבע כי: ממלא המערך יחזיק מערך בו הוא ישמור את ה pid של היצרנים.
כל יצרן עת מתחיל לרוץ שולח לממלא את המזהה שלו, ומחכה להודעה שהוא יכול להתחיל להעביר מספרים ראשוניים לממלא.
עת הממלא מקבל את ה pid של כל היצרנים, הוא שולח להם הודעה שהם יכולים להתחיל במילוי המערך.
כל יצרן, עת מקבל את ההודעה נכנס ללולאה המרכזית שלו, בה הוא שולח מספרים לממלא.

ד. כדי לאותת ליצרנים על סיום יישלח להם היצרן 1-

ה. שמות הקבצים: ex4b1.c, ex4b2.c

ללא קלט
פלט זהה לפלט של תכנית a.

תכנית c

התרגיל יכלול שני תהליכים שיהוו שרתים, יעבדו בשת"פ, כמעין 'מערכת', ויחזיקו את הנתונים הבאים:

א. **שרת רישום**: שומר מספרי תהליכים 'שהתחברו למערכת'. השרת מייצר תור הודעות.
השרת מאפשר שלוש פעולות:
1. הוספת תהליך חדש.
2. בדיקת האם תהליך רשום למערכת.
2. הסרת תהליך קיים.

עת הוא מקבל הודעה המבקשת להוסיף תהליך הוא בודק האם מספר התהליך המופיע בה קיים אצלו. במידה ומספר התהליך לא קיים, וכן יש מקום במבנה הנתונים (מערך לא ממוין) הוא מוסיף את התהליך.
הוא משיב את התשובה (מבחינת תוכן ההודעה): 0 התהליך הוסף בהצלחה, 1 התהליך לא הוסף שכן הוא קיים כבר, 2 התהליך לא הוסף שכן מ"נ מלא.

עת הוא מקבל הודעה המבקשת לבדוק האם תהליך מוכר לו, הוא בודק האם מספר התהליך שנשלח אליו שמור אצלו במ"נ, ומחזיר: 1 = קיים, 0= לא קיים.

עת הוא מקבל הודעה המבקשת להסיר תהליך, הוא מסיר אותו ממ"נ שלו (ואינו משיב דבר).

שרת הרישום ירוץ לנצח. עת הוא מקבל את הסיגנל SIGINT, הוא פונה ל- signal handler, משחרר את התור, ומסיים.
שם קובץ המקור: ex4c1.c

הערכים שיועברו ל: ftok("."," 'c') כלומר התיקיה הנוכחית והאות c.

ב. **שרת אפליקציות**: מאפשר למי שכבר התחבר למערכת לבצוע פעולות: (א) לבדוק ראשוניות, (ב) לבדוק פלינדרומיות. השרת מייצר תור הודעות.
השרת מאפשר שתי פעולות:
1. בדיקת ראשוניות של מספר.
2. בדיקת פלינדרומיות של מחרוזת.

עת הוא מקבל הודעה המבקשת לבצע פעולה, הוא בודק האם המבקש מוכר לשרת הרישום. במידה וכן, הוא מבצע את הפעולה ומחזיר תשובה (0 או 1), במידה והמבקש לא רשום במערכת הוא מחזיר את הערך -1.

סיום: כמו שרת הרישום.
שם קובץ המקור: ex4c2.c
הערכים שיועברו ל: ('d', ".")ftok כלומר התיקיה הנוכחית והאות d

מעבר לשני אלה ירוצו תהליכי לקוח (הם יורצו רק אחרי ששני התהליכים הקודמים כבר רצים, כלומר הרצת הפרויקט תהיה ע"י שיורצו, ראשית, שתי התכניות של שני השרתים, ורק אח"כ יורצו הלקוחות. אפשר, למשל, לפתוח במקביל כמה טרמינלים, ולהריץ מהם כמה לקוחות במקביל):

א. הלקוח מתחבר לשני תורי ההודעות, ונרשם אצל שרת הרישום (במידה ולא ניתן לרשמו הוא מוציא הודעת שגיאה, ומסיים).

ב. בלולאה:

1. הלקוח קורא מהמשתמש את התו n או s המציינים האם המשתמש רוצה לטפל במספר או במחרוזת. עתה הלקוח קורא את הנתון הדרוש (מספר טבעי או מחרוזת). בין ה: s או n לבין הנתון שיש לבדוק יפריד רווח או \n.
2. הלקוח שולח את ההודעה המתאימה לשרת האפליקציות, ממתין לתשובה, ומציג אותה ע"ג המסך.
3. עת הלקוח קורא את התו e הוא שולח הודעה לשרת הרישום כי יש להסירו מהמערכת, והוא מסיים.

שם קובץ המקור: ex4c3.c

דוגמא לקלט ופלט אצל תכנית הלקוח:

```
n
253
not prime
s
madam
palindrome
s
hello
not palindrome
n
251
prime
e
```