תרגיל בית 3 במערכות הפעלה: חלק ייבש

.1

לא מתקיים mutual exclusion. נראה כעת דוגמא נגדית בה שני התהליכים נמצאים בcritical section שלהם באותו הזמן.

נסמן את pid של החוט הראשון ב m ואת הpid של חוט 2 בn. מתקיים $n \neq m$ כי אלו תהליכים שונים.

נתחיל להריץ את חוט 1 עד שנבצע לו החלפת הקשר כפי שנתאר:

בהרצתו הראשונה יספיק:

- לשנות את x להיות m
- לבדוק את התנאי $(y \,\&\&\, y \,!= \, pid)$, בשלב זה y = 0. ולכן הביטוי הלוגי היינו FALSE ולא נבצע את הcontinue.

כעת נבצע החלפת הקשר בין חוט 1 לחוט 2.

חוט 2 יספיק לבצע:

- משנה את x להיות n
- בודק את התנאי $(y \,\&\&\, y \,!= \, pid)$, בשלב זה y = 0 ולכן הביטוי הלוגי הוא FALSE ולא נבצע continue.
- נשנה את y להיות pid של חוט 2: משמע y = n
- בודק את התנאי: $(x \,\,!= \, pid)$. מתקיים כי x היינו n ולכן התנאי היינו FALSE ולא נכנס לcontinue.
- ונכנסים לcritical section

כעת נבצע החלפת הקשר בין חוט 2 לחוט 1:

כעת נמשיך מהנקודה בה עצרנו:

- בדקנו את התנאי הראשון ולא נכנסנו אליו. כעת נשנה את y להיות pid של חוט 1: כלומר y = m
- נבדוק את התנאי $(x \,\,!= \, pid)$, x היינו n וpid הוא m ולכן ניכנס לתנאי וcontinue:הלולאה אינסופית ונכנס לאטרציה נוספת שלה:
- נשנה את x להיות m.
- נבדוק את התנאי$(y \,\&\&\, y \,!= \, pid)$, בשלב זה y היינו m (m אינו 0 – הpid הזה לא יכול להינתן אל תהליכים כאלו). ולכן התנאי FALSE ולכן לא נבצע continue
- נשנה את y להיות m
- נסדוק את התנאי $(x \,\,!= \, pid)$, x שווה לm ולכן FALSE- ולא ניכנס לcontinue.
- נכנסים לcritical section

ובכך הגענו ששני התהליכים נמצאים בcritical section שלהם באותו הזמן.. ולכן לא מתקיים התנאי של mutual exclusion.

.2

נסמן את argv[1] בתור n, ואת argv[2] בתור m:

(הערה: לאורך השאלה נניח כי n אי שלילי)

עבור n = 0: מתקיים כי לא ניכנס לwhile בmain: ולכן לא נייצר חוטים נוספים חוץ מmain: ולכן לא יעודכן הערך של last או sum: והם ישארו ladt = -1 וsum = 0. ההדפסה עבורם תהיה "x y" (ואחריה ירידת שורה), כך שx היינו 0, וy היינו -1.

עבור $m \neq 0$:

הפלטים: הם "x y" (ואז ירידת שורה) , כאשר x הוא המספר $n(n-1)/2$ וy הוא מספר שלם בתחום $[0, n-1]$. (נשים לב כי x הוא בעצם סכום כל המספרים השלמים מ0 ועד n-1).

עבור m = 0: הפלטים האפשריים הם: "x y" (ואז ירידת שורה), כאשר x הוא מספר שלם בתחום $[0, n(n-1)/2]$ וy הוא מספר שלם בתחום [0,n-1].

ולכן ניתן להגיד כי סך הפלטים האפשריים (עבור n אי שלילי) "x y" (ואז ירידת שורה), כאשר x הוא מספר שלם בתחום $[0, n(n-1)/2]$ וy הוא מספר שלם בתחום [0,n-1].

$$stdout = \{ "x\ y" \mid x \in [0, n(n-1)/2], y \in [0, n-1]\} \cup \{"0\ \ -1"\}$$

****הערה: ישנה ירידת שורה אחרי הפלט, שחשוב להזכיר- היא חלק מהקלט. ולכן הצורה האמיתית היינה: *"x y\n"* . אבל מטעמי נוחות סימנו כך.

הוכחה:

עבור *m != 0*:

נשים לב כי במקרה זה הקוד עבור *f* היינו פשוט:

```
void* f(void* p) {
        pthread_mutex_lock(&mutex);
        int v = *(int *) p;
        sum += v;
        last = v;
        pthread_mutex_unlock(&mutex);
}
```

ולכן סה"כ נקבל כי חוט ה*main*, מייצר *n* חוטים נוספים שרצים במקביל (ויותר בדיוק: הם רצים בו זמנית): נסמנם $T_0, \ldots, T_{n-1}$ כך שכל אחד מהם הוא הפונקציה f בצורה הבאה: החוט T_i מריץ את הפונקציה: *f(void * p)* כאשר *p* הוא מצביע ל*int* שבו שמור הערך *i*.

(לולאת ה*while* יוצרת זאת עבור כל הערכים מ*n-1* (שזה *n--*) ועד *0* (שזה *1--*)).

נשים לב כי הערכים המשותפים לכל החוטים $T_0, \ldots, T_{n-1}$: כלומר *sum, last* – הם ה*critical section* של החוטים הללו (כי כל חוט משנה אותם והם רצים בו זמנית) ונשים לב כי בקוד זה כל הפעולות שמבוצעות עליהם בחוטים, מוגנים על ידי *mutex*. ולכן מתקיים כי בכל פעם רק חוט אחד יכול לשנות את *last* ואת *sum*. בנוסף: מתקיים כי סך השינויים שלו מתבצעים פעם אחת בלבד עבור כל מספר *i* בתחום.

ולכן נקבל כי בהכרח כל חוט יקבל בתורו את המנעול, ינעל אותו ויעדכן את *last, sum* בהתאם, ויפתח את המנעול- ואלו סך השינויים שיתבצעו עבר המספר *i*.

מכיוון שהחוטים רצים בו זמנית, אנו לא יודעים מה יהיה סדר קבלת המנעולים ביניהם: והסדר יכול להיות כל קומבינציה שלהם. אבל בסופו של דבר לכל קומבינציה שהיא: מתקיים כי כל אחד מ*n* החוטים הללו עדכן את *last* ואת *sum* – והוא היחיד שעשה זאת בזמן הזה. ולכן מתקיים כי ה*sum* שאותו מעדכן חוט *i*, הוא תוצאת החיבור של כל העדכונים של החוטים שקיבלו את המנעול לפניו. ולכן נקבל כי סה"כ בסוף ריצת החוטים: ה*sum* יהיה תוצאה סיריאלית של כל השינויים שביצעו החוטים: מכיוון שהפעולה על *sum* (חיבור) אינה מושפעת מסדר חיבור האיברים: מתקיים כי נקבל ש*sum* שווה לסכום המספרים: *0+1+..+n-1* שזה *n(n-1)/2*. ובנוגע ל*last*: כפי שאמרנו, כל קומבינציה של נתינת המנעולים בין החוטים, כך שכל חוט מקבל את המנעול פעם אחת בלבד- היא אפשרות תקינה של סדר ההתרחשויות של החוטים. נשים לב כי החוט שנועל את המנעול אחרון: הוא החוט שגם מעדכן את *last* אחרון (כל השינויים על *last* חייבים להתבצע עם מנעול נעול של החוט המצבע). ישנם *n* אפשרויות לחוטים שיהיו המנעול האחרון: ואלו הם החוטים $T_0, \ldots, T_{n-1}$. כשחוט *i* מקבל את המפתח, הוא מעדכן את *last* להיות *i*).

ולכן מתקיים כי ישנן *n* אפשרויות לתוצאת *last* בסוף הריצה: *0,...,n-1*.

כעת נבדוק עבור *m = 0*:

נשים לב כי עבור מצב כי הפונקציה *f* היינה בפועל:

```
void* f(void* p) {

        int v = *(int *) p;
        sum += v;
        last = v;

}
```

נבהיר כי הפעולה *sum+= v* מכילה מספר פקודות מכונה- על ידי: (זוהי דוגמא בלבד, ויכול להיות שמבוצעת קצת שונה בכל מכונה*

Reg1 = sum

Reg2 = v

Reg3 = reg1+ reg2

Sum = reg3

נראה כי עבור מצב זה לכל מספר x בתחום $[0, n(n-1)/2]$ ולכל מספר y בתחום *[0,n-1]* מתקיים כי ישנה הרצה של *main* ככה ש *sum = x, last = y*.

תחילה נראה כי לכל מספר שלם x בתחום $[0, n(n-1)/2]$, מתאימה תת קבוצה של המספרים: *{0,...,n-1}* כך שסכום האיברים בתת הקבוצה שווה ל*x*.

עבור *x = 0*: נבחר לסכום רק את האיבר 0 לסכום: ונקבל את 0 כנדרש.

נניח נכונות התטענה עבור *n* ונוכיח עבור *n+1* כול עוד הן נמצאות בטווח:

קיים מספר שלם *x* באורך *k+1*, מתקיים כי עבור המספר *n* יש ייצג. אם 1 לא נמצא בייצוג של *n* : אז נייצג את *k+1* על ידי הייצוג של *n* ועוד 1.

אם 1 אכן נמצא בייצוג של *k*: נשים לב כי *k+1* חלק מהטווח, ולכן מתקיים כי גם kשהוא חלק מהטווח לא יכול להיות שהוא הסכום המקסימלי: ולכן חייב להיות מספר כלשהו בין *0* ל*n-1* שלא נמצא בייצוג שלו: ניקח את המספר הראשון שמקיים שהוא נמצא בייצוג של *n*, אבל העוקב שלו (שאחריו) אינו נמצא בייצוג של *n* (בהכרח יש כזה כפי שאמרנו), ונחליף בין שני המספרים: בכך נקבל סכום מספרים שהם בין *0* ל*n-1*, כך שסכומם שווה ל*k+1* כנדרש.

נתחיל בהוכחה:

יהי *x* מספר בטווח: $[0, n(n-1)/2]$, אז לפי טענת העזר ישנם תת קבוצה של מספרים שנסמנה Q כך ש $Q \in \{0,..,n-1\}$ כך שסכום איבר Q היינו $x = q_1 + \ldots + q_k$.

יהי *y* מספר בטווח *[0,n-1]* : אזי או ש$y \in Q$ או שלא.

מקרה 1: $y \in Q$. אזי נסמן את $q_k = y$ (חיבור אינו מושפע מסדר ולכן נשנה את סדר המחוברים כך שהוא יהיה האחרון מבין מרכיבי הייצוג של *x*). מתקיים כי *main* מייצר *n* חוטים שרצים במקביל: $T_0, \ldots, T_{n-1}$. כך שהחוט ה*i* מריץ את הפונקציה *f(void* p)* כך ש*p* מצביע ל*int* שהוא *i*. אזי נראה חישוב כך ש*sum = x* ו*last = y* בסוף ריצת כל החוטים בתכנית:

נתחיל מלהריץ את החוטים הבאים: לפי הסדר וללא החלפות הקשר בכלל: (לפי הסדר משמאל לימין: $T_{q1}, \ldots, T_{q_{k-1}}, T_{q_k}$. עד להרצה של T_{q_k}, מתקיים כי ההרצות התנהלו באופן סיראלי ומבוקר: חוט החל לעדכן את ערכים של *last* ו*sum* רק לאחר שהקודם לו סיים. ולכן בסוף חלק זה $sum = q1 + \ldots + q_{k-1}$ ו $last = q_{k-1}$. כעת נריץ גם את האחרון מבניהם: את T_{q_k} (האחרון מבניהם) אבל נבצע בו החלפת הקשר באופן הבא:

- שומר ב*v* את הערך שלו: q_k
- *Reg1 = sum* (כלומר ישמר ב*reg1* הערך $q1 + \ldots + q_{k-1}$)
- *Reg2 = v* (כלומר ישמר ב*reg2* הערך q_k)
- *Reg3 = reg1 + reg2* (ישמר ב*reg3* הערך $x = q1 + \ldots + q_k$)

בשלב זה נבצע החלפת הקשר: נריץ כעת בסדר סיריאלי ללא החלפות הקשר (בסדר כלשהו) את כל החוטים שהערך שלהם אינו בייצוג של *x*: באופן הבא: עבור $T_{q'1},\dots,T_{q'_{r-1}},T_{q'r}$ כך שלכל אחד מהחוטים הללו הערך שלהם לא נמצא בייצוג של *x*: נריץ אותם משמאל לימין בסדר סיראלי ללא החלפות הקשר. כעת בסיום הרצתן, כל שנותר עוד לעשות כדי לסיים את הריצה הוא לעשות החלפת הקשר בחזרה אל T_{q_k} ולהמשיך מאיפה שהפסקנו. נעדכן

- *Sum = reg3* (*reg3* שומר את הערך *x* ולכן *sum* שווה כעת ל*x*)
- *Last = v* (*v* היינו q_k ולכן *last* = q_k).

סיימנו את הריצה עם הערכים המבוקשים: *last = y* ו*sum = x*.

מקרה 2: $y \notin Q$. אזי מבין החוטים שערכיהם אינם מייצגים את *x*: והם $T_{q'1},\dots,T_{q'_{r-1}},T_{q'r}$, נסמן את *y* בתור ה $q_{'r}$. מתקיים כי *main* מייצר *n* חוטים שרצים במקביל: $T_0,\dots,T_{n-1}$. כך שהחוט ה*i* מריץ את הפונקציה *f(void* p)* כך שp מצביע ל*int* שהוא *i*. אזי נראה חישוב כך ש*sum = x* ו*last = y* בסוף ריצת כל החוטים בתכנית:

נתחיל מלהריץ את החוטים הבאים: לפי הסדר וללא החלפות הקשר בכלל: (לפי הסדר משמאל לימין: $T_{q1},\dots,T_{q_{k-1}},T_{q_k}$. עד להרצה של T_{q_k}, מתקיים כי ההרצות התנהלו באופן סיריאלי ומבוקר: חוט החל לעדכן את ערכים של *last* ו*sum* רק לאחר שהקודם לו סיים. ולכן בסוף חלק זה $sum = q1 + \dots + q_{k-1}$ ו $last = q_{k-1}$. כעת נריץ גם את האחרון מבניהם: את T_{q_k} (האחרון מביניהם) אבל נבצע בו החלפת הקשר באופן הבא:

- שומר ב*v* את הערך שלו: q_k
- *Reg1 = sum* (כלומר ישמר ב*reg1* הערך $q1 + \dots + q_{k-1}$)
- *Reg2 = v* (כלומר ישמר ב*reg2* הערך q_k)
- *Reg3 = reg1 + reg2* (ישמר ב*reg3* הערך $q1 + \dots + q_k = x$)

בשלב זה נבצע החלפת הקשר: נריץ כעת בסדר סיריאלי ללא החלפות הקשר (בסדר כלשהו) את כל החוטים שהערך שלהם אינו בייצוג של *x*: באופן הבא: עבור $T_{q'1},\dots,T_{q'_{r-1}},T_{q'r}$ כך שלכל אחד מהחוטים הללו הערך שלהם לא נמצא בייצוג של *x*: נריץ אותם משמאל לימין בסדר סיריאלי ללא החלפות הקשר. כאשר נגיע להרצה של האחרון מבניהם: $T_{q'r}$ נבצע החלפת הקשר באופן הבא: נספיק לעשות לפני החלפת ההקשר:

- שומר ב*v* את הערך $q_{'r}$
- *Reg1 = sum'* (כאשר *sum'* הוא הערך שהתקבל בינתיים ב*sum* עד שלב כה בריצה)
- *Reg2 = v* (כלומר ישמר ב*reg2* הערך $q_{'r}$)
- *Reg3 = reg1 + reg2*
- *Sum = reg3*

נבצע החלפת הקשר: כל שנותר לנו להריץ הוא את q_k: ולכן נמשיך מאיפה שהפסקנו:

- *Sum = reg3* (כאשר בחוט זה *reg3* שווה ל*x*)

- $Last = v$

ריצתו הסתיימה, אבל עוד נותר לנו לסיים את $q_{\prime r}$, ולכן נבצע החלפת הקשר נוספת ונמשיך מאיפה שהפסקנו:

- $Last = v$ (עבור חוט זה v הוא $q_{\prime r}$ ולכן $last = q_{\prime r}$)

סיימנו את ריצות כל החוטים וקיבלנו $sum = x$, $y = last$ כנדרש.

ובכך הוכחנו כי לכל x וy בתחומים הנדרשים- יש פלט שמתאר אותם.

בנוסף: נשים לב כי ערכו המינימלי של sum היינו 0: כל חוט יכול רק להוסיף ולהגדיל את הסכום.

מצד שני הערך המקסימלי של sum היינו $n(n-1)/2$: וזאת מכיוון שנוכל לראות כי כל חוט של f: מעדכן את sum בערכו – ערך זה הוא חיובי ממש ולכן הוא מגדיל את sum: ולכן עבור מצב בו כל החוטים מגדילים במלואם את sum: זה המקסימום: שכן אם כל אחד מn החוטים מתווסף לsum: הסכום המקסימלי יהיה $0+1+...+n-1$: = $n(n-1)/2$ כנדרש.

ולכן אלו הם הערכים, וכל הערכים שהם הפלטים האפשריים של הפונקציה, כנדרש.