

תרגיל בית 1

מגישים: אלעד משעלי, 3261439997, ענבל סהר 325826303

שאלה 1: אין mutual exclusion.

נראה דוגמה שבה התכונות מתבצעות בצורה לא תקינה ושתי התהליכים נכנסים במקביל ל-critical section:

x = 0, y = 0	int pid = getpid () ;
x = 0, y = 0	while (1) {
x = 0, y = 0	int pid = getpid () ;
x = 0, y = 0	while (1) {
x = O, y = 0	x = pid ;
x = O, y = 0	if (y && y != pid) continue ;
x = B, y = 0	x = pid ;
x = B, y = 0	if (y && y != pid) continue ;
x = B, y = B	y = pid ;
x = B, y = O	y = pid ;
x = B, y = O	if (x != pid) continue ;
x = B, y = O	/*starts critical section */
x = B, y = O	if (x != pid) continue ;
x = O, y = O	x = pid ;
x = O, y = O	if (y && y != pid) continue ;
x = O, y = O	y = pid ;
x = O, y = O	if (x != pid) continue ;
x = O, y = O	/*starts critical section */
x = O, y = O	/*finishes critical section */
x = O, y = O	/*finishes critical section */
x = O, y = O }	
x = O, y = O }	

נסביר מעט את האיור, קוד שכתוב בכחול משמעו שהתהליך הראשון מבצע אותו וקוד שכתוב בכתום משמעו שהתהליך השני מבצע אותו, כל מעבר בניהם הוא context switch. סימנו את ה-*pid* של התהליך הראשון כ-*B* ואת ה-*pid* של התהליך השני כ-*O*.

בטבלה ליד מוזנים ערכי x, y במשך כל הריצה. ניתן לראות שכל ה-*if*-ים עובדים לפי ערכי ה- x, y ובעצם יכול להיווצר מצב שבו שני התהליכים נמצאים ב-critical section במקביל.

שאלה 2:

נפריד למקרים:

1. $mode \neq 0$ – במקרה זה תבצע נעילה לפני כל גישה ל- $sum, last$ (למעט בהדפסה) ולכן הערכים יתעדכנו בצורה חוקית. נשים לב שיתכן שהערכים יתעדכנו לא על פי הסדר, לדוגמא החוט הראשי יכול ליצור את כל החוטים ורק אז המעבד יתחיל לתת לחוטים לרוץ, או אפילו שההדפסה תקרה ורק אחריה החוטים יתחילו לעדכן את $sum, last$. לכן בסה"כ sum יהיה סכום איברי הקבוצה של תת קבוצה כלשהי של $\{0, 2, \dots, n-1\}$. $last$ יכול להיות כל ערך בין -1 (אם הוא עוד לא עודכן) לבין $n-1$.
הערה: נשים לב שההדפסה לא מוגנת בנעילה ולכן התוצאות יכולות להיות לא לגמרי נכונות ביחס למצב האמיתי של התוכנית אך מבחינת קלטים אפשריים זה לא פותח שום אפשרויות חדשות כי בסה"כ עלולים לקרוא ערך לא מעודכן.

2. $mode = 0$ – במקרה זה לא תהיה נעילה לפני גישה לערכים, דבר העלול לגרום תוצאות לא צפויות וערכי זבל.

הערה: סביר להניח שעם מימושים סטנדרטיים בשפת מכונה של הפעולות $=, +, =$ (עבור הפעולה $=$ לטעון את הערך מהזיכרון לרגיסטר ואז לטעון את הערך מהרגיסטר לזיכרון ועבור הפעולה $+=$, לטעון מהזיכרון לרגיסטר לטעון מהזיכרון לרגיסטר אחר כמה צריך להוסיף, להוסיף לרגיסטר המתאים ולשמור בזיכרון) עדיין הערכים שיצאו יהיו כמו במקרה הקודם (בתוצאה הסופית של הפלט) אך בדרך יתכנו כל מיני שגיאות ופספוסים.

בסה"כ המשתנה $last \in \{-1, 0, \dots, n-1\}$ והמשתנה sum סכום אברי הקבוצה של תת קבוצה כלשהי של הקבוצה $\{0, 1, \dots, n-1\}$, ויתכן גם שערכי זבל/תוצאות לא צפויות עלולות להתרחש.