

מערכות הפעלה ת"ב 3

איתמר הפלר 214894263

1. אין *mutual exclusion*. נדגים באמצעות הרצה של התהליכים במקביל שמראה כי הם יכולים להיות בקטע הקריטי בו זמנית. שניהם מגיעים אל תוך ה- *while* ו- $x = pid$ (בה"כ נניח תהליך 2 הגיע אחרון ולכן $x = pid_2$). כעת אחד התהליכים מקבל בעבור ה- $if(y \&\& y \neq pid)$ את התוצאה *false* (משום ש- y אינו קיבל ערך עדיין), ולכן ממשיך אל השורה הבאה. כעת תהליך 1 יגיע שני אל $y = pid$ ולכן $y = pid_1$. כעת רק תהליך שתיים יעבור אל הקטע הקריטי, ותהליך אחד יחזור לתחילת ה- *while* (משום ש- $x = pid_2$). כאן יתרחש ה- *switch - context* אל תהליך 1, ולכן כעת $x = pid_1$. משום ש- $y = pid_1$ אז תהליך 1 יעבור את ה- *if*, יקבע את y מחדש (לא רלוונטי), ויגיע ל- *if* השני אותו הוא גם יעבור משום שכעת $x = pid_1$. כעת נמצאים שני התהליכים בקטע הקריטי באותו הזמן, כלומר אין *mutual exclusion*.
2. הפלטים האפשריים לתוכנית הם:

$sum: \sum_{i=1}^n i, last: 1$ זאת כאשר $mode \neq 0$ ולכן יש נעילה על החלק הקריטי של הקוד שמשנה את *sum* ו- *last*, על כן יתעדכנו לפי סדר יצירת ה- *threads*.

$sum: \sum_{i=1}^n i, last: \{i, 1 \leq i \leq n\}$ זאת כאשר $mode = 0$ ועל כן אין נעילה על החלק הקריטי של הקוד, אך בהנחה ש- $sum += v$ הינה פעולה אטומית, מכיוון שחיבור הוא קומוטטיבי אז נקבל בסוף אותו סכום לא משנה איזה *thread* פעל קודם, אך לכן ערך *last* יהיה ה- *thread* שסיים אחרון, ויכול להיות כל אחד מה- *threads*.

$sum: some m \in N, last: \{i, 1 \leq i \leq n\}$ זאת כאשר $mode = 0$ ועל כן אין נעילה על החלק הקריטי של הקוד, לכן הפלט זהה למקודם מלבד שכעת ללא ההנחה ש- $sum += v$ הינה פעולה אטומית, ייתכן שבמהלך ביצועה יש *switch - context* ולכן נאבד חלק מההוספה אל תוך המשתנה *sum* ולכן נקבל תוצאה לא צפויה.