

תכנות מרובה ליבות - תרגול #4

: Shared memory and synchronization

: סוגי רגיסטרים

Safe registers: כאשר מבצעים write, אם משהו עושה read במקביל הוא רואה משהו בטווח הרגיסטר. בנקודה מסויימת בכתבה הערך מתייצב ושם ב-read הערך נראה יציב.

Regular register, flicker: מבצע flicking בין הערכים החוקיים.

Atomic register: אטומי, עומד בלינארזיביליות, כלומר בנקודה כלשהי המוגדרת כנקודה בה קורה אירוע הכתיבה, ומנקודה זו כל קריאה תראה את הערך החדש. זו ההתנהגות שהיינו מצפים שנראה.

: יצירת MRSW מ-MRSW

בהנחה שיש לנו SRSWSafeBoolean: בכתבה נכתוב לכל ה-SRSW במערך הרגיסטרים שיש לנו. כשמישהו קורא, הוא קורא מהרגיסטר הספציפי שמוקצה לו. ראינו את זה בהרצאה. ע"י מערך רגיסטרים של SRSW יצרנו רגיסטר MRSW.

שאלה: אם נשתמש ב-SRSW regular, האם האלגוריתם הזה ישמש ל-MRSW regular?

תשובה: כן, הוכחה: נוכיח שאם מישהו קורא במקביל שמישהו כותב, הוא יראה או את הערך הישן או החדש, ולא שום דבר אחר. למשל, אם מישהו רואה 0, ומישהו בא וכותב 0, מישהו שיקרא לא יראה 1. מהדוגמא בשקופית מס' 88 ניתן לראות זאת. למה? כי אם מישהו כותב במקביל למישהו שקורא, הקורא בכל מקרה יראה או את הערך הישן או את הערך החדש.

: AtomicMRSW המשתמש ב-stampValue

stampValue הוא מעיין רגיסטר שיש לו כמו version-stamp. בהרצאה ראינו איך באמצעותו בונים MRSWAtomic: בונים מערך דו מימדי שלכל חוט יש שורה ועמודה – שם קורא ערכים ומפרסם ערכים. כאשר קורא שורה – קורא את כל הערכים שכולם פרסמו.

קריאה – הקורא בודק את ID שלו, מה הערך שפרסם, ואז עובר ובודק את המקסימום בין כל ה-stamps בשורה שלו. רוצים שכל אירוע יקרה כאילו בנקודה אטומית, שכולם יראו זאת, ולכן כעת הוא מפרסם את ה-stamp החדש.

כתיבה – הכותב מפרסם את הערך שלו במערך – לוקח את ה-stamp שמכיר, מעלה אותו ומפרסם.

שאלה: אם מחליפים את ה-atomic ב-regular, האם זה ישאר אטומי?

תשובה: כן, אם משתמשים ב-stamp רגולרי ולא אטומי כיוון שהלינארזיביליות נשמרת.

שאלה: רוצים לממש רגיסטר SRSW atomic register, write-once, מרובה ביטים. מוצע אלגוריתם: לוקחים את הערך, מפרקים אותו למערך ביטים (בוליאנים). האלג': מפרסמים במערך ראשון את כל הביטים, אח"כ במערך השני ואח"כ במערך השלישי. כיצד לבצע את ה-read?

תשובה: קוראים את הביטים מהסוף להתחלה ומהמערך השלישי לראשון, ונקבל שלושה מספרים (שלושה מערכים). אסור שיתקיים ששני חוטים קוראים במקביל ומחליטים החלטות הפוכות, רוצים לינארזיביליות. יתכן שמישהו כותב במקביל לקריאה. יש נקודה בה חולפים על פני כותב שכותב במקביל אלינו. לכן הפרוטוקול:

- אם הפגישה עם הכותב היא במערך השני, נחזיר את הערך החדש.
- אם הפגישה היתה במערך האחרון (זה שאני קורא ראשון) אז כבר נחזיר את הערך בשני המערכים הראשונים (האחרונים שאני קורא) – הערך החדש.
- אם שני האחרונים שווים – חלפו על פני הכותב במערך הראשון, ולכן יוחזר הערך מהשניים האחרונים. מוחזר הערך הישן – מובטח שכל מי שרץ לפני החזיר גם את הישן.