

Theory Part

1. הגדרת קטע Doorway היא גישה טובה יותר במימוש אלגוריתמי מניעה הדדית מאשר שימוש ב'FCFS' בהתבסס על הסדר שבו בוצעה ההוראה הראשונה בקטע ה-lock, מאחר והסיבה לכך היא שההתנהגות בקטע ה-lock יכולה להיות תלויה באופי ההוראה הראשונה שבוצעה, במיוחד אם היא קריאה או כתיבה, ואם היא ניגשת לאותו מיקום או מיקומים שונים.
במקרה של הוראת קריאה שבוצעה תחילה, ייתכן שלא יהיה צורך להיכנס לקטע הקריטי כלל (כלומר ללא שינוי המשפיע על הזיכרון), שכן תהליכים מרובים יכולים לקרוא בבטחה את אותם נתונים בו זמנית מבלי לגרום להתנגשויות. לכן, שימוש ב'FCFS' בהתבסס על הסדר עלול לחסום ללא צורך תהליכים מגישה לנתונים, גם אם אין התנגשות ממשיה.
מצד שני, אם ההוראה הראשונה שבוצעה היא הוראת כתיבה, חשוב לוודא שרק תהליך אחד יכול לשנות את הנתונים בכל פעם, כדי למנוע עדכונים סותרים. הגדרת קטע Doorway המשתמש במנעול או Token מבטיחה שרק תהליך אחד יכול לשנות את הנתונים בכל פעם, ולמנוע התנגשויות.
אם הקריאות ניגשות למיקומים שונים, ייתכן שלא יהיה צורך להשתמש במניעה הדדית כלל, מכיוון שלא יהיה סיכון להתנגשויות. במקרה זה, שימוש ב'FCFS' עלול לעכב שלא לצורך תהליכי גישה לנתונים.
לסיכום, הגדרת קטע Doorway היא גישה טובה יותר לאלגוריתם של מניעה הדדית, שכן היא מספקת דרך גמישה יותר להבטיח מניעה הדדית, בהתאם לאופי הנתונים וההוראות המבוצעות. שימוש ב'FCFS' בהתבסס על הסדר עלול להוביל לעיכובים או לחסימות מיותרות, וייתכן שלא יהיה מתאים בכל המקרים.

2. מניעה הדדית:

על מנת לרכוש את המנעול, חוט חייב לרכוש בהצלחה כל מנעול פיטרסון בעל שני חוטים מהעלה שלו ועד השורש. המשמעות היא שהשורש חייב לזכות בכל אחת מהתחרויות הבינאריות בינו לבין ה"שותף" שלו עבור כל אחד מהנעילות. כיוון שמנעול פיטרסון בעל שני חוטים מבטיח מניעה הדדית בין שני החוטים החולקים את המנעול, נובע מכך שלכל היותר חוט אחד יכול להחזיק את המנעול בכל עת ובפרט השורש, לכן מתקיימת מניעה הדדית.

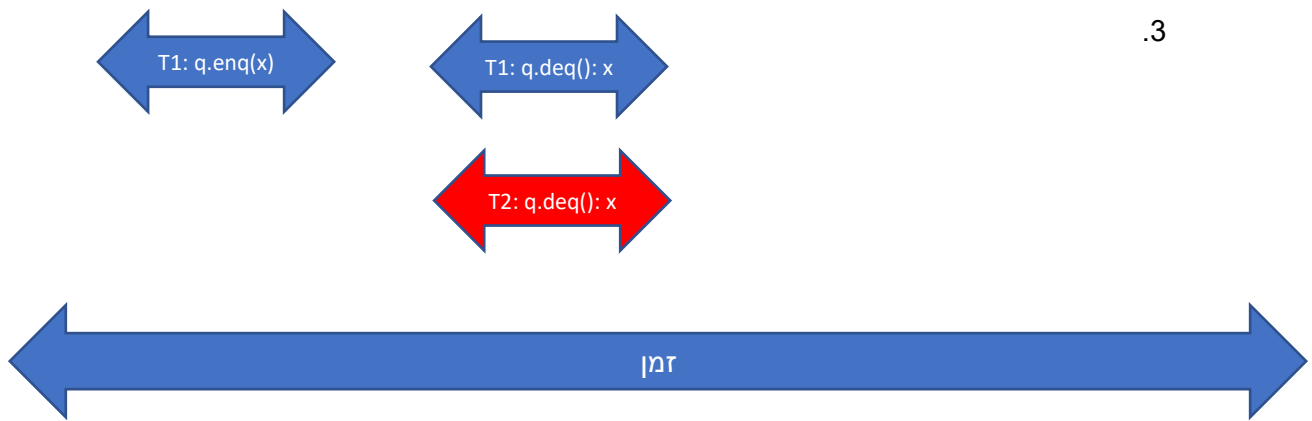
:Deadlock freedom

אם חוט מנסה לרכוש את המנעול, הוא ינסה לרכוש כל מנעול פיטרסון בעל שני חוטים מהעלה שלו לשורש. אם חוט מתעכב למשך זמן סופי כלשהו, הדבר עלול לגרום לחוטים אחרים לרכוש את הנעילה תחילה, אך בסופו של דבר, החוט המושהה יוכל להמשיך ולנסות לרכוש כל נעילה. מכיוון שיש מספר סופי של מנעולים, יוצא שבסופו של דבר חוט כלשהו ירכוש את כולם ולכן ירכוש את המנעול.

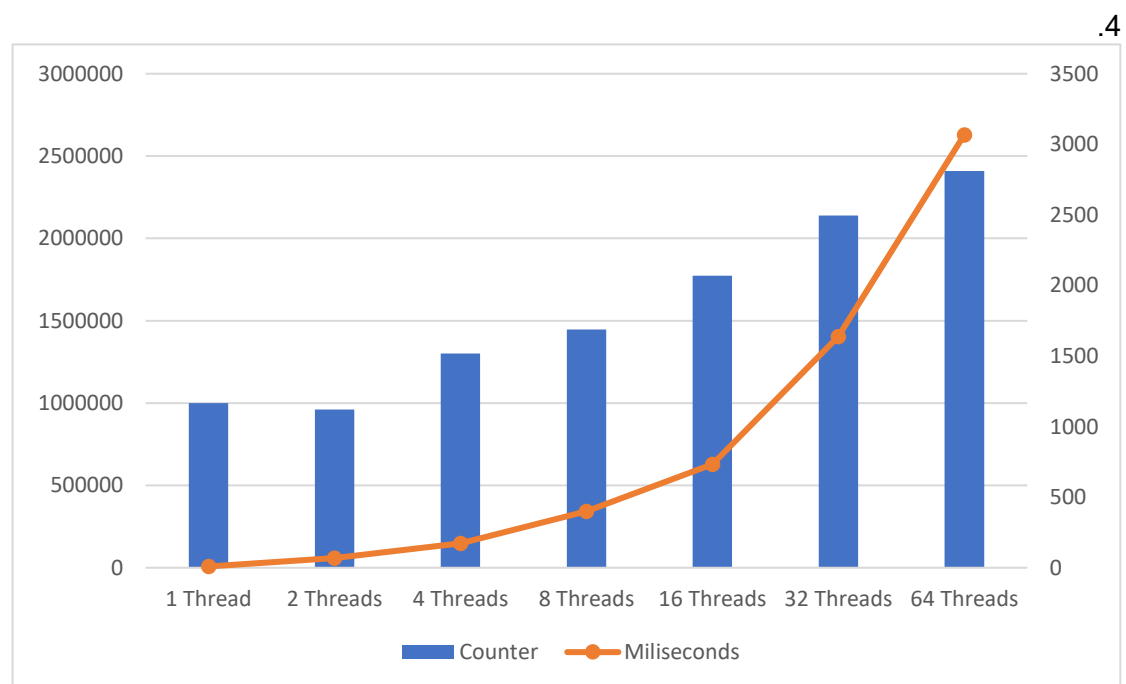
הוגנות:

לא מתקיים.

נתאר תרחיש שבו שני חוטים, A ו-B, חולקים מנעול עלה. חוט A מנסה שוב ושוב להשיג את המנעול, אך תמיד מתעכב לפני שהוא יכול לרכוש את המנעול. בינתיים, חוט B מסוגל לתפוס את המנעול ולשחרר אותו ללא כל דיחוי. מכיוון שיש רק שני חוטים שמתחרים על המנעול, זה אפשרי שחוט אחד תמיד ינצח בתחרויות הבינאריות וימנע מהחוט השני אי פעם לתפוס את המנעול, גם אם החוט המושהה ממשיך לנסות לתפוס את המנעול בתדירות אינסופית.



Programming Part



Finished after 9 milliseconds. The counter is: 1000000, Number of threads: 1

Finished after 68 milliseconds. The counter is: 961375, Number of threads: 2

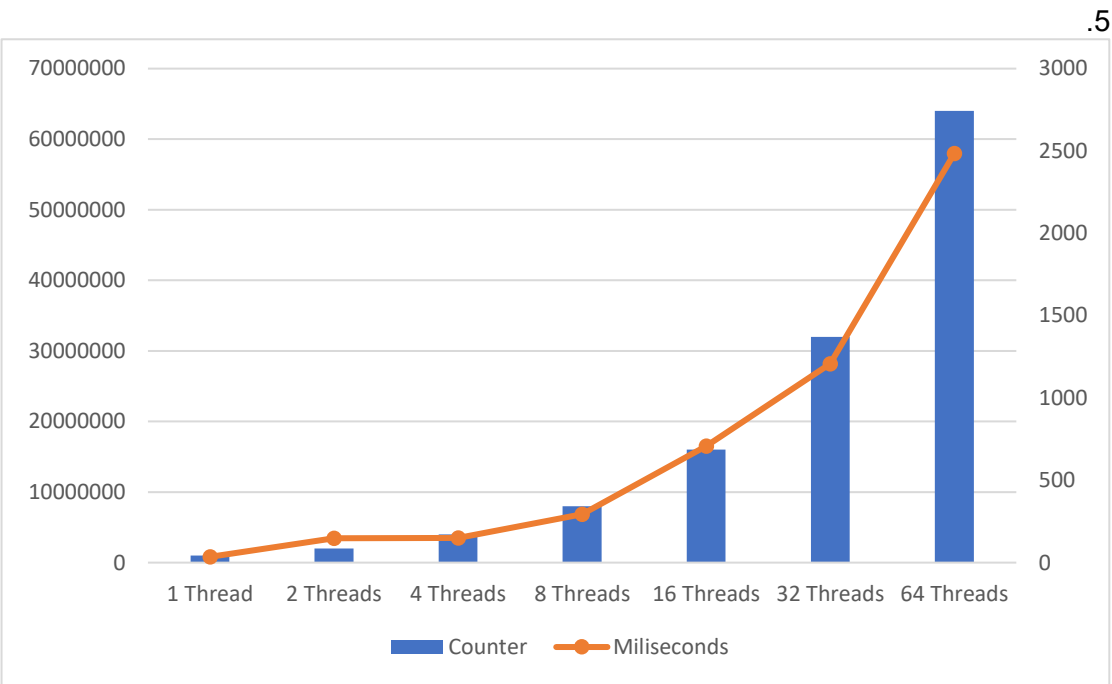
Finished after 173 milliseconds. The counter is: 1301718, Number of threads: 4

Finished after 400 milliseconds. The counter is: 1446339, Number of threads: 8

Finished after 734 milliseconds. The counter is: 1773474, Number of threads: 16

Finished after 1638 milliseconds. The counter is: 2138860, Number of threads: 32

Finished after 3068 milliseconds. The counter is: 2408637, Number of threads: 64



Finished after 35 milliseconds. The counter is: 1000000, Number of threads: 1

Finished after 147 milliseconds. The counter is: 2000000, Number of threads: 2

Finished after 149 milliseconds. The counter is: 4000000, Number of threads: 4

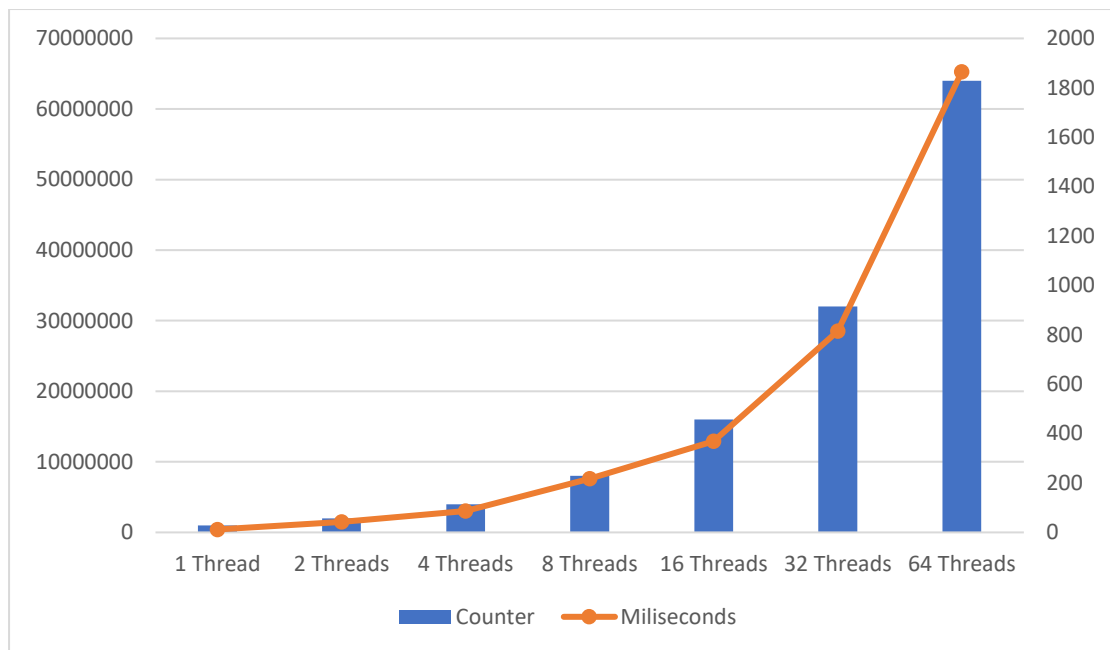
Finished after 294 milliseconds. The counter is: 8000000, Number of threads: 8

Finished after 707 milliseconds. The counter is: 16000000, Number of threads: 16

Finished after 1208 milliseconds. The counter is: 32000000, Number of threads: 32

Finished after 2484 milliseconds. The counter is: 64000000, Number of threads: 64

6.



Finished after 12 milliseconds. The counter is: 1000000, Number of threads: 1

Finished after 43 milliseconds. The counter is: 2000000, Number of threads: 2

Finished after 86 milliseconds. The counter is: 4000000, Number of threads: 4

Finished after 217 milliseconds. The counter is: 8000000, Number of threads: 8

Finished after 369 milliseconds. The counter is: 16000000, Number of threads: 16

Finished after 815 milliseconds. The counter is: 32000000, Number of threads: 32

Finished after 1865 milliseconds. The counter is: 64000000, Number of threads: 64