קורס: מערכות הפעלה

פרויקט מסכם

תאריך הגשה בפועל: 1.02.2023

שם הסטודנט, תעודת זהות, מייל: טולדנו, ליאב, 318239399, liavtoledano@gmail.com

**הנחות והערות:**

1. השתמשתי בThreads ולא Processes.
2. התעלמתי מנושא הFPS מאחר ונאמר בהרצאה שאין צורך להתייחס אליו.
3. הנחתי כי המספר שאותו אנו מבקשים מהמשתמש הוא מספר החוטים שמנתחים את התמונה וישנו רק חוט אחד שקורא את הוידאו.

**אופן הפעולה**

1. התכנית מייצרת משתנה גלובלי בשם results אשר מטרתו לשמור על סדר הדפסה נכון לקובץ הoutput לפי מספר הפריים בסרטון.   
   התכנית עושה זאת בעזרת כתיבת התוצאות של כל פריים (x,y) לאינדקס של מספר הפריים. בהמשך יהיה הסבר מפורט לגבי אופן הפעולה של הכתיבה לקובץ התוצאות.
2. התוכנית מבקשת מהמשתמש את מספר החוטים שאיתם תנתח את התמונות במשתנה num\_of\_threads.
3. התוכנית מייצרת לולאת for על מנת לעבור על כל קבצי הdata וכתיבת המידע לקובץ output בשם המתאים.
4. התכנית מייצרת אובייקט מסוג Lock בשם lock על מנת למנוע מספר חוטים שייגשו למשתנה גלובלי זה.
5. התכנית מייצרת JoinableQueue בשם queue שמטרתו לשמור את ערכי התמונה ואת המספר שלה בתוך הסרטון (תמונה 0, תמונה 1 וכו').
6. התוכנית מייצרת Thread שקורא את המידע שנמצא בקובץ אותו נרצה לנתח – שם הפונקציה שאותה החוט מריץ היא read\_frames אשר מקבלת את הנתיב הרלטיבי לסרטון, את התור, מספר החוטים וlock עבור המשתנה הגלובלי results.
7. התכנית מייצרת Thread שיכתוב את התוצאות לקובץ הoutput המתאים – שם הפונקציה אותה הThread מריץ היא write\_to\_output והיא מקבלת את שם הקובץ אליו אנו נכתוב את התוצאות של הסרטון הנוכחי.
8. התכנית מייצרת num\_of\_threads חוטים שינתחו כל תמונה. שם הפונקציה אותה הם יריצו היא find\_matrix אשר מקבלת את queue ואת lock.
9. התכנית תחכה שהחוט שקורא את כל התמונות יסיים את עבודתו, לאחר מכן יחכה שהחוטים המנתחים והתור יסיימו את עבודתם ולאחר מכן גם החוט הכותב.
10. התכנית תדפיס את זמן ההרצה שלקח לעבד כל סרטון ותאפס את results בשביל לנתח את הסרטון הבא.

**אופן פעולת הפונקציות**

1. **read\_frames(path\_to\_file, queue, num\_of\_threads, lock)**

* פותחת את הקובץ על ידי שימוש ב path\_to\_file
* בתוך לולאת while אינסופית תקרא 600\*800 בתים בקובץ הסרטון, תסדר אותם לפי מימדי התמונה אותם התבקשנו לנתח ותשים את התמונה בתוך queue ביחד עם מספר התמונה בסרטון בtuple : (frame, frame\_num).
* במידה ולא קראנו כלום, הגענו לסוף הסרטון ולכן נשים בתוך התור את הסימן ("###END###", "###END###") כדי לסמן שהגענו לסוף הסרטון.
* לאחר קריאת כל פריים, בעזרת שימוש בlock, נוסיף ל resultsאיבר (0,0), כך שלאחר קריאת כל הפריימים נקבל בתוך results את הערך (0,0) ככמות הפריימים בדיוק.

1. **find\_matrix(queue, lock)**

* בתוך לולאת while אינסופית קוראת בכל פעם איבר מתוך queue.   
  במידה והאיבר הוא ("###END###", "###END###")**,** התכנית תוסיף לresults, בעזרת lock, את הערך "###END###"כדי לסמן שהגענו לסוף הסרטון.
* במידה ולא הגענו לסוף הסרטון, כל איבר בתוך התור נראה כך -   
  (frame, frame\_num), ניקח את הפריים ונחשב את הממוצע שלו ואת ערכיו 5%+ ו 5%- כדי למצוא את גבולות המטריצה אותה אנו מחפשים.
* נגדיר דגל Found שיסמן אם מצאנו מטריצה בפריים זה. נאתחל אותו לFalse.
* נעבור על כל המטריצות בגודל 3\*3 בתמונה ונחפש את המטריצה שעומדת בתנאים. במידה ונמצאה מטריצה שעומדת בתנאים, בעזרת שימוש בlock נשים את האינדקסים המרכזיים של המטריצה בתוך משתנה results באינדקס של frame\_num אותו קיבלנו מהqueue.

1. **write\_to\_output(output\_file)**

* נאתחל משתנה i להיות 0. משתנה זה יצביע על מספר הפריים אותו עכשיו אנחנו צריכים להדפיס לקובץ התוצאות בשם output\_file.
* בתוך לולאת while אינסופית נפתח את output\_file לכתיבה וכל פעם נבדוק האם results[i] עדיין (0,0) כפי שאותחל בהתחלה.
* רק במידה והערך שונה מ(0,0), נדע שפריים במיקום i עבר עיבוד והתוצאה ששמורה בו עכשיו היא תוצאת העיבוד – כלומר האינדקסים האמצעיים של המטריצה הרצויה.
* נכתוב את התוצאות לקובץ כנדרש ונקדם את i לערך הבא.
* במידה ונקרא "###END###", נדע שהגענו לסוף ונצא מלולאת הwhile.
* היתרון בשיטה זו, זה שאנחנו לא צריכים לחכות לכל פריים שיסיים אלא אם פריים 2 סיים לפני פריים 1 אז נחכה ל1 ואז ישר נדפיס את 2 כי הערך באיבר 2 יהיה שונה מ(0,0).