תרגיל בית 3

מועד הגשת התרגיל: עד יום שני 13/12/2021 בשעה 23:55:00.

מטרת התרגיל

- העמקת ההבנה במושגי החוט (Thread) במערכות הפעלה בכלל וב-Windows בפרט.
- עבודה עם מספר חוטים במקביל.
- שימוש ב-Mutex וב-Semaphore לסנכרון גישה למשאבים משותפים בין חוטים.
- הימנעות מ-deadlock-ים.
- צורת ההגשה מפורטת במסמך "הנחיות להגשת תרגילי בית 2021-2022" שבאתר המודל. אנא הקפידו למלא אחר ההוראות.

הגישו פרויקט מלא, כולל קבצי פרויקט (sln, *.vcxproj, *.vcxproj.filters.*) של Visual Studio 2019, באופן שיאפשר לבודק התרגילים לפתוח את הפרויקט על ידי לחיצה כפולה על קובץ ה-solution ולקמפל את הפרויקט ללא התראות או שגיאות.
שם הפרויקט צריך להיות Ex3. שם ה-executable צריך להיות Ex3.exe.

הגישו בנוסף את תיקיית ה-Debug עם קובץ ה-exe.

דגשים

- הקפידו על קוד קריא ומתועד.
- עבדו באיטרציות – בדקו את הקוד שלכם לעתים תכופות בעת הקידוד, ולא לאחר כתיבת התוכנה כולה.
- רשמו לעצמכם את מבנה התוכנה הכללי לפני שאתם מתחילים לקודד.
  - חשבו איזה מודולים ופונקציות אתם צריכים. מתוך הפונקציות, איזה יהיו סטטיות ואיזה פומביות.
  - זכרו כי כל קטע קוד שאתם משתמשים בו יותר מפעם אחת, צריך להיכתב כפונקציה נפרדת. כאשר פונקציה נעשית גדולה ומסובכת, פצלו אותה למספר פונקציות.
  - בפרט, נסו לקבץ במודולים נפרדים פונקציות שקשורות לתהליכים וחוטים (כגון CreateProcessSimple שראינו בתרגול). המודולים האלה ישמשו אתכם גם בתרגילים הבאים.
- זכרו להשתמש בכלי הדיבוג שה-IDE מספק.
- אין דרך אחת נכונה לפתור את התרגיל והתרגיל לא כוון לפתרון ספציפי.
- השתמשו בזיכרון דינמי לאחסן מידע שגודלו אינו ידוע בזמן הקומפילציה. איננכם רשאים להניח חסם עליון שרירותי לגודל המידע. השתמשו בקבועים ושימו לב לשחרור זיכרון דינמי.
- אתחלו את כל הפוינטרים ל-NULL. כל פונקציה שמקבלת מצביע צריכה לבדוק שהוא שונה מ-NULL לפני שהיא עושה dereference (אופרטור *).
- בדקו את ערך החזרה של כל פונקציה, שיכולה להחזיר שגיאה (malloc, WaitForSingleObject וכו'). פעלו בהתאם לערך.
- לפני שאתם משתמשים בפקודת API בפעם הראשונה, רצוי לקרוא את התיעוד שלה ב-MSDN שלה. באופן כללי, רצוי גם לקרוא את הפונקציות שמופיעות ב-MSDN תחת Related Functions.
- הפורום עומד לשירותכם. אנו מעודדים אתכם לנסות תחילה לחפש תשובות באינטרנט, כאשר מדובר בשאלות תכנות כללית.
- יש לשחרר את כל המשאבים בסיום הריצה (File Handles, Thread Handles, וזיכרון דינאמי).
- ניתן להניח קיימות ותקינות הקלט.
- עבור טיפול בקבצים יש להשתמש בספריית WinApi.

בהצלחה!

סקירה כללית

בתרגיל זה תצטרכו לממש מערכת דפדוף של זיכרון וירטואלי .
גודל הדפים והמסגרות הינו 4KB.
לאובייקט דף יש את הפרמטרים הבאים:

1. מספר מסגרת אליה ממופה הדף
2. ביט Valid. כאשר Valid=1 מספר המסגרת שמצוין ב-1. תקין.
3. זמן סיום השימוש במסגרת, יוסבר בהמשך.

בתרגיל תממשו טבלת דפדוף הממפה דפים למסגרות. האינדקס לתוך טבלת הדפדוף מייצג את מספר הדף. מימוש מבני הנתונים הוא לבחירתכם. השדות הנ"ל חייבים להיות חלק ממבני הנתונים.

קלט

קלט התוכנה הוא כדלהלן (בהפעלה מה-command line):

<path to executable> <number of bits in virtual memory> <number of bits in physical memory> <path to input file>

<path to executable> - נתיב התוכנית.
<number of bits in virtual memory> - מספר הביטים בכתובת הווירטואלית במערכת ההפעלה. שלם גדול שווה ל-12, וקטן מ-32.
<number of bits in physical memory> - מספר הביטים בכתובת הפיזית במערכת ההפעלה. שלם וגדול שווה ל-12, וקטן מ-32.
<path to input file> - נתיב לקובץ הקלט.

מספר הביטים בכתובות הפיזיות והווירטואליות מגדיר את כמות הדפים הווירטואליים ואת כמות המסגרות הפיזיות.

בקובץ הקלט מתואר קריאות זיכרון למערכת ההפעלה. פורמט כל שורה בקובץ הקלט הינו:

<time> <virtual address> <time of use>

<time> - זמן הקריאה לכתובת. מספר שלם אי שלילי.
<virtual address> - כתובת וירטואלית שנקראה על ידי מערכת ההפעלה. מספר שלם אי שלילי בייצוג עשרוני. ניתן להניח שלא תינתן כתובת שלא נמצאת במרחב הכתובות הווירטואליות הנתונה.
<time of use> - זמן השימוש בכתובת. מספר שלם חיובי.

זמני הקריאה הינם מונוטוניים עולים.

*הערה: במערכת דפדוף אמיתית, זמן השימוש בדף אינו ידוע מראש.

המשימה

בהינתן קובץ הקלט התוכנית תפתח ותעדכן טבלת דפדוף שממפה דפים וירטואליים למסגרות פיזיות. עבור כל שורה בקובץ, התוכנית תפתח חוט שיטפל בשורה, יעדכן את טבלת הדפדוף בהתאם לתוכן השורה:

1. במידה והדף הרצוי אינו נמצא במסגרת כלשהי יש לפנות מסגרת על פי המדיניות הבאה:
   החוט יעבור על המסגרות הפיזיות מהאינדקס הקטן ביותר ועד לגדול ביותר כלומר בסדר עולה, והדף הראשון שנמצא במסגרת שהשימוש בה הסתיים יפונה והמסגרת תינתן לדף החדש.
   במידה ואין מסגרת פנויה, החוט ימתין לזמן סיום השימוש הקרוב ביותר של מסגרת כלשהי.
2. במידה והדף הנדרש נמצא כבר במסגרת, החוט יעדכן את זמן סיום השימוש במסגרת במידה וזמן סיום השימוש של הקריאה הנוכחית גדול יותר מזמן סיום השימוש בטבלת הדפדוף והחוט ייסגר.
3. יש למנוע הרעבה של מנגנון העדכון.

בונוס: השתמשו במדיניות החלפה LRU עבור פינוי הדפים מהמסגרות.

פלט

התוכנית תפתח קובץ פלט בתיקיה של קובץ הקלט בשם Output.txt והתוכנית תרשום אליה בכל שורה בפורמט הבא:

<time of placement/eviction in frame> <virtual page number> <physical frame number> <P/E>

<time of placement/eviction in/from frame> - הזמן בו מסגרת נתפסה או פונתה

<virtual page number> - מספר הדף הווירטואלי שתפס מסגרת / פונה ממסגרת.

<physical frame number> - מספר המסגרת שנתפסה/פונתה.

<P/E> - P עבור תפיסת מסגרת, ו-E עבור פינוי של מסגרת.

התוכנית תעדכן את קובץ הפלט בכל תפיסה/פינוי של מסגרת.

הערה: פינוי של מסגרת מתרחש כאשר דף אחר תופס אותה, ולא כאשר עבר זמן סיום השימוש.

בסיום הריצה, כאשר עבר זמן השימוש של כל הדפים יש לפנות את כל המסגרות.

דוגמה

עבור הקלט "./Input.txt Ex3.exe 14 13" קבצי הקלט והפלט המצורפים במודל

בזמן 0, מסגרת 0 תיתפס על ידי דף 0 עד לזמן 2000 כיוון שמסגרת 0 היא הראשונה שפנויה.
בזמן 100, דף 2 יתפוס את מסגרת 1 עד לזמן 1100 כיוון שמסגרת 0 תפוסה ומסגרת 1 פנויה.
בזמן 900, דף 1 מחפש מסגרת פנויה וימתין כיוון שאין מסגרת פנויה.
בזמן 1000, זמן הסיום של מסגרת 1 מתעדכן ל-1200 כיוון שיש קריאה נוספת לדף מספר 2 והוא כבר נמצא במסגרת.
בזמן 1200, דף 1 יתפוס את מסגרת 1 כיוון שהמסגרת התפנתה.
בזמן 1500, זמן הסיום של מסגרת 0 מתעדכן ל-2300 כיוון שיש קריאה נוספת לדף 0 והוא נמצא במסגרת וזמן הסיום של הכתובת החדשה גדול מזמן הסיום הנוכחי של המסגרת.
בזמן 2000, דף 3 יתפוס את מסגרת 1 כיוון שמסגרת 0 תפוסה ומסגרת 1 פנויה.
בזמן 2300 כל הדפים סיימו לעבוד וכל המסגרות יפונו בסדר עולה מהמסגרת בעלת האינדקס הנמוך ביותר למסגרת בעלת האינדקס הגבוה ביותר.

מערכת הדפדוף צריכה להיראות כך בזמנים השונים:

Frame number	valid	End of use
N/A	0	N/A
N/A	0	N/A
N/A	0	N/A
0	1	2000

time = 0

Frame number	valid	End of use
N/A	0	N/A
1	1	1100
N/A	0	N/A
0	1	2000

time = 100

Frame number	valid	End of use
N/A	0	N/A
1	1	1200
N/A	0	N/A
0	1	2000

time = 1000

Frame number	valid	End of use
N/A	0	N/A
N/A	0	N/A
1	1	1400
0	1	2000

time = 1200

Frame number	valid	End of use
N/A	0	N/A
N/A	0	N/A
1	1	1400
0	1	2300

time = 1500

Frame number	valid	End of use
1	1	2100
N/A	0	N/A
N/A	0	N/A
0	1	2300

time = 2000