

תרגיל בית 2

מועד הגשת התרגיל: עד יום שבת 7/12/19 בשעה 23:55. לא תהינה דחיות

בודק אחראי: בועז פיש boazfish@mail.tau.ac.il

מטרת התרגיל

- הבנה ושימוש בתהליכים וחוטמים (processes and threads) ב-Windows.
- עבודה מקבילית עם תהליכים וחוטמים.
- שימוש ב-MDSN Documentation.
- שימוש בפונקציות API של תהליכים וחוטמים.

הגשה

צורת ההגשה מפורטת במסמך "הנחיות להגשת תרגילי בית – תש"ף" שבאתר המודל. אנה הקפידו למלא אחר ההוראות.

הגישו פרויקט מלא, כולל קבצי פרויקט (*.sln, *.vcxproj, *.vcxproj.filters) של Visual Studio 2017, באופן שיאפשר לבודק התרגילים לפתוח את הפרויקט על ידי לחיצה כפולה על קובץ ה-solution ולקמפל את הפרויקט ללא התראות או שגיאות.

הגישו בנוסף את תיקיית ה-Debug עם ה-Exe-ים.

דגשים

הקפידו על קוד קריא ומתועד.

עבדו באיטרציות. בדקו את הקוד שכתבתם לפחות בסוף כל סעיף. בדקו תחילה שהקוד שלכם עובד עבור מקרה בדיקה יחיד לפני הרחבה למספר רב של בדיקות.

זכרו להשתמש בכלי הדיבוג שה-IDE מספק.

הפורום עומד לשירותכם. אנו מעודדים אתכם לנסות תחילה לחפש תשובות באינטרנט, כאשר מדובר בשאלות תכנות כלליות.

בהצלחה!

סקירה כללית

בתרגיל זה נלמד לעבוד עם תהליכים וחוסים. לכן, תכתבו מערכת להרצת מספר בדיקות במקביל עם מספר קבצים. סט הבדיקות ירוצו במקביל על כל קובץ.

בתרגיל זה תחשבו ציון סופי של סטודנט בקורס "חוסים מקבילים למתחילים" בפקולטה להנדסה. בקורס זה יש 10 תרגילי בית (20% מהציון הסופי), בוחן אמצע (20% מהציון הסופי), ומבחן סופי (60% מהציון הסופי). הציון על תרגילי הבית הוא הממוצע של 8 התרגילים הטובים ביותר שהסטודנט הגיש. במידה וסטודנט לא הגיש תרגיל בית, הציון על התרגיל הוא 0.

ציון נכשל (מתחת ל-60) בתרגיל בית, בוחן, או מבחן, ייחשב כ-0 לצורך חישוב ציון סופי.

ישנם 2 מועדים למבחן בקורס, כאשר הציון האחרון הוא הקובע.

ניתן להניח 0 במבחן/תרגיל יינתן רק למי שכלל לא ניגש/הגיש.

מימוש

על התוכנה שעליכם לממש לבדוק את כלל הציונים כמעט בו-זמנית. לשם כך, יש לייצר עבור כל בדיקה **חוט** שיהיה אחראי על הרצתו. בנוסף, על סט הבדיקות הכולל לרוץ **כתהליך** נפרד לכל תלמיד.

התכנית תסיים כשכל התהליכים יסיימו לרוץ והתוכנית תדפיס הודעת סיום.

כלומר, עליכם לכתוב שתי תכניות:

1. תכנית הניהול `TestManager.exe` – שתנהל את כלל הבדיקות (תהליכים), כאשר כל סט בדיקה רץ **כתהליך** נפרד – זו התכנית שאנחנו נריץ ונבדוק לכם. תכנית זו מריצה את:
 2. תכנית הבדיקות `TestGrade.exe` – שתריץ את הבדיקות **כחוסים**.

תכנית הבדיקות `TestGrade.exe`

התכנית תריץ את הבדיקות השונות כחוסים נפרדים ותסתיים רק כאשר כל החוסים סיימו את ריצתם. כל חוט יבצע בדיקה של ציון אחר של התלמיד (ע"י קריאת הקובץ הרלוונטי וחילוץ הציון המספרי).

לכל סטודנט יש תיקייה משלו בשם `grades_123456789` עבור סטודנט עם ת"ז `123456789`. בתוך כל תיקייה יופיעו קבצים עם השמות `ex01.txt`, `ex02.txt` וכו' עד `ex10.txt` לציוני תרגילי הבית, `midterm.txt` לציון הבוחן אמצע, `moedA.txt` ו-`moedB.txt` לציוני המבחנים. בתוך כל קובץ טקסט יופיע מספר בלבד.

על החוט הראשי לשלוח את הבדיקות השונות לריצה, להכניס למצב המתנה לסיום ריצת החוסים. יש להמתין לסיום ריצת החוסים באמצעות פונקציית `WaitForMultipleObjects`.

לבסוף, החוט הראשי יחשב את הציון הסופי בקורס, ויכתוב את הציון לקובץ `final_123456789.txt`, שיהיה בתיקייה הרלוונטית. מתוך קובץ זה יילקח הציון הסופי לקובץ התוצאות המלא בסוף התכנית.

על כל בדיקה להתחיל בפקודת `Sleep(10)`.

תכנית הניהול `TestManager.exe`

תכנית הניהול תקרא `TestManager.exe`. תכנית הניהול תנהל את הריצות השונות של תכנית הבדיקות. כלומר, התכנית תריץ את `TestGrade.exe` עבור כל ת"ז של סטודנט ברשימה כתהליך

נפרד. התכנית תסיים כאשר כל התהליכים יסיימו את ריצתם ויכתבו את הציונים הסופיים לקובץ הציונים הסופיים.

התכנית תסתיים כאשר כל התהליכים יסיימו לרוץ ויודפס מידע סיום התכנית. במידה והתכנית הסתיימה בהצלחה, על המסך יודפס "The grades have arrived, captain". במידה והתכנית נכשלה בבדיקה של אחד הציונים, יש להדפיס על המסך "Captain, we were unable to calculate {}" כאשר במקום הסוגריים המסולסלים יופיע הת"ז שהיה בבדיקה בעת הכישלון.

שורת ההרצה של תכנית הניהול:

התכנית תרוץ ע"י שורת פקודה עם הארגומנטים הבאים:

TestManager.exe <Grades Directory>

כאשר Grades Directory היא התיקייה שבה מופיעים תיקייה נפרדת לכל סטודנט. בתיקייה זו תופיע בסוף הרצת התכנית קובץ final_grades.txt עם הפורמט הבא:

123456789 93

234567890 78

כלומר, הדפסה של מספר הת"ז, רווח, ציון סופי (מעוגל כלפי מעלה), ומעבר לשורה הבאה.

בתיקיית Grade Directory מופיע קובץ studentIds.txt בה נמצאים כל תעודות הזהות של התלמידים בשורות נפרדות. ניתן להניח שלכל ת"ז ברשימה יש תיקייה משלה.

שורת הרצה של תכנית הבדיקות:

התכנית תרוץ ע"י שורת פקודה עם הארגומנטים הבאים:

TestGrade.exe <grade_path>

כאשר grade_path מייצג את ממסלול הקובץ הספציפי עם הציון שהחוט אמור לבדוק.

התמודדות עם שגיאות:

יש לבדוק את הצלחה של כל פונקציה שעשויה להיכשל (הקצאת, זיכרון, פתיחת קבצים, יצירת חוט וכו'). במידה שמתרחשת שגיאה, יש לסיים את התוכנית באופן מסודר:

1. יש לשחרר/לסגור בצורה מסודרת את כל המשאבים שהוקצו: זיכרון, חוטים, קבצים וכו'.

2. יש להדפיס הודעת שגיאה למסך.

3. התמודדות עם שגיאה שמתרחשת בחוט שאיננו החוט הראשי:

3.1 אין צורך לעצור מיד את שאר החוטים הלא-ראשיים וניתן לאפשר להם לרוץ עד סיום.

3.2 יש ליידע את החוט הראשי באמצעות שימוש בערך היציאה (exit code) של החוט שבו התרחשה השגיאה.

הנחות:

- ניתן להניח שכל ציון הוא מספר שלם בין 1-3 ספרות.

- ניתן להניח שכל הקבצים הרלוונטיים קיימים ולא ריקים.
- ניתן להניח שמספר הסטודנטים בקורס לא עולה על 10.
- ניתן להניח שכל ת"ז היא באורך 9 ספרות בדיוק.
- ההנחות וההנחיות מתרגיל בית 1 תקפים גם לתרגיל זה.

בהצלחה!