

מיני פרויקט בארגון וניהול קבצים מגשים: אליהו פולק, משה פוגל ועידן בן שמעון.

רקע כללי:

מעבדה זו עוסקת במימוש מנגנון לארגון וניהול קבצים באמצעות פונקציית HASH (הממיינת כל רשומה ורשומה למקומה בקובץ), ובנוסף משתמש באזור גלישה משורשר בתוך הקובץ, לצורך הרשומות שלא נותר להן מקום בבלוק הרשומות שאליו ייעדה אותן פונקציה זו.

מימוש זה ייעשה בחמישה שלבים:

- i. שלב 0 - בניית התשתית הפיזית של הקובץ.
- ii. שלב 1 - בניית התשתית הלוגית של הקובץ.
- iii. שלב 2 - בניית מנגנון פונקציית ה HASH (הכולל בשלב זה חיפוש, קריאת וכתיבת רשומה).
- iv. שלב 3 - מחיקה ועדכון של רשומה מן הקובץ.
- v. שלב 4 - בניית ממשק גרפי עבור המשתמש.

דו"ח שלב 0.

לצורך מימוש התשתית הפיזית נבנו שלוש מחלקות:

:Physical block

מחלקת בסיס המשמשת כחוצץ לצורך כתיבה/קריאה לקובץ.

מכילה מערך מסוג char בגודל 20 המיועד לשימוש שמור (אזור filler) ומערך נוסף מסוג זה בגודל 1000 המיועד למידע המשתמש (אזור data).

מהפן ההתנהגותי המחלקה מכילה בנאי ברירת מחדל, ופונקציה שתפקידה לאתחל את הבלוק באפסים בינאריים (זו פונקציה שהוספנו מעבר לדרישות השלב).

:Current block

מחלקה המכילה שדה מסוג int המכיל את המספר הסידורי של הבלוק מתחילת הקובץ, ובנוסף מופע של המחלקה Physical block כחוצץ.

מהפן ההתנהגותי המחלקה מכילה רק בנאי ברירת מחדל המאתחל את המספר הסידורי (את החלק השייך למחלקה Physical block הבנאי של אותה מחלקה מאתחל).

הסתפקנו האם לאחד את שתי המחלקות הללו אם לאו, בסופו של דבר הוחלט להשאיר את המחלקות על כנן מפני לשון לא מבוררת כל צורכה, בניסוח השדות שתכיל המחלקה Physical File.

: Physical File

המחלקה העיקרית בשלב זה הממשת בפועל את התשתית הפיזית, מכילה:

שדות:

מיני פרויקט בארגון וניהול קבצים

מגשים: אליהו פולק, משה פוגל ועידן בן שמעון.

- **Opened** - שדה בוליאני המציין אם הקובץ פתוח פיזית (מצב true) או סגור (מצב false).
- **Openmode** - שדה המכיל מידע על אופן פתיחת הקובץ (כתיבה, קריאה או שניהם).
- **Filefl** - שדה שהנו מופע של אובייקט מסוג fstream המשמש כצינור הכתיבה והקריאה בפועל לקובץ.
- **WorkingDir** - שדה המכיל את הנתיב המלא של מקום אחסון הקובץ במחשב (אופציונלי גם במיקום שונה מתיקיית הפרויקט).
- **File name** - שדה המכיל את שם הקובץ.
- **File size** - שדה המכיל מספר המציין את גודל הקובץ בבתים.
- **Curr block** - שדה שהנו מופע של אובייקט מסוג Current block המכיל את הבלוק הנוכחי בקובץ (לתכולתו ראה לעיל על המחלקה Current block).
- **FHBuffer** - שדה שהנו מופע של אובייקט מסוג Current block המכיל את הבלוק הראשון בקובץ המכונה file header (מכיל מידע על הקובץ)

פונקציות:

- שלושה בנאים:
 1. **בנאי ברירת מחדל**. תפקידו לאתחל אובייקט מסוג Physical File בלי קישור לאף קובץ קיים.
 2. **בנאי יוצר**. תפקידו לאתחל אובייקט מסוג Physical File תוך כדי יצירת קובץ פיזי חדש (משתמש בפונקציית pcreate).
 3. **בנאי פותח**. תפקידו לאתחל אובייקט מסוג Physical File תוך כדי קישור קובץ קיים לאובייקט.
- **הורס:** פונקציה זו הורסת אובייקט מסוג Physical File ללא מחיקת הקובץ הקשור אליו פיזית.
- **Pcreate:** פונקציה זו יוצרת פיזית את הקובץ תוך כדי שהיא ממלאה את הבלוקים באפסים בינאריים (הכתיבה באמצעות הפונקציה WriteBlock) (בשדה הנתיב הוכנס הנתיב המלא כולל הסיומת HASH מחמת תקלות שנוצרו ביצירת/מחיקת הקובץ).
- **Pdelete:** פונקציה זו מוחקת את הקובץ פיזית באם קיים (תוך עדכון השדות הנחוצים).
- **Popen:** פונקציה זו פותחת את הקובץ לקריאה או כתיבה או שניהם במידה והקובץ קיים (תוך עדכון השדות הנחוצים) (בסוג הפתיחה אין רגישות לאותיות גדולות וקטנות, כמו"כ בשדה הנתיב הוכנס הנתיב המלא כולל הסיומת HASH מחמת תקלות שנוצרו ביצירת/מחיקת הקובץ).
- **Pclose:** פונקציה זו סוגרת את הקובץ באם קיים (תוך עדכון השדות הנחוצים).
- **SeekToBlock:** פונקציה זו מזיזה את ראשי הקריאה והכתיבה לבלוק המבוקש לקריאה כתיבה או שניהם. בכך בלוק זה נהיה הבלוק הנוכחי באובייקט כך שהיא מעדכנת את השדות הנחוצים לשם כך.
- **writeBlock:** פונקציה זו כותבת לבלוק הנוכחי בקובץ.
- **readBlock:** פונקציה זו קוראת לבלוק הנוכחי בקובץ.
- **writeFH:** פונקציה זו כותבת מהחוצץ של ה FHBuffer אל הקובץ.
- **readFH:** פונקציה זו קוראת מהקובץ אל החוצץ ה FHBuffer.

מיני פרויקט בארגון וניהול קבצים מגשים: אליהו פולק, משה פוגל ועידן בן שמעון.

דו"ח שלב 1.

לצורך מימוש התשתית הלוגית נבנו 3 מחלקות המשמשות מעטפת לשלב 0:
(שתי המחלקות הראשונות מהוות חלוקה לוגית של המחלקה הפיזית Physical block עבור בלוק רגיל
ובלוק של File Header).

:LogicalBlock

שדות:

- BlockNr – שדה מסוג int המכיל מספר סידורי של הבלוק בקובץ.
 - NrOfOverflowedRecs - שדה מסוג int המכיל מספר המציינ כמה רשומות גלשו לאיזור הגלישה.
 - OverflowBlockPtr – שדה מסוג int המכיל מספר של בלוק בשרשרת אזור הגלישה המשמש "מצביע" לאזור הגלישה.
 - NrOfRecsInBlock – שדה מסוג char המכיל מספר המציינ את מספר הרשומות בפועל בבלוק.
 - Filler – מערך בגודל 11 מסוג char השמור למידע מערכת.
 - Data – מערך בגודל 1000 מסוג char המשמש למידע המשתמש.
- גודל כל המחלקה 1024 בתים כמתוכנן.

:File Header

שדות:

- BlockNr – שדה מסוג int המכיל מספר סידורי של הבלוק בקובץ.
- FileName – שדה בגודל 12 מסוג char המכיל את שם הקובץ.
- OwnerName – מערך בגודל 10 מסוג char המכיל את שם בעל הקובץ.
- FileSize - שדה מסוג int המכיל מספר המציינ את כמות הבלוקים שבקובץ.
- CreationDate - מערך בגודל 10 מסוג char המכיל את התאריך.
- RecordSize - שדה מסוג int המכיל מספר של גודל הרשומה בבתים.
- NrOfRecsInFile – שדה מסוג int המכיל מספר המציינ את מספר הרשומות בקובץ.
- KeyOffset – שדה מסוג int המכיל מספר המציינ את מיקום המפתח ברשומה.
- KeyType - מערך בגודל 2 מסוג char המכיל את טיפוס המפתח.
- KeySize - שדה מסוג int המכיל מספר המציינ את אורך המפתח בבתים.
- OverflowAreaPtr – שדה מסוג int המכיל מספר המציינ את המספר הסידורי של הבלוק בראש שרשרת הבלוקים הפנויים של אזור הגלישה.
- OverflowAreaStart - שדה מסוג int המכיל מספר המציינ את המספר הסידורי של הבלוק בראש שרשרת הבלוקים של אזור הגלישה.
- Filler - מערך בגודל 950 מסוג char המכיל את המידע על הקובץ.

מיני פרויקט בארגון וניהול קבצים מגשים: אליהו פולק, משה פוגל ועידן בן שמעון.

גודל כל המחלקה 1024 בתים כמתוכנן.

:Hashfile

(בחרנו כאמור שהמחלקה הזו תהיה המעטפת על המחלקה Physical File בשלב 0 שתהיה מופע במחלקה זו, דבר זה היה נראה לנו יותר טבעי שהתשתית הלוגית "תולבש" חיצונית על התשתית הפיזית ותשלים אותה "מעל" ולא הפוך).

:שדות (בשלב זה):

- UserName – שם המשתמש המבקש ליצור או לפתוח קובץ לקריאה/כתיבה.
- LogicalBuffer – שדה המשמש מצביע מסוג המחלקה LogicalBlock המצביע בפועל על החוצץ הפיזי אולם מכיוון שהוא מסוג LogicalBlock הוא מחלק לוגית את החוצץ עפ"י המבנה שלו בכך באה לידי מימוש התשתית הלוגית.
- Updateflag – שדה בוליאני שתוכנו מציין האם קריאת הבלוק הנוכחי הייתה לשם עדכון אם לא.
- LBuffChanged - שדה בוליאני שתוכנו מציין האם הבלוק הנוכחי השתנה מאז קריאת הרשומה האחרונה ממנו.
- LogicalFHBUFFER - שדה המשמש מצביע מסוג המחלקה LogicalFileHeader המצביע בפועל על החוצץ הפיזי של file header (בלוק מס' 0) אולם מכיוון שהוא מסוג LogicalFileHeader הוא מחלק בפועל את החוצץ לוגית עפ"י המבנה שלו. בכך ממומשת התשתית הלוגית על התשתית הפיזית.
- CurrRecNrInBuffer – שדה שמכיל מספר שלם וחיובי המכיל המספר הסידורי של הרשומה הנוכחית מתחילת הבלוק.
- LHBuffChanged – שדה בוליאני שתוכנו מעיד על שינוי שנעשה ב file header מאז שנפתח הקובץ (למשל תוספת רשומה, שינוי ברשומה, מספר הרשומות שגלשו גדל וכד').

:פונקציות:

(הערה כללית: המבנה של הפונקציות כאן מאד מזכיר את הפונקציות של שלב 0. אין תימה בכך, אין מטרה בשלב זה אלא להוסיף חלוקה לוגית לתשתית הפיזית הקיימת...)

- שלושה בנאים:
 1. בנאי ברירת מחדל: בנאי זה תחילה פונה לבנאי ברירת המחדל הפיזי כדי לאתחל את כל השדות הפיזיים, לאחר מכן הוא נכנס לתמונה ומעדכן את השדות השייכים לחלוקה הלוגית בערכי default, אין הוא מתקשר לשום קובץ ממש.
 2. בנאי יוצר: תפקידו לאתחל אובייקט מסוג Logical File ולקשר לו קובץ חדש פיזית ולוגית. הוא משתמש בבנאי יוצר של השכבה הפיזית ובנוסף משלים את כל ההוספות הלוגיות.
 3. בנאי פותח: תפקידו לאתחל אובייקט מסוג Logical File ולקשר לו קובץ קיים לפתיחה – פיזית ולוגית. הוא משתמש בבנאי פותח של השכבה הפיזית ובנוסף משלים את כל ההוספות הלוגיות.
- הורס: פונקציה המשמידה אובייקט קיים בלי להשמיד מסמך קיים, ע"י סגירת כל מסמך המקושר לאובייקט לוגית ופיזית.

מיני פרויקט בארגון וניהול קבצים

מגשים: אליהו פולק, משה פוגל ועידן בן שמעון.

- Hcreate - פונקציה שיוצרת קובץ לוגית ופיזית (משתמשת בפונקציה pcreate כמובן) תוך אתחול כל השדות הנחוצים לתשתית הלוגית (אחת הפעולות החשובות היא חלוקת המסמך לאזור מידע רגיל ואזור גלישה, כך שבלוקי אזור זה ישורשרו במספר סידורי רציף זה אחר זה).
- Hdelete - פונקציה המוחקת את הקובץ פיזית ולוגית. המחיקה הפיזית נעשית כמובן ע"י הפונקציה שבתשתית הפיזית, וכל שפונקציה זו עושה למעשה הוא, לעדכן שדות לוגיים נחוצים כתוצאה ממחיקה זו.
- Hopen - פונקציה הפותחת קובץ לקריאה / כתיבה פיזית ולוגית. הפתיחה בעיקרה נעשית ע"י השכבה הפיזית, הפונקציה בעיקר משנה שדות לוגיים המושפעים מהפתיחה. (כמובן שתהיה משמעות להרשאות בפתיחה לכתיבה).
- Hclose - פונקציה הסוגרת את הקובץ פיזית ולוגית. משתמשת בפונקציית הסגירה של השכבה הפיזית ובפונקציה הבאה:
- Flush - אחת מהפונקציות החשובות המשמשת כ manager/ווסת המחליטה איזה בלוק ייכתב למסמך עצמו. באמצעות קבלת קוד (0-כתיבת רק ה file header, 1-כתיבת הבלוק הנוכחי בלבד, 2- כתיבת בלוק ה file header והבלוק הנוכחי גם יחד). היא משתמשת כמובן לצורך הכתיבה בפונקציות מהתשתית הפיזית: writeBlock, writeFH.

דוח שלב 2:

בשלב זה אנו כבר מתחילים ליישם את מנגנון ה HASH ולפיכך נוספה כאן ספריה של 10 פונקציות hash שונות ע"מ לתת למשתמש אפשרות לממש את המנגנון ב 10 אפשרויות לפי בחירתו.

תחילה עודכנה המחלקה hashfile כך שתכיל מחלקה המכילה עפ"י פונקציית ה hash את החלוקה הפנימית של הקובץ (אזור כתיבה ראשי ואזור גלישה) בנוסף הוכנס שדה המכיל את הקוד של פונקציית ה hash המבוקשת ושדה במחלקה hashfile המכיל את הקוד המזהה של פונקציית ה hash.

(כתוספת משלנו הוספנו שדה חדש limit המכיל ערך המציין את המקום בבלוק שממנו א"א להוסיף רשומה נוספת על אף שיש עוד מקום בבלוק, כמובן שעריך זה תלוי בגודל הרשומה. כמו"כ הוספנו שדה המכיל קוד הנצרך עבור פונקציית העזר help seek שתוסבר בהמשך.)

כאן בעצם נוספו למחלקה פונקציות חדשות: חיפוש רשומה בבלוק, כתיבת רשומה לבלוק, וקריאת רשומה מבלוק.

כל אחת מפונקציות אלו בעיקרון כתובה 3 פעמים, ע"מ לאפשר את ביצוע המטלות הללו כאשר מפתח הרשומה יכול להופיע באחת משלושת הצורות הבאות: כ string, מערך דינמי char* או integer.

מכיוון שלאחר ההמרה של המפתח לצורה אחת (מערך דינמי char*) הפעולה חוזרת על עצמה בכל שלושת הפונקציות כפול שלוש מטלות, הוחלט להשתמש במדיניות ה"קופסאות", כלומר לבנות פונקציות עזר שעושות את אותן הפעולות שחוזרות על עצמן כך שהפונקציות המקוריות רק תעשינה את הפעולות המיוחדות להן ואילו את הפעולות שחוזרות על עצמן תשלחנה לפונקציית העזר המתאימה.

א"כ אלו הפונקציות שנוספו בשלב זה למחלקה hash file:

- Seek - ממירה ראשית את המפתח באם צריך, לאחמ"כ באם האובייקט כבר עומד על הבלוק המבוקש אזי מבצעת עליו חיפוש של הרשומה המבוקשת (ע"י פונקציית העזר help seek) באם לאו מביאה את הבלוק המבוקש ומבצעת עליו את החיפוש. כמובן שאם התברר שהייתה גלישה של רשומות מבלוק זה, יבוצע חיפוש באזור הגלישה.

מיני פרויקט בארגון וניהול קבצים

מגשים: אליהו פולק, משה פוגל ועידן בן שמעון.

- Read - פונקציה הקוראת רשומה מבוקשת מהקובץ כאשר היא משתמש בפונקציית ה seek באם היא נמצאה - או באזור הראשי או באזור הגלישה הבלוק שלה נקרא ל buffer ומתוכו נקראת הרשומה המבוקשת.
- Write - פונקציה הכותבת רשומה לתוך הקובץ כאשר היא משתמשת בפונקציית ה seek לוודא שאכן הרשומה לא קיימת (כאן נכנס לתמונה השדה write state המכיל קוד המודא אם הוא מגיע מפונקציית ה write . באם כן אזי באם לא נמצאה הרשומה לא תפיק הפונקציה seek הודעת חריגה אלא תודיע באמצעות קוד אחר לפונקציה שאכן הרשומה לא נמצאה כך שאפשר לכותבה.
- Helpseek - פונקציית עזר לפונקציית ה seek המבצעת בפועל את עבודת חיפוש הרשומה.
- Chack_if_key_is_in_the_block - פונקציית עזר לפונקציית ה seek אשר כשמה כן היא בודקת האם מפתח רשומה מסוים נמצא בבלוק.
- Help write - פונקציית עזר לפונקציית ה write המבצעת בפועל את עבודת הכתיבה של הרשומה לתוך הבלוק.
- Chack_free_soace_in_block - פונקציית עזר לפונקציית ה write אשר כשמה כן היא האם יש מקום פנוי לכתיבת רשומה בבלוק זה. (והדגש הוא מקום פנוי לרשומה לא סתם מקום פנוי שקטן מגודל רשומה וכאן נכנס לתמונה שדה ה limit המכיל את המידע הזה).
- Helpwrite - פונקציית עזר המחפשת מקום פנוי בתוך הקובץ, בתחילה בחלק הראשי לאחר קבלת מיקום מפונקציית ה HASH ובאם אין שם מקום חיפוש באזור הגלישה.

דו"ח שלב 3.

בשלב זה ממשנו פונקציות נוספות המשמשות למחיקת רשומה, עדכון רשומה, ופונקציה שעיקר תפקידה הוא להסיר חסימת כל פעולה אחרת כאשר התבצעה מכבר קריאה לצורך מחיקה או עדכון של רשומה.

כדרכנו גם כאן בנינו פונקציות עזר שונות לצורך ביצוע שלושת המשימות הנזכרות נפרט אותן ואת תפקידן אחת לאחת.

- Del rec Casing - פונקציית עזר המשמשת כפונקציית אב לצורך מחיקת רשומה, נאלצנו לבנותה מכיוון שהפונקציה שנתבקשנו לבנות בחוברת לא קיבלה כל ערך של מפתח (כיצד נוכל למחוק רשומה בלא לדעת מפתח לפחות?)
- פונקציה זו מממשת את המחיקה בפועל ולאחר מכן מבצעת דחיסה של הקובץ באופן פיזי בין בחלק הראשי של הקובץ ובין באזור הגלישה.
- Delrec - פונקציה ראשית המוחקת בפועל רשומה (באם קיימת כמובן) מתוך הקובץ וממלאה את מקום הרשומה בערכי זבל.
- Press Bloak - פונקציה המבצעת דחיסה של הבלוק באזור הראשי כאשר נמחקה ממנו רשומה.
- Press Block OV - פונקציה המבצעת דחיסה של הבלוק באזור הגלישה בזה אחר זה, כאשר במקום הרשומה שנמחקה מהאזור הראשי מועלת רשומה שהייתה באזור הגלישה למלאות את מקומה. כמובן שכל השדות נחוצי העדכון מעודכנים כגון: מספר הרשומות באזור הגלישה וכד'.
- Serch key by hash - פונקציית עזר ייחודית המקבלת מספר HASH ובאמצעותו מוצאת את מיקום הרשומה באזור הראשי, משמשת לעזור ולמצוא את מיקום הרשומה לצורך מחיקה\עדכון.
- Update - פונקציה ראשית המשמשת להחליף תוכן רשומה אחד באחר ע"י קבלת מפתח והרשומה המעודכנת. כמובן שישנה הקפדה שהמפתח יתאים לרשומה ולא שיוחלף תוכן של רשומה בעלת מפתח שונה.
- Updateoff - פונקציה ראשית שתפקידה להסיר את מצב החסימה הנוצר כאשר רוצים למחוק רשומה או לעדכנה, במצב כזה מופעלת חסימה של כל פעולה אחרת על הקובץ ונוצר צורך בהסרת החסימה. כמובן שתתבצע בדיקה האם מצב חסימה הופעל בכלל.

מיני פרויקט בארגון וניהול קבצים מגשים: אליהו פולק, משה פוגל ועידן בן שמעון.

הסבר לשלב הממשק הגרפי (GUI):

חלון הממשק הגרפי:



כפי שניתן לראות המערכת מחולקת לשלושה חלקים:

פעולות בסיסיות:



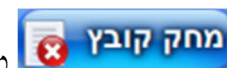
יצירת קובץ חדש, קבלת פרטים נחוצים מהמשתמש כגון: שם המסמך, בעלי המסמך, מיקומו ועוד. אופציה זו קיימת גם ב toolbar תחת הלשונית "קובץ".



פתיחת קובץ קיים (במידה ואכן קיים) שלושה סגנונות פתיחה: קריאה, כתיבה, שניהם. אופציה זו קיימת גם ב toolbar תחת הלשונית "קובץ".



סגירת קובץ קיים במידה והוא אכן פתוח. אופציה זו קיימת גם ב toolbar תחת הלשונית "קובץ".



מחיקת קובץ קיים במידה והוא סגור. אופציה זו קיימת גם ב toolbar תחת הלשונית "קובץ".

פעולות על הקובץ:



כתיבת רשומה לתוך הקובץ – קבלת פרטים מהמשתמש כגון: מפתח הרשומה, שם סטודנט, קורס וכדומה. אופציה זו קיימת גם ב toolbar תחת הלשונית "עריכה".

מיני פרויקט בארגון וניהול קבצים מגשים: אליהו פולק, משה פוגל ועידן בן שמעון.

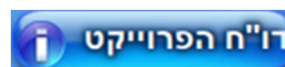


קריאת רשומה מקובץ במידה וקימת, בנוסף ישנה אפשרות לקריאה לצורך עדכון ושינוי הרשומה ע"י תיבת סימון "קריאה לצורך עדכון". אופציה זו קיימת גם ב toolbar תחת הלשונית "עריכה".



מחיקת רשומה במידה וקיימת מתוך הקובץ, אופציה זו קיימת גם ב toolbar תחת הלשונית "עריכה".

מידע על הקובץ:



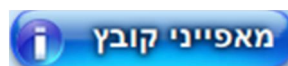
כפתור הפותח את הדו"ח הזה לקריאה (אם אתה קורא כאן סימן שלחצת על כפתור זה....), אופציה זו קיימת גם ב toolbar תחת הלשונית "עזרה".



כפתור החושף את רמת הפיזור של הרשומות בקובץ. אופציה זו קיימת גם ב toolbar תחת הלשונית "כלים".



קובץ רשומות הסטודנטים שבו השתמשנו במהלך הסמסטר לצפייה. אופציה זו קיימת גם ב toolbar תחת הלשונית "עזרה".



כפתור זה חושף את תכונות הקובץ כגון: שם המסמך, מספר הרשומות שבו, מספר הבלוק של תחילת אזור הגלישה ועוד. אופציה זו קיימת גם ב toolbar תחת הלשונית "כלים".



הצגת תפוסת המסמך בתצוגת פאי. אופציה זו קיימת גם ב toolbar תחת הלשונית "כלים".

בנוסף ישנם עוד שתי אופציות הקיימות רק ב toolbar :
יציאה - אפשרות יציאה מהתוכנית, תחת הלשונית "קובץ".
Save – שמירת הקובץ, תחת הלשונית "קובץ".
ביטול עדכון – שחרור מצב נעילה מחמת עדכון שנקטע או מחיקה, תחת הלשונית "עריכה".
אודות – אודות יוצרי הקובץ, תחת הלשונית "עזרה".

מיני פרויקט בארגון וניהול קבצים

מגשים: אליהו פולק, משה פוגל ועידן בן שמעון.

סיכום והמלצות:

בדו"ח זה מימשנו הלכה למעשה את עקרון ארגון ה HASH אשר נלמד בקורס התיאורטי בארגון וניהול קבצים סמסטר לפני כן.

היה מרתק לגלות כיצד הידע התיאורטי מומש הלכה למעשה במערכת מימוש הקבצים שבנינו על כך התודה למנחה שלנו ד"ר משה גולדשטיין שתמך גם בעזרה אישית במהלך הסמסטר.

עם זאת כדאי ומומלץ להרחיב בחוברת הקורס בדברים הבאים :

1. דוגמאות קוד קטנות המשולבות בחוברת הקורס לצד כל שלב (המתאימות לכל שלב) דבר שייתן המחשה מעשית בפרט בהתחלה כאשר החומר מאד מאד תיאורטי. (אומנם היו קבצי קוד שנוספו בחבילה נפרדת אבל זה מעט קשה כאשר החומר לא מסודר ויש צורך בחיפוש – דבר שגורם לתלמיד לפעמים להתייאש מראש)

2. כדאי להרחיב בהסבר מעשי מה הפירוש "התשתית הלוגית של הקובץ" נושא זה היה מעט עמום ומופשט, כדאי להסביר שבעצם מדובר בהלבשה של פוינטר מסוג מחלקה המחולקת בתוכה בחלוקת שדות מסויימת, על מחלקה שבה אין חלוקה זאת בתוכה וכו'.

3. ניכר כי נעשה מאמץ גדול בשלב הקישור בין שתי השפות שבהן כתבנו את הפרויקט, יחד עם זאת אנו חושבים כי כדאי לתת דוגמאות קוד יותר קרובות לנושא הפרויקט, להמחיש יותר כיצד להוסיף קובץ החושף את פונקציות ה C++ אל ה C# או קובץ המקבל פונקציות אלו מצד ה C# ועוד. יש לזכור כי גם אותם תלמידים שעשו את הפרויקט המקדים ב C#, לא התעסקו כמעט כלל בקישור בין שתי השפות, ולכך הנחיצות הגדולה.

המלצה נחמדה לפרויקט עתידי : אנו התעסקנו ברשומות בעלות אורך קבוע, יהיה מעניין לראות כיצד להתמודד עם רשומות בעלות אורך משתנה. היתרון כמובן ברשומות בעלות אורך משתנה הוא נצילות הקובץ ודינמיותו למול אורך קבוע.

ככלל היה פרויקט מעניין ומועיל מאד ועל כך שוב תודתנו לסגל המרצים שהכינו ועמלו להכין את הפרויקט וכל המידע הנלווה שסביבו.

מיני פרויקט בארגון וניהול קבצים
מגשים: אליהו פולק, משה פוגל ועידן בן שמעון.