

מייני פרויקט בארגון וניהול קבצים

מגישים: אורי אשל ודוד אוזן

תוכן עינינים

| | |
|---------|--|
| 3..... | דוח שלב 0 |
| 3..... | מבוא |
| 3..... | הגדרת המבנים המהווים בסיס לשכבה הפיסית |
| 3..... | Sector |
| 3..... | VolumeHeader |
| 4..... | DAT |
| 4..... | dirEntry |
| 4..... | EntryArray |
| 5..... | RootDir |
| 5..... | FileHeader |
| 5..... | המחלקה disk |
| 5..... | תיאור המשתנים של המחלקה disk |
| 5..... | תיאור הפונקציות של המחלקה disk |
| 6..... | דוח שלב 1 |
| 6..... | מבוא |
| 6..... | תיאור פונקציות הנוספות למחלקה disk |
| 7..... | דוח שלב 2 |
| 7..... | מבוא |
| 7..... | תיאור פונקציות הנוספות למחלקה disk |
| 8..... | דוח שלב 3 |
| 8..... | מבוא |
| 8..... | תיאור הפונקציות הנוספות למחלקה disk |
| 8..... | המחלקה FCB |
| 8..... | תיאור המשתנים של המחלקה FCB |
| 9..... | תיאור הפונקציות של המחלקה FCB |
| 10..... | דוח שלב 4 |
| 11..... | GUI |

דו"ח שלב 0

מבוא:

בשלב זה ממומשת התשתית הפיזית של הדיסק. ברמה הפיזית, מסתכלים על הדיסק כרצף של בלוקים - סקטורים - בעלי מבנה בסיסי, ומאורגנים לפי מספר סידורי של הסקטור בדיסק. בלי התייחסות לשום מבנה לוגי מעבר לתווית הזיהוי של הדיסק. לצורך כך הוגדרו המבנים FileHeader, RootDir, DAT, Volume Header, Sector והמחלקה disk.

הגדרת המבנים המהווים בסיס לשכבה הפיסית

Sector

יחידת קלט/פלט הבסיסית המייצגת חלוקת הדיסק, לכל סקטור המבנה הבא:

| שם השדה | משמעות |
|----------|-----------------------------|
| sectorNr | מספר סידורי של הסקטור בדיסק |
| rawData | נתונים |

יש סקטורים בדיסק המיועדים למידע מבני (Root Directory, DAT, Volume Header) וכל השאר מיועדים לנתונים. בדיסק יש 3200 סקטורים כאשר גודל כל סקטור הוא 1024 בתים.

VolumeHeader

סקטור המכיל מידע מבני אודות הדיסק ומהווה תווית זיהוי שלו. לתווית הזיהוי המבנה הבא:

| שם השדה | משמעות |
|----------------|---|
| sectorNr | מספר סידורי של הסקטור בדיסק |
| diskName | שם זיהוי הדיסק |
| diskOwner | שם בעל הדיסק |
| prodDate | תאריך יצור הדיסק |
| ClusQty | סה"כ יחידות הקצאה (clusters) בדיסק |
| dataClusQty | מספר יחידות הקצאה לנתונים בלבד. |
| addrDAT | כתובת הסקטור שמכיל את ה-DAT |
| addrRootDir | כתובת ה-cluster שמכיל את התיקייה הראשית (Root Directory) |
| addrDATcpy | כתובת הסקטור שמכיל עותק שני של ה-DAT |
| addrRootDirCpy | כתובת ה-cluster שמכיל עותק שני של התיקייה הראשית (Root Directory) |
| addrDataStart | כתובת ה-cluster הראשון בדיסק המיועד לנתונים. |
| formatDate | תאריך פירמוט |
| isFormatted | האם כבר מפורמט? (כן / לא) |
| emptyArea | שמור לשימוש עתידי |

גודל תווית הזיהוי הוא 1024 בתים, והיא נמצאת בסקטורים הראשונים בדיסק.

DAT

סקטור המכיל מידע מבני אודות הדיסק ומהווה "מפה" של כל היחידות ההקצאה (clusters) הפנויות בדיסק.

לסקטור DAT יהיה את המבנה הבא:

| שם שדה | משמעות |
|-----------|--|
| sectorNr | מספר סידורי של הסקטור בדיסק |
| DAT | רצף של 1600 ביטים המתארים את כל הclusters בדיסק. |
| emptyArea | שמור לשימוש עתידי |

גודל ה DAT הוא 1024 בתים, והוא נמצא בסקטורים הראשונים של הדיסק.

dirEntry

כניסה של התיקייה הראשית, כאשר התפקיד של הכניסה היא לייצג את הקובץ הנמצא בדיסק.

לכניסה כזאת המבנה הבא:

| שם שדה | משמעות |
|---------------|--|
| Filename | שם הקובץ |
| fileOwner | שם בעל הקובץ |
| fileAddr | כתובת הסקטור הראשון של הקובץ |
| crDate | תאריך יצירת הקובץ |
| fileSize | גודל הקובץ, כמספר סקטורים |
| eofRecNr | מיקום "רשומת" ה-end-of-file (המספר הסידורי של מיקומה מהתחלת הקובץ) |
| maxRecSize | אורך רשומה מרבי |
| actualRecSize | אורך רשומה בפועל |
| recFormat | שדה זה מסמל סוג רשומה כלומר, אורך קבוע או משתנה : "F" או "V" |
| keyOffset | Offset של התחלת המפתח בתוך הרשומה |
| keySize | אורך המפתח, כמספר בתים |
| keyType | טיפוס נתונים של ערך המפתח: "I" - מספר שלם (int) "F" - מספר ממשי (float) "D" - מספר ממשי כפול (double) "C" - מחרוזת תווים |
| entryStatus | שדה זה מעיד על מצב הכניסה הספציפית בתיקייה. המצב יכול להיות אחד מתוך שלושה: כניסה ריקה / כניסה פעילה / כניסה לא פעילה |

גודל כל כניסה הוא 72 בתים.

EntryArray

סקטור המהווה מבנה ביניים שישמש את התיקייה הראשית.

| שם שדה | משמעות |
|---------------|-----------------------------|
| sectorNr | מספר סידורי של הסקטור בדיסק |
| arr[14] | מערך של 14 כניסות של קבצים |
| emptyArea[12] | שמור לשימוש עתידי |

RootDir

התיקה הראשית של הקובץ הקשורה למבנה הדיסק. מכילה בתוכה את כל הכניסות של הקבצים הנמצאים בקובץ.

| שם שדה | משמעות |
|--------|-------------------------------|
| Part1 | חלק ראשון של הכניסות של הדיסק |
| Part2 | חלק שני של הכניסות של הדיסק |

גודל התיקה הראשית הוא 2 סקטורים והיא נמצאת בסקטורים הראשונים של הדיסק.

FileHeader

הסקטור הראשון של הקובץ, כאשר תפקידו הוא לשמש כתווית זיהוי של הקובץ.

| שם שדה | משמעות |
|-----------|--|
| sectorNr | מספר סידורי של הסקטור בדיסק |
| fileDesc | העתק של כניסתו של הקובץ בתיקיה |
| FAT | משתנה אשר מכיל מידע על מיקום הקובץ בדיסק |
| emptyArea | שמור לשימוש עתידי |

המחלקה disk

מחלקה זו מנהלת את הדיסק המדומה שאיתו נעבוד במהלך הפרויקט. מחלקה זו מכילה שדות המכילים מידע מבני על הדיסק וכן משתנים שמהם ניתן לעשות שינויים בדיסק.

| שם שדה | משמעות |
|------------------|---|
| vhd | שדה שמיועד להיות המקום בו ישמרו נתוני ה-Volume Header |
| dat | שדה שמיועד להיות המקום בו ישמרו נתוני ה-DAT |
| rootdir | שדה שמיועד להיות המקום אליו נתוני התיקיה הראשית (ה-Root Directory) |
| mounted | שדה בולאני שישמש כדגל: אם ערכו false, משמעותו שהדיסק המדומה עדיין לא זמין, אם ערכו true, משמעותו שהוא זמין. |
| dskfl | אובייקט מסוג fstream המייצג את הקובץ שמכיל את הדיסק המדומה. |
| currDiskSectorNr | שדה מסוג unsigned int שערכו הוא המספר הסידורי של הסקטור בדיסק שכרגע בחוצץ של קובץ מסוים. |

תיאור פונקציות המחלקה

disk()

פונקציה בונה דיפולטיבית המאתחלת את השדות של המחלקה ובנוסף מסמלת את דגל false=Mounted

disk

פונקציה בונה המקבלת פרמטרים בשביל שם הקובץ, שם בעל הדיסק, יוצרת דיסק חדש במידה והוא לא קיים ועושה Mount.

~disk

תפקידה של הפונקציה ההורסת היא לבצע Unmount לדיסק במידה וזה טרם התבצע.

createdisk

תפקידה של פונ' זו היא ליצור דיסק מדומה. שני הפרמטרים שהפונ' מקבלת הם שם ובעל הדיסק. הפונ' מעדכנת את כל פרטי הסקטורים המבניים של הדיסק, כמו כן היא כותבת לדיסק את כל הסקטורים שלו מלבד המבניים.

mountdisk

תפקידה של פונ' זו היא לפתוח את הקובץ המממש את הדיסק המדומה, היא קוראת את התוכן של הסקטורים המכילים את המידע המבני של הדיסק המדומה, לתוך השדות המתאימים במחלקה disk.

unmounddisk

תפקידה של פונ' זו היא לעדכן את כל הסקטורים המכילים מידע מבני של הדיסק המדומה ולסגור את הקובץ של בדיסק המדומה.

recreatedisk

תפקידה של פונ' זו היא לאתחל ממחדש תא הדיסק המדומה. כלומר הפונקציה תבצע את כל מה שתואר בקשר לפונ' יצירת הדיסק אבל בלי ליצור מחדש את הדיסק המדומה.

getdiskfl()

תפקידה של פונ' זו היא להחזיר מצביע מסוג fstream לכתובת של diskfl שמייצג את הקובץ שמכיל את הדיסק המדומה.

seekToSector

תפקידה של פונ' זו היא להתמקם בסקטור המבוקש בדיסק המדומה

writeSector

ישנם שני מימושים לפונ' זו אך התפקוד זהה. לכתוב מהחוצץ שאליו מצביע לתוך הסקטור המבוקש בדיסק המדומה.

readSector

ישנם שני מימושים אך התפקוד זהה. לקרוא תא הסקטור המבוקש מהדיסק המדומה לתוך החוצץ.

getCurrnetDate()

תפקידה של פונ' זו הוא להחזיר את התאריך הנוכחי.

דו"ח שלב 1

מבוא

בשלב זה ממומשת התשתית הלוגית של הדיסק שהיא שכבה מעל הרמה הפיזית. ברמה הלוגית מסתכלים על דיסק כרצף של סקטורים בעל מבנה שבא לידי ביטוי ע"י DAT שתאפשר לייצג ולנהל שטח פנוי לעומת שטח תפוס, והתיקיה הראשית שתאפשר לנהל קבצים ותת תיקיות בדיסק.

תיאור פונקציות הנוספות למחלקה disk

format

תפקידה של פונ' זו הוא לעשות פירמוט לדיסק המדומה. הפונ' מאתחלת ה DAT במקומות בהם יש מידע מבני אודות הדיסק (שאר המקומות מסומנים כפנויים), וכל הכניסות של התיקיה מהראשית מסומנות כריקות.

howmuchempty

תפקידה של פונ' זו להחזיר את מס' סה"כ ה-CLUSTERים הפנויים בדיסק המדומה.

alloc

תפקידה של פונ' זו לטפל הקצאת שטח (לקובץ) בדיסק. הפונ' מקצה זיכרון עבור קובץ תוך עדכון ה-FAT של הקובץ. הספציפי. הפונ' מקבלת 4 פמטרים:

הפרמטר הראשון מסוג DATtype דימוי FAT אליו מבקשים להקצות שטח.

הפרמטר השני הוא מספר חיובי שקובע כמה סקטורים מבוקשים להקצאה.

הפרמטר השלישי מייצג סוג האלגוריתם להקצאת השטח.

הפרמטר הרביעי (אופציונלי) מייצג את הערך ההתחלתי משם נתחיל לחפש את המקום הפנוי.

allocextend

תפקידה של פונ' זו לטפל בהוספת הקצאת שטח לקובץ בדיסק, הפונ' מרחיבה זיכרון עבור קובץ תוך עדכון ה-FAT של הקובץ הרלוונטי. הפונ' מקבלת את 3 הפרמטרים הראשונים שהמתודה alloc מקבלת. הפונ' מוצאת את הכתובת של הסקטור האחרון של הקובץ ושולחת אותו כפרמטר לפונ' alloc כפרמטר רביעי.

dealloc

תפקידה של פונ' זו לשחרר שטח תפוס ע"י קובץ מסויים. הפונ' דואגת לעדכן הן את ה-FAT של הקובץ והן את ה-DAT של הדיסק כולו.

דו"ח שלב 2

מבוא

בשלב זה מסתכלים על קובץ כרצף של סקטורים, ללא התייחסות לרמה הלוגית שבה מסתכלים על קובץ כרצף של רשומות.

תיאור פונקציות הנוספות למחלקת Disk

createfile

הפרמטר הראשון שם הקובץ
הפרמטר השני שם בעל הקובץ
הפרמטר השלישי קובע את סוג קובץ משתנה או קבוע
הפרמטר הרביעי אורך הרשומה
הפרמטר החמישי מספר הסקטורים הדרושים לקובץ
הפרמטר השישי טיפוס הנתונים של המפתח
הפרמטר השביעי מיקום התחלת המפתח בתוך הרשומה
הפרמטר השמיני אורך המפתח
הפרמטר התשיעי סוג האלגוריתם

delfile

פונקציה זו מבצעת מחיקה של קובץ קיים. הפונקציה משחררת את השטח שהוא תפס בדיסק ע"י מתודה dealloc. הפונ' מקבלת 2 פרמטרים את שם הקובץ ובעל הקובץ. רק בעל הקובץ יכול לבצע פעולה זאת.

extendfile

פונ' זו מוסיפה שטח לקובץ הקיים. הפונ' קוראת ל'allocextend עם סוג האלגוריתם ע"מ להקצות שטח נוסף הדרוש לקובץ. הפונ' מקבלת 4 פרמטרים: שם הקובץ, שם בעל הקובץ, מס' הסקטורים שרוצים להרחיב וסוג האלגוריתם.

דו"ח שלב 3

מבוא

בשלב הזה מסתכלים על קובץ כרצף של קבצים ורשומות. יתבצע ניהול פעולות קלט/פלט לקבצים, מתאפשרת פתיחת קבצים ופעולות בסיסיות כגון: קריאה, כתיבה, עדכון, מחיקה.

תאור הפונקציות הנוספות למחלקה Disk

openfile

מבצעת פתיחה של קובץ קיים תוך כדי עדכון ראש הקריאה/כתיבה למיקום הרלוונטי, ניתן לפתוח את הקובץ בפורמט של קריאה, כתיבה או קריאה וכתיבה, או בפורמט של הרחבה, הקובץ יהיה פתוח לקריאה לכולם, אך על מנת להרחיב או לכתוב בקובץ צריך להיות הבעלים. הפונקציה מחזירה מצביע למשתנה מסוג FCB שבו מעודכנים כל השדות הרלוונטיים של הקובץ, שכולל את תיאור מבנה הקובץ ואת הFAT של הקובץ.

findEOFSectorNr

בהנתן fileheader מחזירה את מספר הסקטור המכיל את הרשומה האחרונה של הקובץ

findEntry

בהנתן שם קובץ הפונקציה מחזירה את אינדקס הכניסה המתאים בrootdir. הפונקציה מחזירה גם את החלק בו הוא נמצא בתוך התיקיה הראשית

המחלקה FCB

FCB מאפשר לטפל בקובץ מבחינת התוכן שבו, כרצף של רשומות. למעשה, כל אובייקט מסוג FCB יאפשר לנהל את כל פעולות הקלט/פלט הלוגיות (לפי רשומות) הקשורות לפתיחה כלשהי של קובץ שבדיסק.

תאור המשתנים של המחלקה FCB

| שם שדה | משמעות |
|-----------------|---|
| d | מצביע לאובייקט מסוג Disk שמייצג דיסק מדומה. |
| fileDesc | העתק של ה-fileDesc של ה-File Header של הקובץ |
| FAT | FAT של הקובץ |
| Buffer | חוצץ קלט/פלט |
| currRecNr | מספר סידורי של הרשומה הנוכחית, בתוך הקובץ |
| currSecNr | מספר סידורי של הסקטור הנוכחי, בתוך הקובץ |
| currRecNrInBuff | מספר סידורי של הרשומה הנוכחית, בתוך הסקטור שבחוצץ |
| isOpen | משתנה שמציין אם הקובץ פתוח |
| format | משתנה שמציין את אופן פתיחת הקובץ. |
| lock | משתנה בוליאני שמציין האם יש רשומה נעולה בקובץ. |

תאור הפונקציות של המחלקה FCB

default constructor – FCB()

הבנאי של FCB יוצר אובייקט ויאתחל את כל השדות עם ערכים התחלתיים רלוונטיים.

constructor – FCB(disk *)

בנאי זה של FCB יוצר אובייקט מסוג FCB, השדה d יאותחל עם המצביע מסוג disk שהתקבל

כפרמטר, ויתר השדות יאותחלו עם ערכים רלוונטיים.

~FCB- destructor

closefile

פונקציה שתפקידה לסגור את הקובץ, ומרוקנת את החוצץ לתוך הקובץ כאשר הקובץ פתוח לכתובה, ומעדכנת את הכניסה הרלוונטית בתיקייה הראשית בשינויים שחלו בקובץ

flushfile

תפקידה של פונקציה זו לכתוב פיזית את תוכן החוצץ בחזרה לסקטור המתאים אם הוא עבר שינוי מאז אותו סקטור נקרא לתוך החוצץ.

read

פונקציה שתפקידה לנהל קריאה מהקובץ, ניתן לקרוא על מנת לעדכן ואז המערכת נכנסת למצב של הקפאה עד שמבצעים את הפעולה הרלוונטית, או שניתן לקרוא בלבד. הפונקציה מוודאת שהקובץ פתוח לקריאה, ושלא הגענו כבר לסוף הקובץ. בנוסף לכך היא מוודאת תקינות לוגית של הפרמטרים. כאשר קוראים רשומה במצב של קריאה בלבד, מתעדכנים השדות שמציינים את הרשומה הנוכחית בקובץ להצביע על הרשומה הבא.

write

תפקידה של פונקציה זו לכתוב את הרשומה שהתקבלה בפרמטר הראשון, לתוך המיקום הנוכחי של הקובץ.

seek

פונקציה שמעדכנת את השדות שמציינים את המיקום של הרשומה הנוכחית למיקום המבוקש. ניתן לקבוע את המיקום המבוקש ע"י מיקום יחסי מתחילת הקובץ/מיקום יחסי מסוף הקובץ/מנקודה כלשהי בקובץ. מתבצעת תקינות לוגית של הפרמטרים המתקבלים, והרשאה לביצוע הפעולה.

updateCancel

תפקידה של פונקציה זו לבטל את מצב הנעילה של הרשומה הנוכחית, שנוצר כתוצאה הביצוע של "קריאה לצורך עדכון".

deleteRec

תפקידה של פונקציה זו למחוק רשומה במקום הנוכחי בקובץ.

updateRec

תפקידה של פונקציה זו לכתוב רשומה במקום הנוכחי בקובץ, לצורך עדכון. הפרמטר הוא מצביע לרשומה שתכנית היישום מבקשת לכתוב כעדכון לרשומה הנוכחית.

findCurrnetSector

פונקצית עזר שתפקידה למצוא את הכתובת של הסקטור הנוכחי של הנתונים של הקובץ.

דו"ח שלב 3

בשלב הקודם יישמנו פונקציות המאפשרות לטפל בקובץ ברמת התוכן שבו: יכולנו לפתח ולסגור קובץ באופנים שונים, יכולנו לקרוא ולכתוב רשומות בקובץ, יכולנו "לטייל" בין רשומות של הקובץ, ויכולנו לעדכן וגם למחוק רשומה. פרט לפונקציה `openfile` ששייכת למחלקה `disk` ומחזירה אובייקט מסוג `FCB` מייצג קובץ פתוח, כל יתר הפונקציות שכתבנו בשלב הקודם הן מימוש של פעולות שניתן לבצע על התוכן של קובץ. לכן, טבעי שכל אלה תהיינה שייכות למחלקה `FCB`. כרגע יש לנו את כל מה שצריך על מנת להשלים את בניית המערכת שלנו לניהול דיסק מדומה, לצורך כך הגדרנו שכבה נוספת, המחלקה `DMS`.

מחלקת `DMS (Disk Management System)`.

למחלקה `DMS` תפקיד מרכזי: ישנו מערך של אובייקטים מסוג `FCB` שיאפשרו לטפל במספר קבצים כמספר האיברים במערך. כל איבר במערך הזה מייצג קובץ פתוח במערכת. כפי שלמדנו בקורס התיאורטי.

שדות

`fcfArray` - מצביע לאובייקט מסוג `FCB` שישמש למערך של אובייקטים מסוג `FCB` מיישם דיסק מדומה.

`fcfArrSize` - אורך המערך.

פונקציות

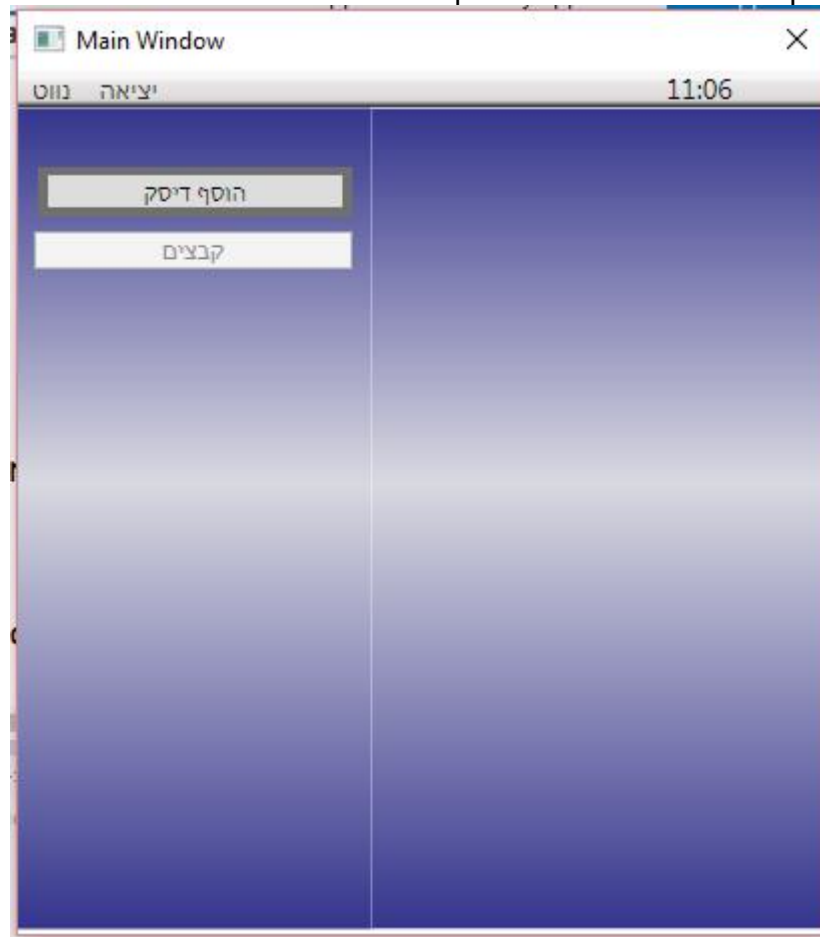
- **`FCB *openfile(disk*, string &, string &, string &)`**
הפרמטר הראשון הוא מצביע לאובייקט מסוג `disk` שמבקשים לפתוח קובץ שבתוכו. יתר הפרמטרים הם בדיוק אותם אלה שבפונקציה המקבילה במחלקה. תפקידה של פונקציה זו לבצע פתיחה של קובץ. המצביע מסוג `FCB` שמוחזר על ידי הפתיחה בפועל יוחזר על ידי הפונקציה הזאת.

- **`FCB *lookforfcf(disk*, string &)`**
הפרמטר הראשון הוא מצביע לאובייקט מסוג `disk` שמבקשים לפתוח קובץ שבתוכו. הפרמטר השני הוא שם של קובץ. תפקידה של פונקציה זו לסרוק את המערך `fcfArray` ולהחזיר אובייקט `FCB` פנוי. לפני שהפונקציה מחזירה אובייקט כזה היא חייבת לבדוק שקובץ בעל אותו שם, באותו דיסק, לא פתוח. אם הכל תקין אבל אין איבר פנוי, הפונקציה תחזיר מצביע ריק.

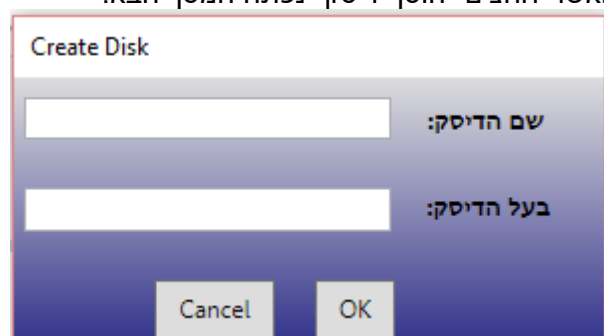
GUI

בשלב הזה מממשים את הממשק הגרפי המאפשר לנהל את הקובץ. ביצענו זאת באמצעות WPF וכתובת קוד בשפה סי שארפ

מסך הפתיחה הראשי של הממשק הגרפי



יש שני כפתורים, הכפתור הראשון מאפשר הוספה של דיסק ואילו הכפתור השני מאפשר גישה לקבצים שיש בדיסק (במידה וקיים דיסק) כאשר לוחצים "הוסף דיסק" נפתח המסך הבא:



על המשתמש כעת להכניס שם לדיסק ושם לבעלים של הדיסק לאחר מכן לחיצה על OK תצור דיסק ותעשה לו פרמוט.

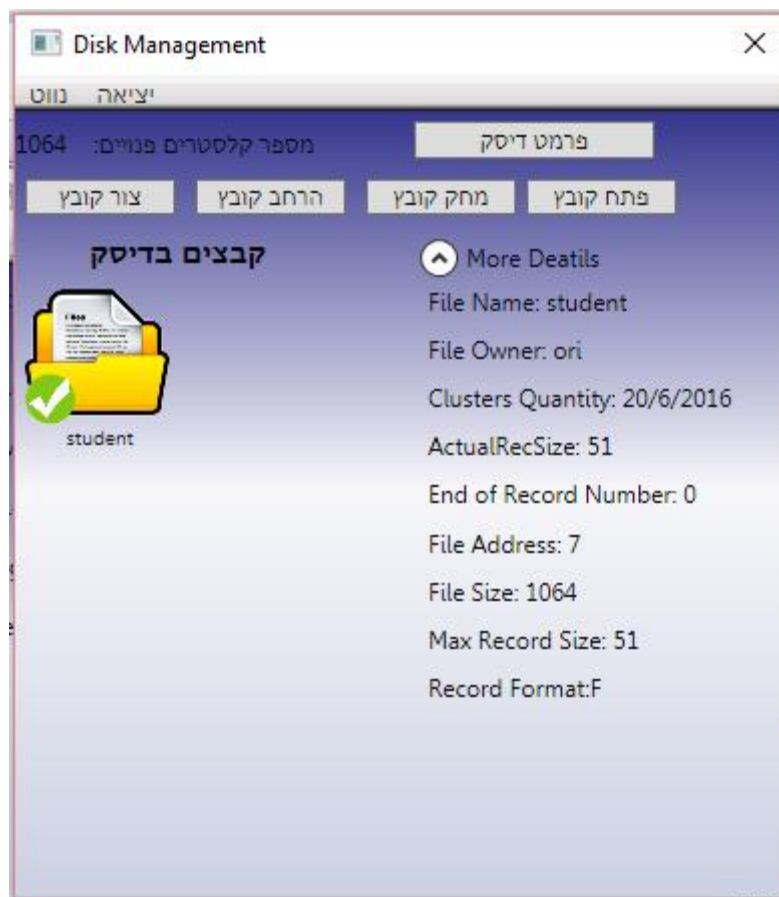


כאשר יוצרים דיסק נוצר user control שברגע שלוחצים עליו מופעל event שמסמן את הדיסק שעליו אנו רוצים לעבוד. כמו כן נפתח expander המכיל פרטים אודות הדיסק שזה עתה סומן.



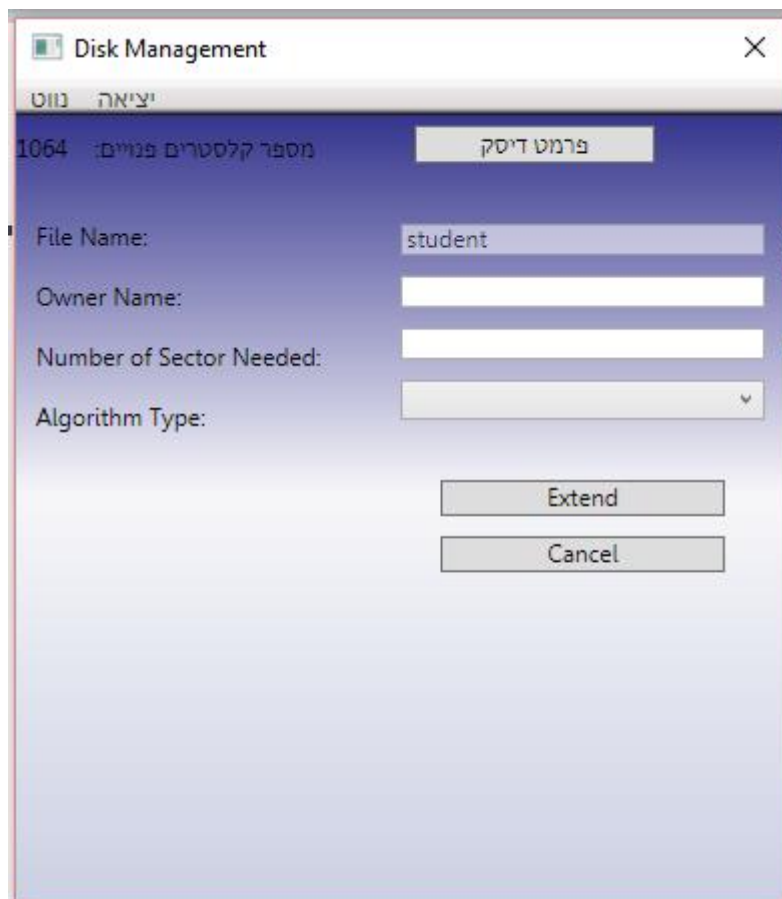
כעת נפתח מסך ניהול קבצים של הדיסק שנבחר. כיוון שאין קבצים האופציה היחידה שיש לעשות היא ליצור קבצים חדשים. כמו כן בכל שלב ניתן לפרמט את הדיסק בהכנסת שם בעלים.

לאחר לחיצה על "צור קובץ" המסך משנה את התצוגה כך שהוא מראה אפשרויות להוספת קובץ חדש כאשר על המשתמש להכניס את שם הקובץ וכו'. בנוסף על המשתמש לבחור את סוג האלגוריתם ואת סוג הרשומה שתהיה בקובץ (first/best/worst)

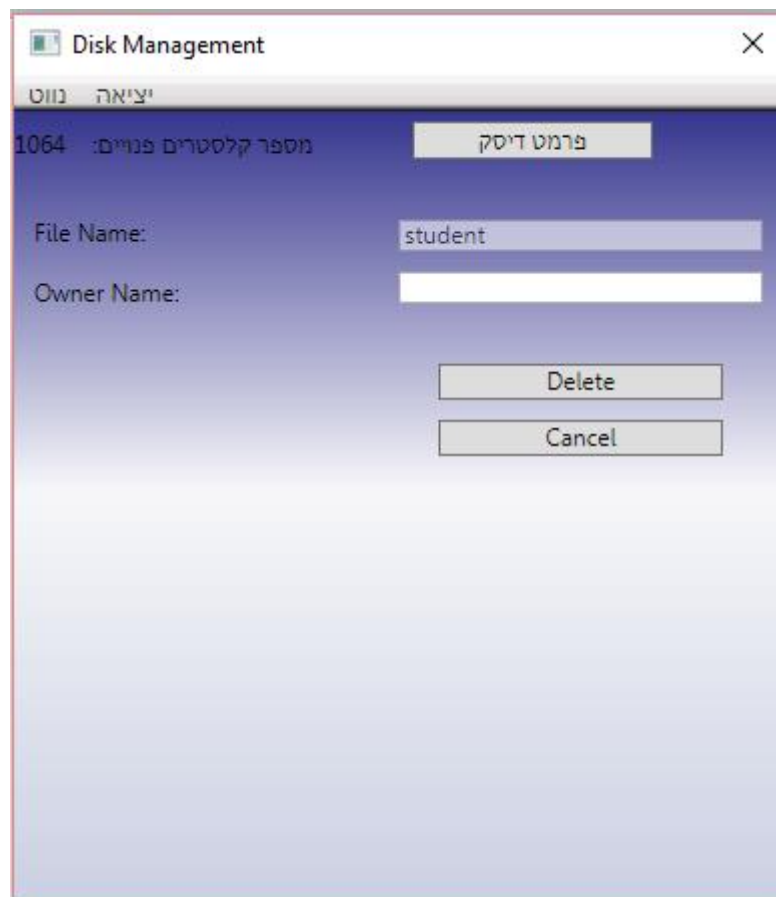


לאחר ההוספה ניתן לראות שיש user control עבור כל קובץ שנוצר, כשלחיצה עליו תתן לי את פרטיו בexpander ותאפשר לעשות פעולות על הקובץ.

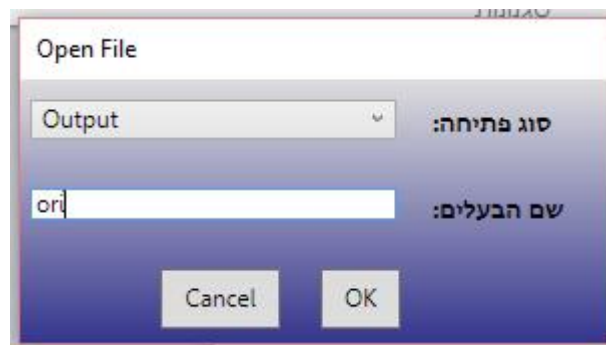
דוגמא למסך הנפתח כאשר לוחצים על "הרחב" קובץ:



דוגמא למסך הנפתח לאחר לחיצה על "מחק קובץ"



כעת נעבור למסך ניהול רשומות שעל הדיסק ע"י לחיצה על "פתח קובץ"




כאשר החלון נפתח ל'Output' תנתן אפשרות להוסיף רשומות לתוך הקובץ.

Files Management

יציאה

נווט



File Name: student

Open Type: Output

Record Type: Student

ID:

Address:

Date of Birth:

בחר תאריך

15

Add

Clear Fields