

Excercise

No 1

תרגיל בית מס' 1

נושא: מיני-טרמינל למערכת קבצים עם תמייה ב-*reference counting*

דדרין: מוצ"ש, 06/05/2023, 23:59

הגשה ביחידים

בהצלחה רבה!

תיאור התרגיל

תרגיל זה עוסק במערכת קבצים משודרגת, בה לאובייקטי קבצים יש פונקציונליות מתقدמת ותמייה ב-*reference counting*. מעבר לשדרוג טיפוס הקובץ הסטנדרטי הקיים בשפה כבירת מחדל `fstream`: `std::fstream`, יהיה עליו להציג עולם-מונחה עצמים של מערכת קבצים על כוון וירטואלי המוגדרת באמצעות מבנה היררכי של תיקיות.

חלק א': אובייקט קובץ משודרג בעל *Reference Counting*

בחולק זה של התרגיל עליהם למשמש אובייקט קובץ משודרג תחת דרישת למימוש של המרכיבים הבאים:

- [1] אופרטור [] לקריאה בלבד של تو בודד בגישה אקראית; פונקציה זו תחזיר את התו הכתוב בקובץ נתון ע"פ מיקומו. באם המיקום חורג מגבולות הקובץ הקיים, תיזרק חריגה.
- [2] אופרטור [] לכתיבה של تو בודד בגישה אקראית; פונקציה זו תכתב تو נתון בקובץ במיקום נתון. באם המיקום חורג מגבולות הקובץ הקיים (החל מאינדקס אחד לאחר סוף הקובץ), תיזרק חריגה.
- [3] פונקציית "נגעה בקובץ" בשם `touch`; פונקציה זו تعدכן את חתימת הזמן של קובץ במערכת הקבצים ע"פ שם שיונן לה ארגומנט, או **תיצור קובץ ריק בשם זה, באם לא היה קיים**. בכל מקרה, פונקציה זו אינה משנה את תוכנו של הקובץ.
- [4] פונקציית העתקה קובץ בשם `copy`; פונקציה זו תעתק את תוכנו של קובץ מקור לקובץ מטרה, ע"פ שמותיהם הנתונים.
קובץ המתරה אינו בהכרח קיים לפני הרצת הפקודה.
- [5] פונקציית מחיקת קובץ בשם `remove`; פונקציה זו תמחוק את הקובץ הנוכחי ממערכת הקבצים.
- [6] פונקציית העברת קובץ בשם `move`; פונקציה זו תעתק את תוכנו של קובץ מקור לקובץ מטרה, ע"פ שמותיהם הנוכחיים, ולאחר מכן תמחוק את קובץ המקורי. **קובץ המתרה אינו בהכרח קיים לפני הרצת הפקודה.**
- [7] פונקציית הדפסה בשם `cat`; פונקציה זו תדפיס את כל תוכנו של הקובץ.
- [8] פונקציית מנית שורות/מילים/תווים בשם `wc`; פונקציה זו תמנה ותדפיס את מספר השורות, מספר המילים ומספר התווים הכלול בקובץ (ההדפסה תוכל את שלושת הערכים בשורה אחת של פלט).
- [9] פונקציית יצרת "קישור חזק" בשם `hard link` לאובייקט קובץ אחר (אשר בהכרח קיים); הקישור עצמו יתנהג כקובץ לכל דבר.

בנוסף לדרישות אלו, נתונים אילוצי המימוש הבאים:

1. אובייקט הקובץ מוגדר באמצעות **עטיפת מצביע לтипויים** `fstream` `std::`, אשר בהכרח יוקצה דינאמית על ה-*heap*.
2. על אובייקט הקובץ לתמוך במנגנון *reference counting* (RC) הממומש על-ידייכם.
3. כל שגיאה הנובעת משימוש בקבצים (למשל, אי-קיים של קובץ המושא לפעולה), בנוסף לשגיאות המפורשות של האופרטורים [2]+[1], צריכה להיות מטופלת באמצעות מנגנון חריגות מבנה שיתוכן וימומש על-ידייכם.

נושאים מתקדמים בתכנות מונחה עצמים, אביב 2023

חלק ב': אובייקט תיקייה קבצים וירטואלית

בחלק זה של התרגיל עליכם למשמש אובייקט תיקייה קבצים וירטואלית תחת דרישת למימוש של המרכיבים הבאים:

- [10] פונקציית יצירת תיקייה בשם `zmkd`; אם כבר קיימת תיקייה בשם זה, הדבר יחשב לשגיאה.
- [11] פונקציית שינוי תיקייה העובודה הנוכחיית בשם `chdir`; אם לא קיימת תיקייה בשם זה, הדבר יחשב לשגיאה.
- [12] פונקציית מחיקת תיקייה בשם `zrmkdir`; פונקציה זו תמחוק את כל תוכנה של התיקייה הננתונה ותתי-התיקיות שלה, ותסיר אותה לחלוון הווירטואלי (שימו לב כי פקודה זו שונה בתוצאתה מפקודת המערכת המקבילה בינויקס).
- [13] פונקציית הדפסת תוכן התיקייה במיקום העובודה הנוכחי בانون הווירטואלי בשם `ls`; פורמט הדפסה כרצונכם.
- [14] פונקציית הדפסה בשם `zootoot`; פונקציה זו תדפיס את כל שמות הקבצים והתיקיות בanon הווירטואלי, בהירארכיה הנכונה, כאשר על ערך המונה (*reference counter*) של כל קובץ להופיע בסמוך לשם הקובץ (פורמט הדפסה כרצונכם).
- [15] פונקציית הדפסת מיקום עובודה הנוכחי בanon הווירטואלי בשם `pwd`; פורמט הדפסה כרצונכם.

הערות:

1. אובייקטי התיקיות קיימים בanon הווירטואלי בלבד ואין/am أمرורים להשתקף במערכת הקבצים עלייה מורצת התכנית (כלומר, בגיןוד לקבצים, התיקיות הן וירטואליות לחלוון); בהתאם, פקודות התיקיות רלוונטיות רק ביחס לanon הווירטואלי.
2. שמה של תיקייה בanon הווירטואלי מוגדר באמצעות מיקומה האבסולוטי במערכת הקבצים; **תיקיית הבסיס נקראת /**.
3. אתם רשאים להסתמך על אובייקט הקובץ בהגדרת אובייקט התיקייה (כמו במערכות יוניקס).
4. מערכת הקבצים אינה רשאית לאפשר "קישור חזק" עבור תיקיות (בגיןוד לקבצים).
5. אתם חופשיים לבחור מבני נתונים כרצונכם לצורך אחסון מערכת התיקיות והקבצים.

חלק ג': מיני-טרמינל לפקודות קבצים ותיקיות

בחלק זה של התרגיל אתם תמשחו תכנית המדממת סביבת טרמינל עלanon וירטואלי המוגבלת למספר פקודות קבצים ותיקיות, בהתאם על טיפוס הקובץ המשודרג וטיפוס התיקייה שימושיתם בחלקים א'–ב'.

הרעיון הוא שבharצת התכנית שלכם, יפתח ערך קלט חופשי להזנת פקודות מצד המשתמש, בהסתמך על הפקודות [1–15] שלעיל. התchipיר החוקי המתאים לפקודות [1–15] בתכנית שלכם מותואר באמצעות המפרט הבא, הכולל את פקודת הייצה [16]:

```
[1]      read FILENAME POSITION
[2]      write FILENAME POSITION CHARACTER
[3]      touch FILENAME
[4]      copy SOURCE_FILENAME TARGET_FILENAME
[5]      remove FILENAME
[6]      move SOURCE_FILENAME TARGET_FILENAME
[7]      cat FILENAME
[8]      wc FILENAME
[9]      ln TARGET_FILENAME LINK_NAME
[10]     mkdir FOLDERNAME
[11]     chdir FOLDERNAME
[12]     rmdir FOLDERNAME
[13]     ls FOLDERNAME
[14]     lproot
[15]     pwd
[16]     exit
```

הקלדת כל אחת מן הפקודות צריכה לכלול את שם הפקודה והารגומנטים הרלוונטיים, מופרדים ברווח, ולסיום ENTER. שימושם לב שפקודות הקבצים [15–13, 1–8] הם בעלות פלט, אשר צריך להיות מודפס לערך הפלט הסטנדרטי מיד לאחר הריצתן; כל שאר הפקודות נטולות פלט, והתכנית אמורה לשוב למסך האזנה לקלט מן המשתמש מיד לאחר ביצוען.

נושאים מתקדמים בתכנות מונחה עצמים, אביב 2023

במקרה של חירגה מן הפורוטוקול המתוואר, או במקרה של שגיאת קבצים / זמן-ריצה, על התכנית להדפיס הודעה שגיאה מתאימה ולחזור למצוב האזנה לקלט מן המשתמש.

הרצצת התכנית

לאחר קומpileציה של התכנית לכדי קובץ הרצה בשם myVFS, הרצתו בטרמינל תפתח עירוץ האזנה לקלט חופשי מן המשתמש עבור מערכת הקבצים ביחס לתיקיית ההרצה; להלן דוגמא לרצף אפשרי של קלטים:

```
[terminal] $ ./myVFS
[input]     pwd
[output]   V/
[input]     touch V/test.txt
[input]     write V/test.txt 0 d
[input]     read V/test.txt 0
[output]   d
[input]     write V/test.txt 1 e
[input]     write V/test.txt 2 J
[input]     write V/test.txt 3 o
[input]     write V/test.txt 4 n
[input]     write V/test.txt 5 g
[input]     cat V/text.txt
[output]   ERROR: no such filename
[input]     cat V/test.txt
[output]   deJong
[input]     mkdir V/tmp/
[input]     chdir V/tmp/
[input]     pwd
[output]   V/tmp/
[input]     ls V/tmp/
[output]   V/tmp/:
[input]     copy V/test.txt V/tmp/t.dat
[input]     copy FS.cpp V/tmp/f.cc
[input]     ln V/tmp/f.cc V/tmp/FS.cpp
[input]     lroot
[output]   V/:
[output]   test.txt 1
[output]   V/tmp/:
[output]   f.cc 2
[output]   t.dat 1
[output]   FS.cpp 1
[input]     rmdir V/tmp/
[input]     lroot
[output]   V/:
[output]   test.txt 1
[input]     exit
[terminal] $
```

נושאים מתקדמים בתכנות מונחה עצמים, אביב 2023

dagshim

אתם כמובן נדרשים לפתור בעל דיזיין מונחה-עצמים. שימו לב ליחסים בין הישויות השונות - יש כאן ישות מנהלת, שהיא איננה אובייקט הקובץ. "העולם" מיוצג באמצעות המיני-טרמינל שתבנו, וימומש באמצעות אובייקט "CONN" וירטואלי" או "מערכת קבצים אלטרנטיבית". למשל, בעניין סוגיות האחריות לפירוק אובייקט הקובץ המשודרג - הוא יפרק ע"י ישות אחרת, או אפילו ע"י העולם, ולא ע"י עצמו (כך, למשל, גם ע"י יצירתו באמצעות פקודה touch). לכן, מבחינה טכנית של הגדרות התרגיל – רשימת פעולות הקבצים [1-8] לא בהכרח תמורה לקובץ הcotor של אובייקט הקובץ.

תרגילים זה מומלץ לעבוד בסביבה מבוססת-יוניקט.

במקרים של אי-תקינות בעבודה מול המשתמש (למשל, קבצים שאינם קיימים), או במקרה של שגיאות זמן-אמת, יש להפעיל מגנון חריגות בשכבת אובייקט הקובץ, לתפועו אותו בשכבה המיני-טרמינל, ולהציג למשתמש הודעות שגיאה מתאימות. בכל מקרה, מצב בו חריגה אינה נתפסת ובאייה לкриיסת התכנית תיחשב לשגיאה.

אתם מוזמנים להשתמש בספריית STL על שלל מבני הנתונים, האיטרטורים והאלגוריתמים שלה.

- בתרגיל זה **איןכם** רשאים להשתמש במצביים החכמים הסטנדרטיים של תקן C++: `unique_ptr`, `shared_ptr`, `weak_ptr`.
- אין להשתמש בפקודות מערכת לשםימוש פקודות הקבצים (למשל בפעולות פקודת system), אך אתם בהחלט מוזמנים לבסס את שלב הבדיקות שלכם, לפני ההגשה, על פקודות המערכת.
- עליכם לוודא כי התכנית עוברת קומpileציה ++ג' התואמת את הקומpileר שעל שרת המכללה לא כל שגיאות או זהירות כלשהן, ורצה בהצלחה.
- עליכם לתעד את הקוד באמצעות הערות המתארות בקצירה את הפונקציות השונות.

הגשה

- הכנו קובץ בשם `txt`. README הכולל את שם ותעודת הזהות של הסטודנט/ית המגיש/ה; בקובץ זה אתם מוזמנים לכלול הערות ותיעוד כללי לגבי המימוש שלכם.
- עליכם להגיש במערכת Moodle קובץ ארכיב `tar/zip/rar` בשם `xxx.1x` המכיל את כל קבצי המקור (מסמך/מימוש הנדרשים לקומPILEציה, וכן את הקובץ `txt`. README).

אי-הקפדה על ההנחיות תגورو הורדה בציון התרגיל. לא תתקבלנה הgeshot באיחור!

Music

1c 156

የኢትዮጵያ የቅርቡ በኋላ የቅርቡ ስራውን ጥንቃቄ ነው እና አንድ የቅርቡ በኋላ የቅርቡ ስራውን ጥንቃቄ ነው እና አንድ

לעומת נסיבות מינימליות, מילויים מינימליים יוצרים מושג של איזון וריבוי.

? נוֹנָן נְסֵן נִי - "נְגַדֵּל וְנִזְהַר" (ט) . 3

• (תוקף גזע) בגדים מודפסים - גלימתם על גוףם

four flip 2018 code My File DASH myself 2013

. (נִזְבֵּחַ שָׁמֶן) בְּרוּ לְכָה וְיִגְעַן

שלום לסטודנטים בנושאים מתקדמים,

להלן התייחסות לעניינים שונים ולשאלות הקשורות עם חלק מהסטודנטים - שימושם כהברות וdagשים לגבי התרגילים:

* המושג "ירוטואלי" מתיחס כאן כמושג הפור ל"פיס" (אין קשור לפולימורפים!). הרעיון הוא לא ליצור תחזקה אצל המסתמש במני-טרמינל לפיה קיים כוון לעילו מtbodyות פועלות פיסיות, למחרת שבפועל לא קיים כוון כזה, אלא מעתפת שנתמכת בעזרת הפוקודות. לדוגמה, ניתן ליצור "ירארכיה ירוטואלית" עמוק של 10 תקיות, ובמה הרבה קבצים חדשים/מוסתקים, אבל למעשה הירארכיה כזו לא תהיה קיימת פיסית בדיסק, אלא רק מתחדשת ומונחת ע"י התכנית. לשם כך ניתן להיעזר במבני נתונים בסיסיים (וקטור, מפה), וכן צורך למשוך עץ קבצים אמיתי.

** כמו כן, אין דרישת לצור קבצים אמיתיים (ניתן להשתמש באובייקטים אחרים ולשמור תוקן קבצים בזיכרון), אך שימו לב שיש לאפשר עבודה עם קבצים קיימים במערכת הקבצים (ראו דוגמא את הבדיקה מול FS.cpp). בפועל, בסיסים כל הרצאה לבלתי יוצרים קבצים בתיקיית ההרצה הלוקאלית, בהם היה שימוש במהלך הרצאה. אין להם משמעות בתום הרצאה, ויש לנקוטם.

```
ש (כפונקציה גלובלית) בה תוכלו  
void ftouch(const char* file){  
    ofstream xfile;  
    xfile.open(file);  
    xfile.flush();  
    xfile.close();  
}
```

שימוש לבש החתונאות משתנה על גבי פלטפורמות שונות באופן ד' משמשות -- ולכן שימוש בקוד הנוכחי יספיק בתרגיל זה.

**** על הפקודה זו לבצע קישור (ניתן ל��וא עליה בתייעוד סטנדרטי של UNIX), שם היא דזוקא נחשבת לקישור רר). בפקודה זו למשה מתרחש השימוש המשמעותי היחיד של מגנון-h-RC - בשאר הפקודות המנגנון פעיל, אבל מקבל ערכיהם של אפס או אחד

הפקודה יכולה לשתף קובץ במקום פיזי או וירטואלי להישור במקום וירטואלי בלבד (הכינוי השני אינו נדרש, מפני שהוא יכול לאלץ את מערכת ההפעה לבצע קישור חזק במקומות פיזיים ללא שימוש במנגנון system). מהבחינה הזאת, מדובר בפקודה שלולוונית לקישורים בעולם הווירטואלי בלבד. אם קיים כבר קישור בשם זה, לא ניתן לדרכו אותו (כלומר, תהיה התעלמות מהפעלת הפקודה או שמאצא בהדעת שיבאה – לרבותה)

אנו חווים נקל' פרשנויות לפחות זו – סימטרית וא-סימטרית – בהקשר לדוגמא שניתונה בהנחות התراجיל, תתקבל תגנוגות לפחות לפיה המטפים ימי עלי' עריכים פומטריים 2,2, וגם הטענה מומלצת בהנחתה בה' בעל' עריכים 1,2.

classes

My File :

sub classes/structs :

struct BC :

Fields :

- fstream file
- int count
- string fileName
- string data
- size_t fileSize

Methods :

- BC(string& fileName)
- operator++(); // ++count
- operator--(); // -count

class IndexOutOfRangeException / class OpenFileException :

public exception :

Fields :

- string message

Methods :

- IndexOutOfRangeException / OpenFileException (string message)
- const char* what()

Fields :

- BC* fileBC
- string name

Methods:

- MyFile()
- MyFile(const string& name)
- MyFile(const MyFile&)
- ~MyFile()
- MyFile& operator=(const MyFile&)
- char& operator[](int offset)
- const char& operator[](int offset) const
- friend ostream& operator<<(ostream&, MyFile&)
- void wc() const
- void cat() const
- void operator++()
- void operator--()
- int getBcCount() //BC count
- writeToFile()
- MyFile& operator>>(MyFile& o) const //copy this to
- bool operator<(const MyFile&) const
- const string& getFileName() const

Directory:

sub classes/structs:

DirectoryAlreadyExistException : public exception :

Fields:

- string message

Methods :

- DirectoryAlreadyExistException(string message)
- char* what()

Fields :

- string fullName // "d₁ / d₂ / d₃ / this"
- string name
- Directory* parent
- set<Directory> inDirectories
- set<MyFile> Files

Methods :

- Directory(string name);
- Directory(const Directory&)
- mkdir(string name) throw DirectoryAlreadyExistException
- rmdir()
- getName() const
- getFullName()
- getInDirectories()
- getInFiles()

- get Parent()
- void printContent() // print all the files and dirs from this
- printInDirectories()
- printInFiles()
- bool operator< (const Directory&) const
- friend ostream& operator << (ostream&, Directory&) // full name
- bool operator == (const string&) const
- bool operator == (const Directory&) const
- bool operator != (const string&) const
- bool operator != (const Directory&) const

VirtualDrive:

sub classes/structs:

FileNotFoundException/DirectoryNotFoundException : public exception :

Fields:

- string message

Methods :

- FileNotFoundException/DirectoryNotFoundException (const string)
- const char* what()

Fields:

- Directory* homedirectory // V
- Directory* currentDirectory
- deque<string> dirs
- string fileToFindName
- string dirToFindName
- deque<MyFile> systemFiles
- bool systemFile

Methods :

- VirtualDrive();
- VirtualDrive(const VirtualDrive&) = delete
- VirtualDrive(VirtualDrive&) = delete
- ~VirtualDrive()
- VirtualDrive& operator=(const VirtualDrive&) = delete
- VirtualDrive& operator=(VirtualDrive&) = delete

- void read (const string & fileName)
- void write (char c, int offset)
- void touch (const string & fileName)
- void copy (const string& source , const string& dest) //only MyFile
- void remove (const string& myFileName) //only MyFile
- void move (const string& source , const string& dest)
- void cat (const string& myFileName)
- void ln (const string& source , const string& dest) //only MyFile
- void mkdir (const string & dirName)
- void chdir (const string & dirName)
- void rmdir (const string & dirName)
- void ls (const string & dirName) // current location
- void lproot ()
- void pwd ()
- void wc (const string&)
- void chdir (const& string)
- void splitArgs (const string&)
- MyFile& getTheFile ()
- Directory& getTheDirectory ()
- void reset ()

Terminal :

Fields:

- Virtual Drive vd
- map < string , int > commands

Methods:

- Terminal ()
- static void printMenu ()
- void run ()
- static bool checkNArgs (int argsNum, string args)
- void tillCommands ()