

Könyvtár

Főmenü:

A Könyvtár program amit választottam, angol nyelvű menüvezérelt program (magyar betűkkel, vagyis nem ASCII betűkkel nehezebb dolgozni azért angol nyelvű). A menüben üdvözljük a felhasználót, és megmutatjuk a különböző lehetőségeket/funkciókat amelyek közül lehet választani. Ezt egy hátul tesztelő ciklus és egy switch-case szerkezet használatával lehet megvalósítani. Beolvassuk a felhasználói inputot (ami valós értékű), és belépünk abba a függvénybe, amit a felhasználó választ. Ha nem helyes inputot beolvas a program, akkor leírjuk hogy az nem valid, és újra kell próbálkoznia.

Függvények:

Ez a program képes egy könyvtár adatainak az eltárolására, új könyvek létrehozására, a régiek módosítására, törlésére, a könyvek szerző, cím, kiadási év és téma alapján való keresésére. Ez lesz a program első része. A másik része pedig az olvasók való információ kezelése. Mint például: kölcsönzés adatainak kezelésére (melyik könyv épp kinél van), megmondani egy adott könyvről, hogy az ki van-e kölcsönözve, és ha igen, kinél, és kilistázni egy adott olvasónál lévő könyveket.

Első rész (könyv adatok kezelése):

Az összes főfüggvénynek nincsen visszatérési, mert mi csak műveleteket csinálunk a függvényekben és azután visszatérünk a főmenüre. És mivel az összes függvényt fájlkezelésre használjuk, mindenhol található egy error vizsgálati kódrészlet. Az első rész összes függvénye a „könyvek.txt” fájlt használ.

Az első függvényt beolvasáshoz használjuk (view). Fájl megnyitjuk olvasási módban, ha NULL, akkor hiba van, ellenkező esetben minden sort az fgets segítségével olvasunk, és kiírjuk a képernyőre a sorszámával (kivéve az első 2-t, mivel ez a fejléc).

Az új könyvek létrehozásához használjuk az add függvényt. Ehez 2 fájlt használunk, az első, az eredeti fájl amelyből beolvassuk az adatokat, a második pedig egy ideiglenes fájl, amelybe átmásoljuk az összes adatot és az új felhasználótól beolvasott információt. Ennek a függvénynek csak 1 paramétere van, ami meghatározza hogy hol írjuk az adatot. A felhasználótól megkérdezzük a könyv szerzőjét, címét és a műfajt. Egy dinamikus_tomb nevű segéd tömb használatával, dinamikusan annyi helyet foglalunk ezeknek amennyi kell ezeknek, visszatérünk az értékükkel a főfüggvénybe, majd felszabadítjuk. A kiadási évhez létrehoztam egy struktúrát amelyben tároljuk az év, hónap és nap értékeit (valós számok), ÉÉÉÉ.HH.NN formátumban adjuk, és csinálunk esetszétválasztást, ha valaki pl. évnek nagyobb

mint 2022 ad, vagy hónapnak 12-nél nagyobb számot, stb. Az összes adat beolvasása után bezárjuk mindkét fájlt, töröljük az eredetit, átnevezzük az ideigleneset az eredeti nevére és visszatérünk a menübe.

A kereséshez (search függvény) strstr-el vizsgáljuk az inputot és a fájl tartalmát és az fgets és egy buffer karaktertömb változó használatával leírjuk a képernyőre. Fontos hogy ha valaki kisbetűvel írna a címet akkor nem működne, szóval toupper-t használunk az első betűre. Ha a felhasználó 0-át ad inputnak, akkor visszatérünk a search függvényre.

Törléskor (delete_book) pedig megjelenítünk az összes sort a view() függvény használatával, adunk paraméternek egy törölni kívánt sort (pl: a felhasználó választ egy sort a choose_book() segédfüggvény-el) , és azt töröljük.

- Változók: A felhasználó által megadott sor (delete_line), bool értékű ciklus feltétele (keep_reading), jelenlegi sor (current_line), fájl név (filename), ideiglenesen használt fájl (temp), karaktertömb buffer változó

Egy do while ciklusban fgets használatával a buffer-ben tároljuk az első sort, addig amíg nem tartunk az utolsó sornál. Azután ezt a temp fájlba tároljuk, ha teljesül az a feltétel, hogy current_line != delete_line, vagyis ha a jelenlegi sor nem a törölni kívánt sor. Végül, bezárjuk a fájlokat, az eredeti fájlt töröljük és átnevezzük a másikat az eredeti nevére. (megegyezik az add() algoritmussal)

A módosítás/kicserélés függvényben (edit/replace) a felhasználó választ egy sort, és amikor elérjük azt a sort, akkor töröljük (a delete_book függvényt hívjuk és a választott sort adunk paraméternek) és hozzáadunk egy újat (add függvény).

Második rész (olvasók adatainak kezelése):

Az itt látható függvények az „olvasok.txt” fájlt használják.

Olvasók struktúra (reader): Az olvasó információinak tárolására szolgál. Ide tartozik egy 4 karakter hosszú azonosítószám (ID), egész típusú életkor, karaktertömb típusú kikölcsönzött könyvnevű változó, (amelyet a rent() függvényben használunk) és azon belül még egy struct ami a nevekhez használjuk. Ennek 2 altípusa van: keresztnév és vezetéknév.

Reader_info függvény: Az olvasók menüje.

Az első függvényt beolvasáshoz használjuk (display). Beolvassuk a fájl összes tartalmát, azután visszatérünk a főmenüre. Ezt karakterenként beolvassuk és leírjuk a képernyőre EOF-ig.

Egy új olvasó hozzáadásához viszont (create) beolvassuk a felhasználótól az adatokat, betároljuk egy reader típusú változóba, amelyet dinamikusan foglaltuk, leírjuk a fájlba és felszabadítjuk.

Rent: Ebben a függvényben 2 különböző fájlt fogunk használni (az előzőek hasonlóan: átmásolás->modosítás->átnevezés->törlés). Ebben a függvényben minden sor beolvasásra kerül egy pufferváltozóban, és kérjük a felhasználót, hogy válasszon sorszámot (melyik könyvet bérelni szeretne), valamint egy olvasóazonosítót. Itt is do-while ciklust használunk, 2 feltétellel. Először is ellenőrizzük, hogy elértük-e a fájl vége, ha igen, akkor a (boolean) keep_reading változó hamisra van állítva és kilépünk a ciklusból. A második pedig hogy elértük-e a módosítandó sort. Ha igen, akkor tároljuk a kívánt könyvet a (reader) r.book_rented változóban, és ellenőrizzük ha a könyvet korábban kölcsönözték vagy sem. Ha nem, akkor módosítjuk a sort az azonosítóval és az [R] hozzáadásával. Végül, ha egyik feltétel sem teljesül, akkor egyszerűen csak írjuk a sort a másik fájlba. A ciklusból kilépéskor az r változót megadjuk a rent2 függvénynek.

Rent2: Ez a függvény nagyon hasonlóan működik, mint a bérleti függvény, ahol 2 fájlunk van, az egyiket olvassák, a másikat pedig módosítják. A do-while ciklus ugyanaz, kivéve a 2. feltételt, ahol ellenőrizzük, hogy az id egyezik-e a pufferben olvasottakkal, ha igen, beírjuk a 2. fájlba.