Telefonkönyv

Fejlesztői dokumentáció

A következő dokumentáció a függvények felbontásának, csoportosításának, működésének megmagyarázásáról fog szólni.

A következő könyvtárakat használja a program:

(Hivatalos)

- *stdio.h*
- *stdlib.h*
- *stdbool.h*
- *string.h*

(Saját készitésű)

- *str.h*
- *kiir.h*
- *filekez.h*

Hivatalos függvények miért?

- Az stdio.h a szabványos kimenet használata miatt
- Az stdlib.h a memória kezelés miatt
- Az stdbool.h az igaz/hamis-ok miatt
- A string.h pedig a sztringkezelő függvények miatt.

Saját függvények és könyvtáraik:

str.h:

Itt deklaráljuk a láncolt listát ,valamint itt vannak azon függvények melyek a láncolt listával kapcsolatosak.

A láncolt listába a következő elemek szerepelnek:

- int idx ←Index
- char vnev[50+1] ← Vezetéknév
- char knev[50+1] ←Keresztnév

- char nem[1+1] ←Nem (0=Nő,1=Férfi)
- char varos[30+1] ←Város
- char szam[15+1] ←Telefonszám
- char mail[30+1] ← E-mail
- char foglalkozas[50+1] ← Foglalkozás
- struct Rekord *kov ← A következő elemre mutató pointer

A könyvtárban a következő függvények találhatóak:

- teljlistazas(Rekord *eleje): megkapja a láncolt lista elejére mutató pointert , és for ciklussal bejárja és kiíratja az összes adatot
- feltlistazas(Rekord *eleje): bekér a felhasználótól egy számot mely megfelel a szabványos kimeneten kijelzetteknek, azután bekér a felhasználótól egy feltételt ami alapján keressen. Ezek után megvizsgáljuk melyik feltételt választotta és végig megyünk a láncolt listán. Ha egyezés van kiírjuk a személy összes adatát ,és növeljük a db egész szám típusú változónkat 1-el.Legvégül kiírjuk a db értékét.
- rmeret(Rekord *eleje) végig meg a láncolt listán és vissza ad egy int értéket ami megmondja hogy hány eleme van a láncolt listának.
- add(Rekord *eleje)bekér a felhasználótól egy új személy adatait melyet a szabványos kimenetet jelzett módon kell megadni és ezen adatokat a láncolt lista végére fűzi.
- mod(Rekord *eleje)bekéri a felhasználótól várt indexet melyet módosítani szeretnénk ,majd a láncolt listán elmegyünk addig amíg nem találjuk meg az azonos indexet. Ezek után várunk a felhasználótól egy számot mely a szabványos kimeneten jelez ,hogy mi minek az értéke. Miután felhasználó megadta a számot bekérjük szabványos bemeneten az új elemet amit eltárolunk a megfelelő helyen.
- free(Rekord * eleje) felszabadítja a láncolt lista által lefoglalt memória területet

kiir.h:

itt egyetlen függvény található ,csak azért van külön könyvtárban ,mert az str.h-ba se és a filekez.h-ba se lehet besorolni ,és a main.c-ben sok helyet foglal el

- menuk(): szabványos kimenetre ki írja a menüpontokat

filekez.h:

E könyvtár fájlkezelő függvényekkel foglalkozik.

- letezik(FILE *f) megnézi ,hogy a függvénybe beírt az létezik-e ha igen akkor TRUE visszatérési értéket ad ,ha nem akkor FALSE-t.
- fhossz() megnyitja a megadott fájlt és végig megy az összes karakteren és eltárolja a \n-eket és a \r-eket majd visszatér ezzel az értékkel.
- feltolt(Rekord *eleje) megnyitja a megadott fájlt meghívja az fhossz()-t és eltároljuk értékét ,majd for ciklusban végig fscanf-et használva beolvassuk az összes elemet a láncolt listába. Tudjuk hogy hányszor fog lefutni a program mivel az fhossz() megmondja.
- lement(Rekord *eleje)megnyitja az adott helyen található fájl-t „write” módként és fprintf()-et használva beleírja a fájl-ba a láncolt lista összes elemét , ezután pedig sortörést rak.
- vcard_import(Rekord *eleje) Ez a függvény először kér a felhasználótól egy címet melyen az importálandó elem van. Ezek után megnézi a program hogy létezik-e a fájl. Amennyiben igen egy új láncolt lista elemet hozunk létre aminek először megadjuk az indexét az rmeret() által, utána megkeressük a láncolt listában az utolsó helyet és oda szúrjuk. A beolvasás felelős algoritmus 11-szer fut le mivel a program csak a sajátos vCard-ot tudja beolvasni ami pontosan 11-soros minden sorban a megfelelő adatok vannak(megfelelő alatt azt értem ,hogy a program mindent egyes sort külön kezel le, ezért ha nem jó sorrendbe érkeznek a memóriahiba is fennállhat).Minden egyes sor-t egy bizonyos karakterig beolvassuk majd onnan egy másik bizonyos karakterig olvasunk ,így megtudjuk a kezdőindexét a beolvasandó szövegnek, és azt is tudjuk ,hogy meddig tart. Az intervallumban minden egyes karaktert átmásolunk egy karakterláncba ,és majd azt a strcpy függvénnyel a láncolt lista végére fűzött elemhez.
- vcard_export(Rekord *eleje)Felhasználó által megadott indexet megkeresi a láncolt listában és azt importálja a programban leírt sajátos .vcf formátumba.

main.c:

Itt csak a main() függvény található ,melyben a láncolt lista inicializálása, a menuk() meghívása és más egyéb függvények meghívása történik. Ezeken kívül itt van itt egy switch case mely érzékeny a felhasználó által adott bemeneti értékre.Minden egyes eset csak meghívja azt a függvényt ami felelős a menuk()-ban írottakért.

Ha a programban szeretnénk módosítani a rekord.txt helyét fontos ,hogy a következő helyeken kell azt átírni:

- main() → read
- fhossz() → read
- feltolt() → read
- lement() → write