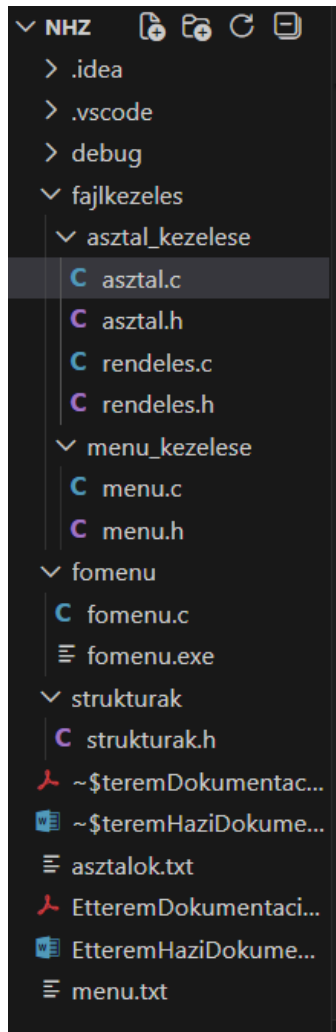


FORRÁSFÁJL DOKUMENTÁCIÓ

Programnevek: fomenu.c, asztal.c, rendelesek.c menu.c

NHZ mappa felépítése:



Struktúrák:

MenuElem

Elraktározza egy étel nevét (maximum 100 karakter hosszú) és árát, illetve a kov pointert ami a láncolt lista következő elemére mutat.

AsztalElem

Elraktározza egy asztal azonosítóját, férőhelyét, elhelyezkedésének sorát illetve oszlopát, hogy foglalt-e és a rendeléseit (maximum 500 karakter hosszú), ami „null” ha nincsen rendelése. A végén a kov pointer itt is a láncolt lista következő elemére mutat.

RendelesElem

Elraktározza az adott asztalhoz tartozó rendelt étel(ek) nevét (maximum 100 karakter hosszú) és, hogy hány darabot kért belőle, illetve a kov pointer ami az asztal következő rendelésére mutat

fomenu leírása:

Megjelenik a főmenü ahol a felhasználó 5 függvény közül tud választani amit egy switch kezel a main függvényben:

- **1. ujuvendeg:** A program megkérdezi a felhasználótól, hogy hányan jöttek (a bemenet egész szám), és az alapján végigfut a létező asztalokon, és ha van olyan aminek a férőhelye megegyezik ezzel a számmal és nem foglalt, akkor oda leülteti őket (az adott asztal foglalt változója 1 lesz), amúgy meg ha nincs akkor ezt elmondja a felhasználónak és addig csinálja még nincs üres hely a megadott számhoz vagy a felhasználó nem ad 0-át válaszként. Ehhez a függvény a szabad_hely_keres() függvényt használja.
- **2. meglevovendeg:** A program megkérdezi a felhasználót, hogy melyik asztallal akar foglalkozni (a bemenet egész szám), ezt megnézi, hogy létezik-e az asztalok láncolt listájában és, hogy foglalt-e, és ha igen akkor megjelenik az almenü:
 - **5. rendeles:** A felhasználó megadhatja, hogy milyen étel (eket) akar fogyasztani és abból hány darabot. Ez elraktározódik a hozzájuk tartozó struktúra rendeléseibe. „END 0” bemenettel tud kilépni ebből.
 - **6. szamla:** Kiírja az asztal összes rendelését és, hogy összesen mennyivel tartoznak, majd a program kitörli a rendeléseit és üres lesz újra a program.
 - **0. vissza:** Visszalép a főmenübe.
- **3. foglaltsagiterkep:** Kirajzolja a függvény az étterem jelenlegi foglaltsági térképét. Az asztalok sor és oszlop paraméterei alapján helyezi el őket a térképen.
- **4. menukezelese:** Lehetősége van a felhasználónak ugyanabban a formátumban újabb ételt hozzáadni a menühöz vagy megváltoztatni valamelyik árát vagy törölni ahol csak az étel nevét kell megadni. Ez a menukezelese() függvényben történik, ahol a láncolt lista memóriacímét adjuk paraméterként, hogy tényleg tudja változtatni a függvény (kétszeres indirekció).

- 5. asztalkezelese: Lehetősége van a felhasználónak ugyanabban a formátumban újabb asztalt hozzáadni az asztalok láncolt listájához, vagy törölni belőle, ahol csak az asztal azonosítóját kell megadni. Ez az asztalkezelese() függvényben történik, ahol a láncolt lista memóriacímét adjuk paraméterként, hogy tényleg tudja változtatni a függvény (kétszeres indirekció).

Ha a felhasználó a 9-et adja meg válaszként, akkor a program kilép, elmenti láncolt listákat a menu.txt és az asztalok.txt fájllokba az asztalok_elment() és a menu_elment() függvények segítségével majd felszabadít minden dinamikusan foglalt területet az asztal_felszabadit() és menu_felszabadit() függvényekkel és véget ér a program.

asztal leírása:

A következő függvények az asztalok láncolt lista raktározásáért, az asztalok kereséséért és a helyfoglaltsági térkép paramétereinek megadásáért felelnek:

asztal letezik

Egy változóba elraktározza az asztalok.txt fájlt majd egy feltétellel megnézi, hogy létezik-e vagy sem, ennek alapján ad igazt vagy hamist visszatérési értéként.

asztal beolvas+rendelesek beolvas és asztal létrehoz

Ha az asztal_letezik() hamist adott, tehát létrehozni kell, akkor az asztal_eldonto() függvényben elkezd kéri a felhasználótól az asztal azonosítóját, férőhelyét, sorát, oszlopát és, hogy foglalt-e, ezeket paraméterként megadja a asztal_latrehoz függvénynek ami létrehozza velük a láncolt listát. Emellett az asztalkezelese() függvényben is szerepel, hogy a főmenüből is tudjunk új asztalt hozzáadni a láncolt listához. Mindkét esetben úgy hozza létre, hogy az új elemet beszúrja a láncolt lista elejére.

Ha az asztal_letezik() igazt adott, tehát beolvasni kell, akkor az asztal_beolvas megnyitja az asztalok.txt fájlt, és ideiglenesen elmenti soronként az asztal azonosítóját, férőhelyét, sorát, oszlopát, foglaltságát és rendeléseit 6 bufferváltozóba, amiket meg berak a láncolt listába. Itt az étel nevét dinamikus memóriakezeléssel tárolja el. a rendelések értelmezéséhez a rendelesek_beolvas függvényt használja, ami tokenizálja a rendeléseket és egy 3. láncolt listába tárolja el. Ez lesz az asztal->rendelesek tartalma.

asztal kitorol

Az asztalkezelese() függvényben szerepel. A felhasználó megadja a törölni kívánt asztal azonosítóját. A függvény végigiterál a láncolt listán, amíg meg nem találja a keresett azonosítójú asztalt. Ha megtalálta, először meghívja a rendeles_felszabadit() függvényt az

asztalhoz tartozó rendelésekre, hogy elkerülje a memóriaszivárgást. Végül felszabadítja az struktúrát és összefűzi a listát.

asztal_felszabadit és rendeles_felszabadit

Miután elvégeztük a fájlbeolvasást és megtettünk az adatokkal mindent amit akartunk, akkor felszabadítjuk a dinamikusán tárolt adatokat illetve struktúrákat. Mivel a struktúrán belül a rendelések önmagába egy láncolt lista, ezért az asztalok felszabadításánál a `rendeles_felszabadit()` függvény külön felszabadítja az adott asztalhoz tartozó összes rendelet.

szabad_hely_keres

Az `ujvendeg()` függvény hívja meg. A függvény megkapja a vendégek számát (egész szám) és az asztalok láncolt listáját. Végigiterál a listán, amíg nem talál egy olyan asztalt, amelynek a férőhelye megegyezik a vendégek számával és nem foglalt (`foglalt == 0`). Ha talál ilyet, annak a foglalt változóját 1-re állítja, és visszatér az asztal címével. Ha nem talál, NULL értékkel tér vissza.

asztal_kereso

A `meglevovendeg()` függvény használja. A függvény megkapja a keresett asztal azonosítóját. Végigiterál a láncolt listán, amíg nem talál egyező azonosítójú és foglalt (`foglalt == 1`) asztalt. Visszatér a talált asztal címével, vagy NULL értékkel.

asztal_eldonto

Ez a fájlkezelés vezérlő függvénye. Itt vannak összekapcsolva az asztalok fájlkezeléséhez tartozó függvények, a `asztal_letezik()`, a `asztal_beolvas()` illetve a `asztal_letrehoz()`. Ennek visszatérési értéke a helyesen elraktározott láncolt lista.

asztalok_elment

A főmenüből való kilépéskor hívódik meg a `main` függvényben. Megnyitja az `asztalok.txt` fájlt írásra, majd végigiterál a listán, és minden asztal adatát (ID; férőhely; sor; oszlop; foglalt; rendelések) egy sorba írja. Ha egy asztalhoz vannak rendelések, akkor sztringé alakítja őket először az `sprintf` függvénnyel, ha nincsenek rendelések, akkor a rendelések mezőbe a „null” sztring kerül.

maxsor és maxoszlop

A foglaltsagiterkep() függvény használja a térkép méreteinek meghatározásához. A függvények végigiterálnak az asztalok listáján, és megkeresik a legnagyobb sor és oszlop értéket, ezzel megadva a térkép rácsának maximális dimenzióját.

menu leírása:

menu letezik

Egy változóba elraktározza a menu.txt fájlt majd egy feltétellel megnézi, hogy létezik-e vagy sem, ennek alapján ad igazt vagy hamist visszatérési értéként.

menu létrehoz és menu beolvas

Ha a menu_letezik() hamist adott, tehát létrehozni kell, akkor a menu_eldonto() függvényben elkezdni kérni a felhasználótól az étel nevét és árát, ezeket paraméterként megadja a menu_latrehoz függvénynek ami létrehozza velük a láncolt listát. Emellett a menukezelese() függvényben is szerepel, hogy a főmenüből is tudjunk új ételt hozzáadni a menühoz. Mindkét esetben úgy hozza létre, hogy az új elemet beszúrja a láncolt lista elejére.

Ha a menu_letezik() igazt adott, tehát beolvasni kell, akkor a menu_beolvas megnyitja a menu.txt fájlt, és ideiglenesen elmenti soronként az étel nevét és árát két bufferváltozóba, amiket meg berak a láncolt listába. Itt az étel nevét dinamikus memóriakezeléssel tárolja el.

menu kitorol

A menukezelese() függvényben szerepel, a felhasználó megadja a törölni kívánt étel nevét, majd a függvény végig iterál a láncolt listán amíg meg nem találja a keresett ételt, és azt felszabadítja.

menu valtoztat

A menukezelese() függvényben szerepel, a felhasználó megadja a változtatni kívánt étel nevét és árát, majd a függvény végig iterál a láncolt listán amíg meg nem találja a keresett ételt, és annak árát megváltoztatja.

menu felszabadit

Miután elvégeztük a fájlbeolvasást és megtettünk az adatokkal mindent amit akartunk, akkor felszabadítjuk a dinamikusan tárolt adatokat illetve struktúrákat. Először a menü struktúrában tárolt étel nevét szabadítjuk fel majd magát a struktúrát.

kiiromenu

Miután elraktároztuk a menüt egy láncolt listába, megadjuk a függvénynek a láncolt listát paraméterként és az végi iterál rajta és minden menü elemet kiír a konzolra.

menu_eldonto

Ez a fájlkezelés vezérlő függvénye. Itt vannak összekapcsolva a menü fájlkezeléséhez tartozó függvények, a menu_letezik(), a menu_beolvas() illetve a menu_letrehoz(). Ennek visszatérési értéke a helyesen elraktározott láncolt lista.

etel_kereso

A rendeles() függvényben szerepel, mikor a felhasználó megadja, hogy melyik ételből akar rendelni, akkor a függvény ezt megkapja paraméterként, illetve a menü láncolt listáját is, amin végigiterál, és ha talál olyan ételt aminek neve megegyezik a megadott étellel akkor igazzal tér vissza, különben hamissal.

etel_ar

A szamla() függvényben szerepel, az etel_kereso() függvényhez hasonlóan működik, annyi különbséggel, hogy ha talál olyan ételt, akkor nem igazzal hanem magával a láncolt lista elemmel tér vissza, amit a szamla() függvény az összeg kiszámolásához használ fel, különben meg NULL-al tér vissza.

menu_elment

A főmenüből való kilépéskor hívódik meg a main függvényben. Megnyitja az menu.txt fájlt írásra, majd végigiterál a listán, és minden étel adatát (név;ár) egy sorba írja.

rendeles leírása:

Az alábbi függvények a rendelési láncolt lista létrehozásáért és kezeléséért felelnek, amely az AsztalElem struktúrában található.

rendelesek_beolvas

Az asztal_beolvas() függvény hívja meg az asztalok.txt fájl beolvasásakor a rendelések értelmezéséhez. Paraméterként megkapja az asztal rendeléseit tartalmazó sztringet (pl. "Hamburger:1,Kebab:1"). Tokenizálja a sztringet vesszők (,) mentén, és minden tokenből (étel:darab) a sscanf segítségével kiolvassa az étel nevét és a darabszámot. Ezután meghívja a rendeles_hozzaad() függvényt, hogy az új rendeléseket egy láncolt listába tárolja. Végén felszabadítja az ideiglenes másolatot.

rendeles_hozzaad

Az `rendelesek_beolvas()` és a `rendeles()` függvények hívják meg. Létrehozza az új rendelési elemet (`RendelesElem`), amely elraktározza az étel nevét és a rendelt darabszámot. Az új elemet a láncolt lista elejére szúrja be.

kiirorendelesek

A `kiiroasztalok()` és a `rendeles()` függvények hívják meg. Végigiterál az adott asztal rendelési láncolt listáján, és kiírja az egyes tételeket a darabszámukkal együtt.

rendeles_kereso

A `rendeles()` függvény használja, hogy megkeresse, létezik-e már az adott étel az asztal rendelései között (hogy csak a darabszámot kelljen növelni). Visszatér a talált rendelés címével, vagy NULL értékkel.

rendeles_felszabadit

Az `asztal_felszabadit()`, a `szamla()` és az `asztal_kitorol()` függvények hívják meg. Felszabadítja az adott asztalhoz tartozó minden rendelés láncolt listájának elemeit (`RendelesElem`). Fontos, hogy először felszabadítja a **dinamikusan tárolt ételnevet** (`rendelesnev`), majd magát a rendelési struktúrát.