Tantárgy	A programozás alapjai 1.
Feladat	Étterem
Dátum	2021.11.27.

SZÜKSÉGES KÖRNYEZET

1. MinGW GCC compiler az etterem.exe fájl legenerálásához
2. Szabványos függvénykönyvtárak
3. debugmalloc a memóriakezelés ellenőrzéséhez (deps mappában)
4. econio a konzolos megjelenítéshez (deps mappában)

A fordítás lépései (2 lehetőség)

1. make paranccsal, mert van Makefile a mappában.
2. gcc etterem.c asztalok.c asztalok.h funkciok.c funkciok.h megjelenites.c megjelenites.h menu.c menu.h rendelesek.c rendelesek.h deps/debugmalloc.h deps/econio.c deps/econio.h -o etterem parancs futtatásával.

MODULOK

1. **Asztalok:** Leírja az asztalok tárolására alkalmas struktúrákat és definiálja az asztalok memória- és fájlkezelő függvényeit. (a rendelések fájlkezelő függvényei is ide tartoznak)
2. **Menü:** Leírja a menü tárolására alkalmas struktúrákat és definiálja a menü memória- és fájlkezelő függvényeit.
3. **Rendelések:** Leírja a rendelések tárolására alkalmas struktúrákat és definiálja a rendelések memóriakezelő függvényeit.
4. **Funkciók:** Definiálja az adatstruktúrák kezelésére alkalmas függvényeket. (nagyrészt a specifikációban kifejtett funkciók megvalósításai)
5. **Megjelenítés:** A konzolos menü megjelenítéséért felelős modul, egy állapotgépet valósít meg.

ADATSTRUKTÚRÁK

A fájl adatainak eltárolására a láncolt lista a legalkalmasabb, mivel nem lehet előre tudni, hogy hány eleme van a fájlnak. A lent kifejtett konstrukcióval konstans idő alatt hozzá lehet fűzni új elemet a lista végéhez. Ez nagy előny a tömbbel szemben, ahol új memóriafoglalást és átmásolást kell végrehajtani új elem hozzáfűzésénél.

A menü adatainak eltárolása

struct Menupont	Egy menüpont adatait eltároló struktúra

int azonosito	A menüpont azonosítója (0-tól számozva, a fájl sorai szerint)
int ar	A menüpont ára Ft-ban
char *nev	A menüpont neve (dinamikusan van foglalva)
struct Menupont *kov	A következő menüpontra mutató pointer, láncolt listához

struct Menu	A láncolt lista lényeges adatait tároló struktúra
struct Menupont *eleje	Az első menüpontra mutató pointer (bejáráshoz)
struct Menupont *vege	Az utolsó menüpontra mutató pointer (hozzáadáshoz - O(1))

A rendelések adatainak eltárolása

struct Rendeles	Egy megrendelés adatait eltároló struktúra
Menupont *termek	A rendelt termékre mutató pointer
int darab	A megrendeléshez tartozó darabszám
struct Rendeles *kov	A következő rendelésre mutató pointer

struct Rendelesek	Egy rendeléssorozatot eltároló láncolt lista lényeges elemeit tartalmazó struktúra (pl: egyes helyekhez tartozó rendeléseknél)
struct Rendeles *eleje	Az első rendelésre mutató pointer (bejáráshoz)
struct Rendeles *vege	Az utolsó rendelésre mutató pointer (hozzáadáshoz – O(1))

Az asztalok adatainak eltárolása

struct Pozicio	Egy asztal pozícióját eltároló struktúra
int X	Az asztal pozíciója az X tengelyen
int Y	Az asztal pozíciója az Y tengelyen

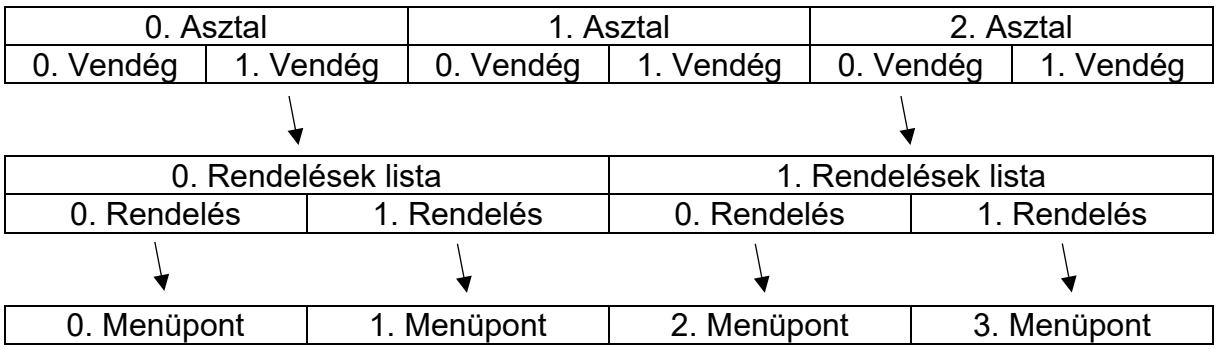
enum Statusz	Egy asztal státuszát kifejező enum
SZABAD (0)	Azt jelenti, hogy az asztal nincs megnyitva, nem ül ott senki
FOGLALT (1)	Azt jelenti, hogy az asztal éppen meg van nyitva, ülnek ott vendégek.

struct Asztal	Egy asztal adatait eltároló struktúra
int azonosito	Az asztal azonosítója (0-tól számozva a fájl sorai szerint)
struct Pozicio pozicio	Az asztal elhelyezkedése az éttermen belül

enum Statusz statusz	Az asztal státusza (szabad / foglalt)
int ferohely	Az asztal helyeinek száma (hány szék van)
Rendelesek *hely_rendelesek	Az asztal helyeihez tartozó rendelések (minden helyhez egy láncolt lista)
struct Asztal *kov	A következő asztalra mutató pointer, láncolt listához

struct Asztalok	A láncolt lista lényeges elemeit eltároló struktúra
struct Asztal *eleje	Az első asztalra mutató pointer (bejáráshoz)
struct Asztal *vege	Az utolsó asztalra mutató pointer (hozzáadáshoz – O(1))

Az adatstruktúrák kapcsolata (példa)



FONTOSABB FÜGGVÉNYEK

Az asztalok modulban

- **Asztal *asztal_foglal(int azonosito, struct Pozicio pozicio, enum Statusz statusz, int ferohely)**

  Létrehoz egy dinamikusan foglalt asztalt a megadott adatokkal. NULL-t ad vissza, ha nem sikerült a foglalás.

- **void asztalok_felszabadit(const Asztalok *asztalok)**

  Felszabadítja az asztalokat tartalmazó láncolt lista elemeit.

- **int asztalok_beolvas(char *fajl, Asztalok *asztalok)**

  Beolvassa a fájlból az asztalok adatait és létrehozza a láncolt listát, majd paraméterlistán visszaadja. Visszatérési értéke 1, ha nem sikerült megnyitni a fájlt. 2, ha nem sikerült a hozzáadás a listához. 0, ha sikerült a beolvasás.

- **int asztalok_kiir(char *fajl, const Asztalok *asztalok)**

  Kiírja a fájlba az asztalok adatait. Visszatérési értéke 1, ha nem sikerült megnyitni a fájlt. 0, ha sikeres volt a kiírás.

- **int rendelesek_beolvas(char *fajl, const Menu *menu, Asztalok *asztalok)**

  Beolvassa a fájlból a rendelések adatait, hozzáfűzi őket a megfelelő asztal megfelelő helyének rendeléseihez. Visszatérési értéke 1, ha nem sikerült megnyitni a fájlt. 2, ha nem sikerült a hozzáadás a listához. 0, ha sikerült a beolvasás.

- **int rendelesek_kiir(char *fajl, const Asztalok *asztalok)**

  Kiírja a fájlba a rendelések adatait. Visszatérési értéke 1, ha nem sikerült megnyitni a fájlt. 0, ha sikeres volt a kiírás.

A menü modulban

- **Menupont *menupont_foglal(int azonosito, int ar, char *nev)**

  Létrehoz egy dinamikusan foglalt menüpontot a megadott adatokkal. NULL-t ad vissza, ha nem sikerült a foglalás.

- **void menu_felszabadit(const Menu *menu)**

  Felszabadítja a menüt tartalmazó láncolt lista elemeit.

- **int menu_beolvas(char *fajl, Menu *menu)**

  Beolvassa a fájlból a menü adatait és létrehozzá a láncolt listát, melyet paraméterlistán vissza is ad. Visszatérési értéke 1, ha nem sikerült megnyitni a fájlt. 2, ha nem sikerült a hozzáadás a listához. 0, ha sikerült a beolvasás.

- **int menu_kiir(char *fajl, const Menu *menu)**

  Kiírja a fájlba a menü adatait. Visszatérési értéke 1, ha nem sikerült megnyitni a fájlt. 0, ha sikeres volt a kiírás.

A rendelések modulban

- **Rendeles *rendeles_foglal(Menupont *termek, int darab)**

  Létrehoz egy dinamikusan foglalt rendelést a megadott adatokkal. NULL-t ad vissza, ha nem sikerült a foglalás.

- **void rendelesek_felszabadit(const Rendelesek *rendelesek)**

  Felszabadítja a rendeléseket tartalmazó láncolt lista elemeit.

A funkciók modulban

- **int asztal_hozzaad(struct Pozicio pozicio, enum Statusz statusz, int ferohely, Asztalok *asztalok)**

  Létrehoz egy új asztalt a megadott adatokkal, majd hozzáfűzi a megadott lista végéhez. Visszatérési értéke 1, ha nem sikerült hozzáadni az asztalt. 0, ha sikerült.

- **Asztal *asztal_keres(int azonosito, const Asztalok *asztalok)**

Azonosító alapján megkeres egy asztalt a megadott listában. Ha nem talált semmit, akkor NULL-t ad vissza.

- **int asztal_megnyit(int azonosito, Asztalok *asztalok)**

  A megadott azonosítójú asztal státuszát foglaltra állítja. Visszatérési értéke 1, ha a megadott azonosítóval nincsen asztal. 0, ha sikerült a megnyitás.

- **int asztal_lezar(int azonosito, Asztalok *asztalok)**

  A megadott azonosítójú asztal státuszát szabadra állítja. Visszatérési értéke 1, ha a megadott azonosítóval nincsen asztal. 0, ha sikerült a lezárás.

- **int asztal_torol(int azonosito, Asztalok *asztalok)**

  A megadott azonosítójú asztalt törli a láncolt listából. Visszatérési értéke 1, ha nem található az asztal, vagy foglalt a státusza. 0, ha sikerült a törlés.

- **int menupont_hozzaad(int ar, char *nev, Menu *menu)**

  Létrehoz egy új menüpontot a megadott adatokkal, majd hozzáfűzi a megadott lista végéhez. Visszatérési értéke 1, ha nem sikerült hozzáadni a menüpontot, 0 ha sikerült.

- **Menupont *menupont_keres(int azonosito, const Menu *menu)**

  Azonosító alapján megkeres egy menüpontot a megadott listában. Ha nem talált semmit, akkor NULL-t ad vissza.

- **int menupont_torol(int azonosito, Menu *menu, const Asztalok *asztalok)**

  A megadott azonosítójú menüpontot törli a láncolt listából. Visszatérési értéke 1, ha nem található a megadott azonosítóval menüpont, vagy ha van hozzátartozó rendelés. 0, ha sikerült a törlés.

- **int rendeles_hozzaad(int termek_azonosito, int darab, const Menu *menu, Rendelesek *rendelesek)**

  Létrehoz egy új rendelést a megadott adatokkal, majd hozzáfűzi a megadott lista végéhez. Visszatérési értéke 1, ha nem sikerült felvenni a rendelést. 0, ha sikerült.

KONZOLOS MENÜ

A konzolos menü egy állapotgépet valósít meg, ennek állapotai a MenuAllapot enumban találhatóak meg. Minden állapothoz tartozik egy vezérlőfüggvény, amely a következő állapot értékét adja vissza. A program ez az állapotkód alapján tudja, hogy éppen mit kell megjelenítenie, vagy milyen adatokat kell beolvasnia.

TARTALOMJEGYZÉK