

Dokumentace projektu

Kompatibilita:

Jako první položku dokumentace uvádím kompatibilitu, protože pracuji na Linux Kubuntu 20.04, který používá jiný kompilátor než Windows. V mém projektu se projevuje jen jediný rozdíl, že pro mazání konzolového okna používám příkaz `system("clear")` namísto `system("cls")`.

Protože jsem nějak nezjistil, jak upravit šířku konzolového okna v Linuxech, je při spuštění programu vyžadováno vstupní heslo. Všiml jsem si totiž, že po použití funkce `gets()` se konzolové okno přizpůsobí šířce vypisovaných stringů. Pokud byste to chtěl vidět nepřizpůsobeno, je v kódu vyznačeno, kterou část zapoznámkovat, aby se první vypsání databáze provedlo „nepěkně“.

Heslem může být jakýkoliv řetězec, například „ahoj“.

Popis chování programu

Nejdříve je uživatel vyzván k zadání hesla z již zmíněných důvodů. Po zadání hesla se nagenereuje databáze z dat v náhodném CSV souboru ve složce programu. Do konzolového okna se vypíše databáze a pod ni menu. Program v menu nabídne uživateli operace, co může uživatel s databází dělat. Jednotlivé operace s databází obsluhuje stavový automat realizovaný přes `switch()` v nekonečné `while` smyčce. Operacím jsou přiděleny číselné hodnoty, které program přijímá od uživatele. To platí i pro vstupy od uživatele při provádění operací. Jak uživatel vybere možnost a potvrdí, stavový automat přeskočí do pole, kde se daná operace provádí, provede ji (s případným dalším vstupem od uživatele) a vrátí se do menu. Takto se může v programu pracovat donekonečna (jen velikost paměti počítače omezuje velikost databáze). Pokud se v menu zvolí hodnota 0, dynamicky alokovaná paměť se vyčistí a program se ukončí.

Pro použité funkce a struktury je vytvořena knihovna „`logic_for_database`“. V main souboru je realizován stavový automat.

Popis vytvořených struktur:

```
typedef struct homework t_homework;
```

Definuje jednu položku databáze. Specifikována je v hlavičkovém souboru.

```
typedef struct database t_database;
```

Zde je definována struktura „`database`“ obsahující ukazatele na první a poslední položku databáze. Její užití zjednodušuje práci s databází jako argumentem a návratovou hodnotou funkcí programu.

Popis vytvořených funkcí:

`t_database read_from_csv(char* str);`

Načte data z CSV souboru a vytvoří z nich dynamicky alokovaný jednosměrný lineární seznam. Přijímá řetězec, ve kterém je název souboru pro načtení.

Zásadní pro čtení dat ze souboru jsou následující knihovní funkce:

`fgets()` – načte se jeden řádek souboru do řetězce buffer

`strtok(řetězec, oddělovací znak)` – vrací řetězec, do kterého načte hodnotu z řetězce buffer než potká oddělovací znak; pro pokračování dělení řetězce funkcí `strtok()` musí být po každém cyklu funkci `strtok()` dána jako první argument hodnota `NULL`.

`t_database init();`

Vytvoří databázi z náhodného souboru ve složce.

`void show_database(t_database homeworks);`

Graficky zpracuje a vypíše databázi do konzole.

`t_database add_item(t_database homeworks);`

Přidá na konec databáze položku, jejíž atributy (hodnoty sloupců) jsou definovány uživatelem.

`t_database append_csv(t_database homeworks);`

Přidá nakonec databáze data z CSV souboru vybraného uživatelem. Uživatel je omezen jen na CSV soubory vytvořené tvůrcem programu ve složce programu.

`t_database delete_item(t_database homeworks);`

Vymaže položku databáze zvolenou uživatelem podle ID položky. V haldě funkcí `free()` uvolní datový prostor smazané položky.

`t_database sort_homeworks(t_database homeworks);`

Řadicím algoritmem typu *bubblesort* setřídí aktuální databázi podle uživatelem vybraného sloupce. Uživatel též vybírá, jestli se má třídit vzestupně či sestupně. Pro abecední řazení je použita funkce `strcmp()`, která porovná dva řetězce a podle hodnot prvků ASCII tabulky určí jejich abecední pořadí.

`void save_to_csv(t_database homeworks);`

Uloží aktuální databázi do složky programu pod zadaným názvem od uživatele. Požadovaný název souboru je třeba zadat s koncovkou „.csv“.

`void clear_and_end(t_database homeworks);`

Uvolní dynamicky alokovanou paměť databáze před koncem programu.