Cvicny zapoctovy priklad

Vašim úkolem bude naprogramovat velmi jednoduchou simulaci fronty transakcí v bance. Soucasti kostry jsou tyto soubory:

- zadani.c

  Jediny soubor, ktery budete menit. Sem prijde vase reseni. Zaroven tu mate predepsany struktury, ktere budete vyuzivat (kdykoliv si vsak muzete zavest dalsi).

- functions.h

  Zde jsou nadefinovany funkce, ktere se vam budou hodit, z komentaru snad jejich vyznam pochopite.

- functions.c

  Deklarace funkci z functions.c.

- Makefile

  Pripravil jsem pro vas i Makefile, diky tomu muzete kod kompilovat zavolanim prikazu **make**.

- users.txt

  Binarni soubor obsahujici zakazniky banky (ve formatu **struct user**).

- plainttext.txt

  Obsahuje jmena uzivatelu, jejich hesla a stav uctu - slouzi pro testovani a kontrolu, zda vam program spravne funguje.

Program se bude volat s prave jednim argumentem, jehoz vyznam je jmeno souboru, ktery obsahuje informace o zakaznicich banky. V nasem pripade je tento soubor **users.txt**. Nebude-li tento argument zadan, nebo bude-li jich zadano vic, vypiste vhodnou chybovou hlasku a program ukoncete. Podobne, pokud se soubor nepodari otevrit. Pro jednoduchost vsak muzete soubor, v pripade, ze se jej podarilo otevrit, povazovat za korektni - pri cteni z nej nemusite osetrovat chyby.

Prvnim ukolem tedy bude nacist uzivatele (zakazniky banky) a vhodne si je ulozit, aby s nimi bylo dale mozne pracovat.

Nasledne se program bude uzivatele neustale ptat na dalsi prikaz (podobne jako prikazova radka), ukonci se prikazem **quit**. V tento moment nezapomente uvolnit veskerou naalokovanou pamet!

Prehled trivialnich prikazu, ktere program musi chapat:

- help

  Vypise uzivateli napovedu. Prislusnou funkci jiz mate v kostre naprogramovanou.

- print users

  Vypise vsechny uzivatele a stav jejich uctu.

- print transactions

  Posledni trivialni prikaz. Vypise vsechny transakce v tomto formatu:

  jmeno odesilatele -> jmeno prijemce castka

Nyni se dostavame k jadru ukolu: co je to tedy ta transakce a kde se berou?

Na zacatku programu nejsou transakce zadne. Muzeme je vsak postupne pridavat pomoci prikazu **add transaction**. Po zadani prikazu budu pozadan o autentizaci: program po mne bude chtit uzivatelske jmeno a heslo. Pokud toto probehne uspesne, budu dotazan komu chci penize poslat (zadam uzivatelske jmeno prijemce) a nakonec pochopitelne zadam i prislusnou castku.

Tento prikaz musi osetrovat nekolik veci a ve vsech nasledujicich situacich v pripade selhani nalezite uvolnit pamet, zahodit posledni transakci, ale zachovat stare a byt pripraven prijimat nove. Take by mel informovat uzivatele vhodnou hlaskou ohledne toho, co selhalo.

- autentizace

  Uzivatel zada nespravne uzivatelske jmeno/heslo.

- jmeno prijemce

  Uzivatelske jmeno prijemce neni v databazi.

- castka

  Snazim se poslat vic nez mam na uctu. Zde se neni mozne ridit jen podle polozky **balance** v **struct user**. Musim se divat i na vsechny dalsi transakce z daneho uctu (prichozi transakce ignorujeme). Mam-li tedy na uctu 50 a poslu 20 uzivateli A, program mi nasledne nemuze dovolit poslat vice nez 30. Prikaz **print users** vsak v prislusnem sloupci porad vypisuje 50!

Protoze je provest transakci drahy proces, chceme minimalizovat jejich pocet. Z toho duvodu musite i slucovat transakce mezi dvema stejnymi ucty (nebudou existovat dve transakce se stejnym odesilatelem i prijemcem). Napriklad posle-li uzivatel A uzivateli B 20, a pote nekdy 40, slouci se tyto dve transakce do jedne (o castce 60). Neslucuji se vsak transakce opacne: A posila penize B a B posila penize A jsou dve ruzne transakce.

Par dalsich poznamek (a rad):

- Pomoci #**include** muzete pripojit libovolne dalsi vhodne knihovny, podobne muzete definovat vlastni funkce, struktury... – cokoliv budete potrebovat.

- Autentizace uzivatele heslem probiha tak, ze se vezme heslo, ktere zadal, jeho sul, a obe tyto veci se predaji funkci **hash**. Vysledny hash se pak porovna s prislusnym atributem **passhash**.

- Hlaseni chyb je vhodne smerovat na standardni chybovy vystup.
- Vsimnete si, ze po celou dobu vubec nemenite obsah struktury **struct user** (s vyjimkou nacteni pochopitelne). To plati i pro jeji atribut **balance**.