Opis

Asembler je realizovan kao funkcija `string assemble(string path_to_source)`, koja dobija kao parameter putanju do fajla sa izvornim kodom, a vraća putanju do fajla sa objektim kodom. Klase koje su deo implementacije date su u nastavku.

- `Line` – predstavlja jednu liniju izvornog koda na asembleru. Pomoću ove klase vršene su provere ispravnosti linije, izvlačenje podataka iz nje (labela, instrukcija, direktiva, argumenata), kao I enkodovanje instrukcija
- `Log` – klasa koja je koriščenja za vođenje logova tokom izvršavanja. Sadrži 2 funkcije – `std(string log)` i `error(string log)`, koje se koriste za standardno logivanje I za logovanje grešaka.
- `ContentTable` – predstavlja sadžaj sekcije. Ima metode za dodavanje vektora bajtova, kao I za ispis na standardni zadati `ostream`, I pretvaranje sadžaja u `string`.
- `RelocationTable` – klasa koja predstavlja tabelu za relokacije. Sadrži svoju strukturu `Entry`, koja predstavlja jedan ulaz u tabelu za relokacije. Sadži metode za dodavanje novog zapisa o relokaciji, kao I dohvatanje vrednosti odgovarajućih zapisa. Klasa sadrži metode za pretvaranje sadžaja u čitljiv oblik (`string`), kao I za ispis na odgovarajući tok podataka.
- `SymbolTable` – klasa koja predstavlja tabelu simbola. Sadrži svoju strukturu `Entry`, koja predstavlja jedan ulaz u tabelu simbola. Sadži metode za dodavanje novog ulaza, kao I dohvatanje vrednosti odgovarajućih ulaza. Klasa sadrži metode za pretvaranje sadžaja u čitljiv oblik (`string`), kao I za ispis na odgovarajući tok podataka.

Prethodno pomenute klase su smeštene u fajlove koji sadrže odgovarajući naziv - <class>.cpp za implementaciju klase, I <class>.h za deklaraciju, gde <class> predstavlja puno ili skraćeno ime klase.

Ostali fajlovi koji se koriste su

- asm.h I asm.cpp – deklaracije I implementacija funkcije za asembliranje
- my_util.h - deklaracije eksterne promeljive tipa klase `Log` koja se koristi za logovanje
- main.cpp – sadrži ulaznu tačku programa, odnosno fukncju main, koja prima 1 argument sa komandne linije, a koji predstavlja putnju do fajla sa izvornim kodom i poziva funkciju za asembliranje.

Za prevodjenje je dovoljno pokrenuti make file (`make1`) koji je dostavljen sa rešenjem, ili prevesti ga ručno pomoću komande (uz prethodno instaliranje verzije 4.9 g++ kompajlera).

```
g++-4.9 -std=c++0x -o exe1 main1.cpp asm.cpp content_table.cpp
line.cpp  log.cpp  my_util.cpp reloc_table.cpp  sym_table.cpp -I.
```

Rešenje je testirano testovima koji su dostavljeni uz rešenje, prostim pokretanjem `exe1` fajla sa argumentom naziva testa. Rezultati asembliranja su objekti fajlovi koji predstavljaju ulazne fajlove i testove za drugi zadatak, pa se nalaze u folderu sa testovima drugog zadatka. Nakon asembilaranja na standardnom izlazu ispisana je poruka o uspešnosti asembliranja, odnosno o greškama na koje se naišlo ukoliko asembliranje nije uspelo.

Naredbe za instaliranje verzije 4.9 g++ kompajlera date su u nastavku

```
sudo add-apt-repository ppa:ubuntu-toolchain-r/test
sudo apt-get update
sudo apt-get install g++-4.9
```