**Dvoprolazni assembler**

**-Sistemski softver-**

**-Domaći zadatak-**

**Generalni opis rešenja:**

* Klasa *Parser* je pomoćna klasa koja ima samo statičke metode koji obrađuju stringove.
* Objekat klase *Section* čuva bitne podatke o sekciji kao što su naziv, veličina, sadržaj sekcije.
* Objekat klase *SymbolTable* sadrži tabelu simbola. Pri tom svaki symbol ima naziv, indeks sekcije u tabeli simbola kojoj pripada, kao i polja da li je definisan i da li je globalan, i vrednost.
* Objekat klase *RelTab* (tabela relokacija) sadrži sve relokacione zapise za tu sekciju (svaka sekcija ima svoju tabelu relokacija).
* Klasa *Asembler je klasa* čiji objekat izvršava oba prolaza i kreira odgovarajuće strukture podataka.

**Opis rada asemblera:**

* Metodom *firstPass*() se prolazi kroz ulazni fajl prvi put i tada assembler “uči” o svim simbolima. On napravi tabelu simbola i upiše sve vrednosti za sve definisane simbole.
* Metodom *secondPass*() se prolazi kroz ulazni fajl drugi put i tada assembler dekoduje instrukcije i direktive na osnovu njih popunjava izlazni fajl (predmetni program).
* Asembler, takođe, izbaci sve lokalne simbole iz tabele simbola pre nego što je upiše u izlazni fajl. U tabeli ostaju samo simboli koji opisuju sekcije i globalni simboli (uvezeni ili izvezeni).
* Asembler prijavljuje greške tako što navede u kojoj je liniji došlo do greške u asembliranju

**Način popunjavanja izlaznog fajla:**

* Prva 3 bajta su iskorišćena za slova E, L, F nalik na pravi elf format.
* Zatim sledeće 4 reči (od po 4 bajta) predstavljaju redom broj text sekcija, broj data sekcija, broj bss sekcija i veličinu tabele simbola.
* Zatim za svaku text sekciju se upiše njen indeks u tabeli simbola, njena veličina i veličina tabele relokacija za tu sekciju.
* Isto se uradi i za svaku data sekciju.
* Za svaku bss sekciju se upiše njen indeks u tabeli simbola i njena veličina.
* Onda se u izlazni fajl upiše sadržaj svih text sekcija, jedna za drugom, a zatim i sadržaj svih data sekcija jedna za drugom.
* Onda se upišu tabele relokacija za svaku text sekciju, potom i za svaku data sekciju (jedan zapis sadrži 3 polja tip int: offset, indeks u tabeli simbola, i tip relokacije).
* Na kraju se upiše tabela simbola (jedan zapis iz tabele sadrži: ime simbola, indeks sekcije kojoj pripada, polje koje određuje da li je globalan, definisan, i na kraju vrednost simbola).

*Testovi*:

|  |  |
| --- | --- |
| Ulazni fajl | Izlazni fajl |
| prvi.txt | prviOutput |
| drugi.txt | drugiOutput |
| treci.txt | treciOutput |
| cetvrti.txt | cetvrtiOutput |

*Neke od bitnih usvojenih pretpostavki:*

* Usvojeno je da kod PC relativnog adresiranja dozvoljavamo samo simbole iz iste tekuće sekcije.
* Aritmetički izrazi su omogućeni ali samo u vidu razlike dva simbola, u slučaju da programer želi da sazna razliku između određenih adresa (npr. počeci različitih sekcija).
* U tabeli simbola se izvoze samo globalni simboli i simboli sekcija (lokalni se odstranjuju).