**Emulator**

**-Sistemski softver-**

**-Domaći zadatak-**

**Generalni opis rešenja:**

* Na ulaz emulatora se dovodi nekoliko fajlova. Prvi fajl treba da bude skript fajl koji određuje kojim redom i na kojim adresama će se učitati sekcije. Ostali fajlovi predstavljaju objektne fajlove koji su izlaz asemblera i koji trebaju biti povezani, učitani u memoriju i izvršeni.
* Objekat klase *Linker* uzima jedan po jedan fajl, i iz svakog čita potrebne podatke kao što su tabele simbola i tabele relokacija. Naravno, i za svaku sekciju njen sadržaj upiše u novi objekat klase Section.
* Kada su svi fajlovi obrađeni, prelazi se na analizu skript fajla. Parsira se red po red, obrađuju izrazi i izračunavaju adrese na koje se smeštaju sekcije. Kada se završi obrada skripta, u memoriju se smeštaju redom sve sekcije koje nisu navedene u skriptu.
* Kada znamo koje su adrese sekcija možemo izračunati sve apsolutne adrese simbola.
* Onda se prelazi na razrešavanje relokacija, za sve učitane fajlove.
* Kada su i relokacije izvršene, program je spreman da bude učitan u memoriju. Učitava se od adrese 0.

Ovime je odrađen posao povezivanja i učitavanja programa u memoriju. Sada treba emulirati učitani program.

Objekat klase *Emulator* obavlja emulaciju programa na sledeći način:

* Čita adresu *reset* prekidne rutine na adresi 0, skače na nju, u njoj se poziva funkcija *main* koju korisnik mora da implementira.
* Emulator čita i obrađuje po jednu instrukciju u svakoj iteraciji glavne petlje.
* Na kraju svake iteracije se proveravaju zahtevi za prekid i obrađuju prekidne rutine za koje postoji zahtev (osim ako nisu maskirani prekidi u PSW-u). Da bi se izvršila timer prekidna rutina mora biti setovan 30. bit u PSW registru.
* Emulacija se završava kada korisnik pritisne taster *ESC* na tastaturi.
* Tastatura i timer poseduju svoje niti koje se izvršavaju paralelno sa glavnim programom emulatora. Timer na svaku sekundu postavi zahtev za prekid, dok tastatura čeka da korisnik pritisne neki taster i tada postavlja zahtev za prekid. Kada se pritisne taster on se postavi u bafer i emulator upiše na odgovarajuću adresu u ulazno-izlaznom prostoru taj karakter.
* Veličina steka koja je dodeljena korisnikovom programu je *1kB.* Stek raste ka nižim adresama, a SP pokazuje na poslednju zauzetu lokaciju.

**Testovi:**

* Kako bi se program regularno izvršavao, korisnik treba da vodi računa da se nikad ne skoči u neki deo koda koji nije za izvršavanje. U testovima je to realizovano tako što u glavnom programu testa postoji beskonačna petlja gde program čeka neki na neki događaj, tipično pritisak nekog tastera.
* Prvi test je demonstracija prekidnih rutina timer-a i tastature. Na početku testa je postavljen odgovarajući bit u PSW registru kako bi timer prekidna rutina mogla da se izvršava. Posle svake protkle sekunde timer rutina se izvršava i ispisuje karakter .(tačka), dok se na pritisak tastera poziva prekidna rutina za tastaturu i na ekranu se ispisuje sukcesivni karakter u odnosu na pritisnut.
* Drugi test poziva učitava dva broja (do maksimalne veličine int, kako bi stali u registar) i onda računa najveći zajednički delilac za ta dva broja (GCD, iterativno). Na ekranu se ispisuje GCD i vraća se na početak i čeka na unos sledeća dva broja. Kraj unosa jednog broja se označava pritiskom tastera ENTER (new line).
* Treći test je kombinacija čitanja jednocifrenog broja, timer prekidne rutine i rekurzivne funkcije. Korisnik unosi jednocifreni broj (bez pritiska tastera ENTER). Sve vreme je aktivna prekidna rutina timer i ispisuje tačku na ekranu periodično. Kada korisnik unese broj, poziva se funkcija fact koja računa faktorijel zadatog broja (prenos kroz registar). Na ekranu se ispisuje izračunata vrednost, i program se vraća na početak da čeka korisnika da unese novi broj.
* Svaki test se završava pritiskom tastera ESC.

U datoj tabeli je naznačeno kom testu pripadaju koji fajlovi.

|  |  |
| --- | --- |
| Prvi test | prviTest.asm prviTestIVT.asm |
| Drugi test | drugiTest.asm drugiTestIVT.asm |
| Treći test | treciTest.asm treciTestFact.asm treciTestIVT.asm |

Komande koje treba uraditi da bi se testirao program (za npr. prvi test):

* ./asembler prviTest.asm prviTest.o
* ./asembler prviTestIVT.asm prviTestIVT.o
* g++ -pthread -o emul emulator.cpp main.cpp linker.cpp symtab.cpp
* ./emul skript.txt prviTest.o prviTestIVT.o

Napomene:

* Fajl skript.txt je zajednički za sve testove.
* U svakom testu se podrazumeva da data sekcija sa adresama prekidnih rutina (predstavlja IVT) se smešta od adrese 0.
* Usvojeno je da prekidna rutina tastature postavlja status registar tako da korisnički program zna da na adresi za ulaz postoji nepročitan karakter. Korisnik je dužan da pročita karakter sa odgovarajuće adrese.
* Ulazno-izlazni proctor i operativna memorija su razdvojeni.