Универзитет у Београду

Електротехнички факултет



**Sistemski softver**

**Emulator**

Jovan Spasojević 0118/2017

# Uvod

Projekat je pisan u c++ in a Linuxu pa da bi mogao da se prevede treba instalirati g++ komandom: **sudo apt install g++**

Da bi se preveo program u src folderu se pokrene komanda:

**g++ emulator.cpp symbol.cpp section.cpp cpu.cpp -lpthread -o emulator**

# Citanje fajlova

Moze vise fajlova da se stavi za argumente a ako hoce da se postavi sekcija na odredjenu lokaciju u memoriji koristi se **-place={ime\_sekcije}@{lokacija}.**

Emulator na pocetku za svaki ulazni fajl ako moze da otvori kreira tabelu simbola i tabelu sekcija. Sve tako kreirane tabele se stavljaju u dva niza jedan za tabele sekcije drugi za tabele simbola.

Takodje postoji tabela sekcija izvrsnog programa i tabela simbola izvrsnog programa.

Na pocetku pokusavaju da se rasporede sekcije koristeci place argumente. Ako se desi da ne postoji sekcija u tabelama ispisuje se greska i zavrsava program.

Posle toga se za svaku tabelu sekcija iz niza pronalaze sekcije i smestaju u tabelu sekciju izvrsnog programa. Za simbole sekcije koje su ovako obradjene njihove vrednosti se menjaju u tabelama sekcija niza da pokazuju na nove vrednosti posle smestanja u izvrsni program. Takodje se vrednosti iz niza bajtova sekcije prebacuju u odgovarajucu adresu niza MemBase koji predstavlja memoriju emulatora.

Ako sve prodje ok prelazi se onda na uzimanje svih simbola iz niza tabela simbola koji su definisani i prebacivanje njihovih ulaza u tabelu simbola izlaznog fajla. Ako dodje do konflikta ispisuje se greska i zavrsava program.

Posle ovog prolaza prolaze se za svaku tabelu sekcija da li postoji neki simboli koji nisu definisani. Ako postoje zavrsava se program.

Nakon ovoga za svaku relokaciju u nizu u tabelama sekcija pronalazi se simbol koji se koristio i njegova nova vrednost(nadje se sekcija u kom pripada simbol i doda se njegov offset) i upisuju se na adresu u MemBase niz.

## Emulator

Za emulator pocetnu tacku treba smestiti u prekidnu rutinu za ulaz 0 jer procesor uvek pocinje od njega. Stek mora da se inicijalizuje od strane programera ili dolazi do greske pri koriscenju instrukcija koje koriste stek.

Za svaku instrukciju u izvrsnom programu cita se bajt po bajt i desifruje.

U zavisnosti od desifrovanog ili pokusava da odradi instrukciju ili baca izuzetak za pogresnu insrukciju.

Ako se pokusa pristup preko memorijske granice ili stek predje granice memorije program se zavrsava i ispisuje odgovarajuca poruka.

Postoji struktura psw koja se koristi da imitira psw registar. Ima funkcije za postavljanje vrednosti bitnih bitova i za citanje i postavljanje pojedinacnih bajtova ili celih reci.

Postoje i strukture za registre koje ima funkcije za citanje i postavljanje bajtova ili celih reci.

Postoji csip koji na pocetku svake while petlje emulatora dobije vrednost PC-a. Onda se umesto PC-a inkrementira csip. Razlog za ovo je u slucaju greske da nemoramo da menjamo PC-a jer nismo njega uopste inkrementirali.

Csip-ova vrednost se daje PC-u kada se uspesno izvrsi instrukcija,ako treba da mu se doda(nece zbog jumpova na primer).

# Testovi

Test1:

Ulaz:sa\_casa\_izmenjen direktorijum

Izlaz:Greska jer je pokusano definisanje istog simbola u 2 fajla.

**./emulator -place=text@0x4000 proba1.o proba2.o proba3.o**

Tes2:

Ulaz:sa\_casa direktorijum

Izlaz:Greska jer ne moze program da se smesti u memoriju

**./emulator -place=text@0xFFFF proba1.o proba2.o proba3.o**

Tes3:

Ulaz:sa\_casa direktorijum

Izlaz:Greska jer nisu definisani svi simboli.

**./emulator -place=text@0x4000 proba1.o proba2.o**

Test4:

Ulaz:sa\_casa direktorijum

Izlaz:Greska jer se sekcije preklapaju

**./emulator -place=text@0x4000 -place=data@0x4005 proba1.o proba2.o proba3.o**

Test5 :

Ulaz:test\_svega direktorijum

Izlaz:Test prolazi kroz sve instrukcije i izvsrava ih. Ispisuje vrednosti registra r3 u razlicitim trenutcima.

**./emulator -place=ivt@0x0000 -place=timer@0x0064 -place=program@0x4000 emul2.o emul1.o -place=keyboard@0x00C8 -place=error@0x012C emul3.o -place=test@0x01F4 test\_all.o**

Test6:

Ulaz:**ispis 1000 brojeva** direktorijum

Izlaz:U fajlu emul1.o definisan je deo vezan za inicijalizaciju sekcija i ivt tabela i deo glavnog programa. Emul2.o testira neke instrukcije sa razlicitim tipovima adresiranja. Emul3.o sadrzi funkciju koju poziva emul1.o.

**./emulator -place=ivt@0x0000 -place=timer@0x0064 -place=program@0x4000 emul2.o emul1.o -place=keyboard@0x00C8 -place=error@0x012C emul3.o**

Test7: **Testiranje tastature i tajmera**

Ulaz**:** **test\_tajmer\_keyboard direktorijum**

Izlaz:Svaki klik tastature ispisuje karakter i usporava tajmer za jedan nivo vise ako se pritisne karakter ’u’ i smanjuje ako se pritisne karakter ’d’. Tajmer ispisuje tick kada se izvrsavan njegova prekidna rutina. Program se zavrsava pritiskom ’e’.(Case sensetive je program)

**./emulator -place=ivt@0x0000 -place=timer@0x0064 -place=program@0x4000 emul2.o emul1.o -place=keyboard@0x00C8 -place=error@0x012C emul3.o**