**Sistemski softver – Zadatak 1**

**Opis problema**

Zadatak je napraviti dvoprolazni asembler, ulaz je tekstualni fajl, koji se prevodi do linije .end, izlaz asemblera je tekstualni fajl baziran na skolskoj varijanti elf formata sa tabelom simbola, relokacionim tabelama i sadržajem sekcija.

**Opis rešenja**

Program prima 3 argumenta preko komadne linije: nastavak -o,naziv izlaznog objektnog fajla(zavrsava se sa .o nastavkom) I naziv asemblerskog fajla(zavrsava se sa .s nastavkom).

Sastoji se fajlova: main.cpp (u kome se nalazi main funkcija), i types.h ( pomoćne strukture podataka ulaz u tabeli simbola, ulaz u relokacionoj tabeli,struktura koja opisuje adresiranje operanda i struktura za opis sekcije).

U main-u prvo se desava prvi prolaz kojem se šalju referenca na ulazni fajl gde se nalazi asemblerski kod. Čitanjem red po red i tokenizaciom se indentifikuju labele, popunjava se tabela simbola, određuju se početne adrese i veličine korišćenih sekcija.

U main-u se zatim desava drugi prolaz. Čitanjem red po red(istog onog fajla kao iz prvog prolaza) i tokenizacijom se vrši definisanje globalnih simbola kao takvih, popunjavanje sadržaja sekcija heksadecimalnim ciframa u little-endian formatu, kao i indentifikovanje nepoznatih simbola i pravljenje relokacionih zapisa gde god je korišćeno pc relativno adresiranje sa nerazrešenim simbolom ili apsolutna adresa.

Posle drugog prolaza pravi se predmetni fajl u modifikovanom elf formatu i upisuju se redom podaci dobijeni asembliranjem: tabela simbola ( naziv, izvorna sekcija, pomeraj od početka sekcije, lokalitet i indentifikacioni broj), relokacione tabele simbola za svaku korišćenu sekciju (pomeraj od početka sekcije, vrsta relokacije, indentifikacioni broj iz tabele simbola) i sadržaji sekcija u heksadecimalnom zapisu.

Kada se program uspesno izvrsi asembliranje na konzoli se ispisuje program zavrsen bez gresaka.

**Uputstvo za prevođenje i pokretanje programa**

**Prvo se instalira g++ pomocu naredbe:**

sudo apt-get install g++

**Zatim posto sam korstio eclipse IDE potrebno je bilo instaliriti java vituelnu masinu(samo zbog eclipse-a ne zbog projekta)**

**sudo apt-get install openjdk-7-jre-headless**

Program se može pokrenuti unutar Ubuntu virtulne mašine preuzete iz laboratorije 26. Pre pokretanja programa potrebno je instalirati i podesiti GCC kompajler verzije 6 ili novije.

Program se prevodi sledećom komandom:

g++ -std=c++0x main.cpp types.h -o izvrsniProgram

Program se pokrece sledecom komandom:

./izvrsniProgram -o nazivIzlaznogFajla.o nazivUlaznogFajla.s

**Testovi**

Cilj je bio ispitati sto vise kombinacija instrukcija.

test.s je bukvalno 9. Zadatak sa vezbi I proverene su sve moguce kombinacije oko PC reletavnih I abolutnih relokacija,test1.s sadrzi detaljnije provere svih mogucis kombinacija adresiranja koje je moguce upotrebititi.

test.s

**.global** a, c

**.extern** b

.equ g, 7

**.text**

jeqw $a

jeqw $e

jeqw b

jeqw $d

addw r1h[a], [r5l]

**d:** **.word** d

movw r0, b

movw c, r0

movw r0, e

**.data**

.skip 8

**e:** **.word** a

**.word** e

**.word** .bss

**a:** **.word** b

.bss

**c:** .skip 8

.end

test.o

#tabela simbola

# ime sek vr. vid. r.b.

.bss .bss 0 l 8

.data .data 0 l 7

.text .text 0 l 4

a .data 14 g 0

b UND 0 g 2

c .bss 0 g 1

d .text 21 l 6

e .data 8 l 5

g CON 7 l 3

#.rel.text

# ofset tip vr.

2 PC\_REL 0

6 PC\_REL 7

10 ABS 2

18 ABS 0

21 ABS 4

26 ABS 2

30 ABS 1

36 ABS 7

#.rel.data

# ofset tip vr.

8 ABS 0

10 ABS 7

12 ABS 8

14 ABS 2

#.text 38

a4 8e fe ff a4 8e 06 00 a4 a0 00 00 a4 8e 05 00

2c 83 00 00 4a 15 00 24 20 a0 00 00 24 a0 00 00

20 24 20 a0 08 00

#.data 16

00 00 00 00 00 00 00 00 00 00 08 00 00 00 00 00

test1.s

.equ g,4

**.text**

movw r1,&g

orb r1h[a],r1

orw r1h[255],r1

addw r1,r2

addb r1h,r2h

addb r1h,255

subw r1,\*255

andw r1,[r2]

andb r1h,[r2h]

jmpw $a

ret

**.data**

.skip 5

**a:**

.skip 5

.end

test1.o

#tabela simbola

# ime sek vr. vid. r.b.

.data .data 0 l 3

.text .text 0 l 1

a .data 5 l 2

g CON 4 l 0

#.rel.text

# ofset tip vr.

7 ABS 3

37 PC\_REL 3

#.text 40

24 22 00 04 00 60 63 05 22 64 83 ff 00 22 2c 22

24 28 23 25 28 23 00 ff 34 22 a0 ff 00 5c 22 44

58 23 45 9c 8e 03 00 c4

#.data 10

00 00 00 00 00 00 00 00 00 00