Sistemsko programiranje (IR3SP)

Domaći zadatak:

Tema: Konstrukcija jednoprolaznog asemblera za Von-Neuman arhitekturu

Student: Golubović Aleksandar 36/2012

Opis asemblera:

Implementirani asembler ima definisanu sintaksu po uzoru na GNU asembler. Moguće je definisati odvojene sekcije za kod, podatke i neicijalizovane podatke. Omogućena je upotreba komentara koji moraju početi znakom # i prostiru se do kraja ulazne linije. Za punjenje registara konstantama mogu se koristiti konstantni i simbolički literali.

Ograničenja asemblera:

- Simboli koji su definisani spolja moraju se eksplicitno uvesti direktivom .extern, za svaki simbol koji nije definisan assembler prijavljuje grešku.
- U direktivama .wordh i .wordl je dozvoljeno korišćenje simbola, dok kod direktive .char nije dozvoljeno.
- Instrukcije i registri su case-sensitive, potrebno je sve pisati malim slovima engleskog alfabeta.
- Instrukcije su dozvoljene samo u .text sekcijama.
- U .bss sekciji nisu dozvoljeni inicijalizovani podaci

Opis rešenja:

Obrada izvornog koda realizuje se u jednom prolazu. Svaka učitana linija deli se na tokene koji predstavljaju labelu, mnemonik ili direktivu, i operande instrukcije. Informacije o svakoj liniji čuvaju se u strukturi Line na osnovu kojih se vrši dalja obrada. Kada se naiđe na definiciju labele ona se dodaje u tabelu simbola kao lokalna, ukoliko pre ili posle toga ne postoji direktiva .global koja bi je označila

kao izvoznu. Višestruka definicija simbola nije dozvoljena, javlja se greška. Sve tabele koje se koriste u toku asembliranja (tabela simbola, tabela sekcija) implementirane su kao objekti klase TextTable.

Prevođenje:

gcc -o asembler FileManager.cpp Line.cpp RetTable.cpp Source.cpp SymTable.cpp TextTable.cpp Exception.cpp -lstdc++ -std=c++11

Pokretanje:

./asembler ulaz.txt izlaz.txt

Testovi:

Test 1 – Više text i data sekcija sa različitim početnim adresama.

Ulaz:

```
.global a, b
.text 56 #text sekcija
add r5, r3, 10
a:      sub r4, r3, 63
c:      and r8, r10
.data9 55
.long a
.wordl c
.text1
push psw #stavlja psw na
stek
b:      pop r0
.text4 989
ldc r2, c
mov r3, r5
.bss
.skip 10
.end
```

Izlaz:

```
#symTable
ime             sek.      vr.       vid.      r.b.      vel.      adr.      RWE
UND             0         0         l         0         0         0         ---
a               3         4         g         1
b               6         4         g         2
.text           3         0         l         3         12        56        r-e
c               3         8         l         4
.data9          5         0         l         5         6         55        rw-
.text1          6         0         l         6         8         0         r-e
.text4          7         0         l         7         12        989       r-e
.bss            8         0         l         8         10        0         rw-
#ret.text
#ofset          tip             vr.
#.text
0A 00 4C 01 3F 00 0C 05 00 00 28 16
#ret.data9
#ofset          tip             vr.
0               R_ABS_32        1
4               R_ABS_16L       3
#.data9
00 00 00 00 08 00
#ret.text1
#ofset          tip             vr.
#.text1
00 00 00 5E 00 00 00 60
#ret.text4
#ofset          tip             vr.
0               R_ABS_16H_5     3
4               R_ABS_16L_5     3
#.text4
00 00 A0 7C 00 01 80 7C 00 00 D4 64
```

Test 2: U ovom primeru testira se korišćenje asemblerskih direktiva i učitavanje konstanti i simboličkih literala u register, kao i komentari.

Ulaz:

```
.global a
.extern b
.text 90
ldcl r1, a#komentar
a:      je x
.long m
c:      .skip 3#komentar2323
.data1
.long 16, b
x:      .char 1, 6, d
.wordh x, 131071#okofkofka
.skip 5
m:      .align 8
.global c
.end
mov r4, r2
ddfksaofkoaskf
asdka
a
```

Izlaz:

```
#symTable
ime             sek.      vr.       vid.      r.b.      vel.      adr.      RWE
UND             0         0         l         0         0         0         ---
a               3         4         g         1
b               -1        0         g         2
.text           3         0         l         3         15        90        r-e
x               7         8         l         4
m               7         20        l         5
c               3         12        g         6
.data1          7         0         l         7         24        0         rw-
#ret.text
#ofset          tip             vr.
0               R_ABS_16L_5     1
4               R_PC_21         7
8               R_ABS_32        7
#.text
00 00 40 7C 04 00 00 2E 14 00 00 00 00 00 00
#ret.data1
#ofset          tip             vr.
11              R_ABS_16H       7
4               R_ABS_32        2
#.data1
10 00 00 00 00 00 00 00 01 06 64 00 00 01 00 00 00 00 00 00 00 00 00 00
```

Test 3: U ovom primeru se testiraju skokovi
Ulaz:

```
.global a, b
.text 56 #text sekcija
je a
a:      jmp [x]
c:      call poziv
jne lok
lok:    jge g
.data39 23
.long a
.char 98
x:      .wordl c
.text1
poziv: push psw #stavlja psw
na stek
b:      ret 8
.text7
ldc r2, c
mov r3, r5
.bss
g:      .skip 3
.end
in r4, r6
```

Izlaz:

#symTable

ime	sek.	vr.	vid.	r.b.	vel.	adr.	RWE
UND	0	0	1	0	0	0	---
a	3	4	g	1			
b	10	4	g	2			
.text	3	0	1	3	20	56	r-e
x	9	5	1	4			
c	3	8	1	5			
poziv	10	0	1	6			
lok	3	16	1	7			
g	12	0	1	8			
.data39	9	0	1	9	7	23	rw-
.text1	10	0	1	10	8	0	r-e
.text7	11	0	1	11	12	0	r-e
.bss	12	0	1	12	3	0	rw-

#ret.text

#ofset	tip	vr.
0	R_PC_21	1
4	R_PC_24	9
8	R_PC_21	10
16	R_PC_21	12

#.text

```
FC FF 1F 2E 01 00 00 5B FC FF 1F 56 00 00 00 32 FC FF 1F 36
```

#ret.data39

#ofset	tip	vr.
0	R_ABS_32	1
5	R_ABS_16L	3

#.data39

```
00 00 00 00 62 08 00
```

#ret.text1

#ofset	tip	vr.

#.text1

```
00 00 00 5E 08 00 00 58
```

#ret.text7

#ofset	tip	vr.
0	R_ABS_16H_5	3
4	R_ABS_16L_5	3

#.text7

```
00 00 A0 7C 00 01 80 7C 00 00 D4 64
```