Elektrotehnički fakultet,

Univerzitet u Beogradu

Dvoprolazni asembler

Sistemski softver

Domaći zadatak

Sofija Tanasković 2014/0214

**Opis problema**

Prvi deo projekta se odnosi na pisanje dvoprolaznog asemblera čiji ulaz predstavlja tekstualni fajl čija je sintaksa navedena u postavci zadatka. Asembler tokom prvog prolaza popunjava tabelu simbola koju čine lokalne i globalne promenljive i sekcije, a zatim tokom drugog prolaza parsira direktive i instrukcije i generiše relokacionu tabelu. Izlaz asemblera je tekstualni fajl nalik na ELF format objektnih fajlova.

**Opis rešenja**

Na početku se vrši čitanje ulaznog fajla red po red (dok ne dođe do '\n' ili ';' ), tom prilikom se brišu komentari i prazni redovi. Svaki pročitani red se zatim deli na podstringove, koji se čuvaju u jednom vector<string> , pomoću funkcije:

void split(const string &s, char\* delim, vector<string> &v);

Kada stigne do tokena .end, čitanje ulaznog fajla se završava. Zatim se tako pročitan fajl prosleđuje funkciji:

void passOne(vector<Symbol\*>& symtab, vector<vector<string>> asem, vector<string>& greske);

Ova funkcija predstavlja prvi prolaz asemblera. Ona se sastoji iz glavne petlje koja čita red po red (tj. po jedan vector<string>). Za svaki od redova određuje se vrsta prve pročitane reči, proverava se da li je ona labela, oznaka sekcije, mnemonik, direktiva ili nešto drugo. U zavisnosti od toga se poziva odgovarajuća funkcija kojoj se prosleđuje data reč na dalju obradu. I u prvom i u drugom prolazu koristi se promenljiva location\_counter koja ukazuje na adresu unutar tekuće sekcije, prilikom svake promene sekcije ona se resetuje na 0, a uvećava se u zavisnosti od direktive (DB uvećava za 1, DW za 2, DD za 4) ili u zavisnosti od toga da li je u pitanju instrukcija od 4B ili od 8B (4B za regdir i regind načine adresiranja, 8B za ostale). Ako se u prvom prolazu pročita neka od instrukcija, samo se uvećava location\_counter, a ako se pročita labela, sekcija ili definiciija, onda se formira novi Symbol koji se ubacuje u tabelu simbola. Tabela simbola je predstavljena jednim nizom simbola za koji je iskorišćena struktura vector.

Ako se u nekom od pročitanih redova nađe greška, u fajl errors.txt biće ispisana odgovarajuća poruka o grešci. Nakon prvog prolaza biće formirana tabela simbola koja se dalje prosleđuje funkciji:

void passTwo(vector<Symbol\*>& symtab, vector<vector<string>>& asem, vector<RelTable\*>& reltables, vector<string>& greske);

Ova funkcija predstavlja drugi prolaz asemblera. Ona se sastoji iz glavne petlje koja opet čita red po red (tj. po jedan vector<string>). Obrada redova je slična kao u prvom prolazu, s tim što se sada obrađuju i instrukcije, a ne obrađuju se direktive .global i DEF, jer su obrađene u prvom prolazu. A umesto tabele simbola se prave relokacioni zapisi. Funkcija koja određuje kom simbolu je potrebna relokacija je:

string relocationNeeded(vector<string> str, vector<Symbol\*>& symtab, vector<string>& greske);

U drugom prolazu se obrađuje i ORG direktiva tako što ce njena vrednost dodaje svim simbolima sekcije koja sledi ORG direktivi. Klase i strukture koje su korišćene su Symbol, Instruction, State, RelSymbol, RelTable i one predstavlaju simbol iz tabele simbola, instrukciju, trenutno stanje (čine ga trenutna sekcija, indikator greške, itd), relokacioni simbol i relokacionu tabelu, respektivno.

**Pokretanje**

Pre buildovanja je potrebno pokrenuti sledeće komande koje služe za instaliranje nove verzije g++ kompajlera:

|  |
| --- |
|  |

sudo add-apt-repository ppa:ubuntu-toolchain-r/test  
sudo apt-get update  
sudo apt-get install gcc-4.9  
sudo update-alternatives --install /usr/bin/gcc gcc /usr/bin/gcc-4.9 50  
sudo apt-get install g++-4.9  
sudo update-alternatives --install /usr/bin/g++ g++ /usr/bin/g++-4.9 50

Zatim se program builduje sledećom komandom:

g++ -std=c++11 –g –o asembler klase.cpp drugi.cpp mejn.cpp

A pokreće se komandom:

./putanja/asembler /putanja/naziv\_ulaznog\_fajla.s

**Testovi**

**Test 1:**

.data

pocetakdata:

a DEF 10

b DEF 20

c DEF 30

DB c - b

DB lab2

lab1:

e DEF 40

DD e-a-b

DD 8 DUP ?

krajdata:

.text

LOADUB R1, #1

LOADUB R2, R1

LOADUB R3, $5

ADD R1, R1, R1

SUB R2, R1, R1

lab2:

STOREB R4, [R2+e+b-a-c]

JZ R1, krajdata

JMP krajdata-pocetakdata+lab1

MUL R8, R8, R9

PUSH R10

PUSH R11

JLEZ R1, [R8]

INT R12

.end

**Izlaz 1:**

#TabelaSimbola

SEG 1 .data 1 0x00 0x26 RW

SYM 2 pocetakdata 1 0x00 L

SYM 3 a -1 0x0a L

SYM 4 b -1 0x14 L

SYM 5 c -1 0x1e L

SYM 6 lab1 1 0x02 L

SYM 7 e -1 0x28 L

SYM 8 krajdata 1 0x26 L

SEG 9 .text 9 0x00 0x48 RX

SYM 10 lab2 9 0x1c L

#rel.data

0x01 A 9

.data

0a 1c 0a 00 00 00 00 00 00 00 00 00 00 00 00 00

00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

00 00 00 00 00 00

#rel.text

0x28 A 1

0x30 A 1

.text

18 08 80 10 01 00 00 00 18 10 01 10 18 18 f1 10

05 00 00 00 40 08 01 30 40 08 02 31 00 20 e2 11

14 00 00 00 00 00 c1 04 26 00 00 00 00 00 c0 02

28 00 00 00 40 42 08 32 00 00 0a 20 00 00 0b 20

00 40 41 09 00 00 0c 00

#end

**Test 2:**

ORG 0x100

.data

.global lab4

a DEF 1

b DEF 2

DD lab1

DD lab2

DD lab3

DD 16 DUP ?

DW 16 DUP 4

ORG 0x5000

.text.0

regi DEF 0x80

.global START

START:

LOAD R2, #2

PUSH R2

POP R11

LOAD R3, b-a

PUSH R3

POP R12

lab1: LOAD R4, [R2-6]

PUSH R4

POP R13

CALL lab3

.text.2

lab2:

ADD R2, R2, R3

SUB R3, R3, R2

AND R2, R2, R3

ASL R6, R7, R8

ASR R9, R10, R11

JMP lab1

.text.3

lab3:

PUSH R4

LOAD R4, SP

LOAD R0, R4

JNZ R0, $lab2

.end

**Izlaz 2:**

#TabelaSimbola

SEG 1 .data 1 0x100 0x6c RWO

SYM 2 lab4 0 0x00 G

SYM 3 a -1 0x01 L

SYM 4 b -1 0x02 L

SEG 5 .text.0 5 0x5000 0x38 RXO

SYM 6 regi -1 0x80 L

SYM 7 START 5 0x5000 G

SYM 8 lab1 5 0x5020 L

SEG 9 .text.2 9 0x00 0x1c RX

SYM 10 lab2 9 0x00 L

SEG 11 .text.3 11 0x00 0x14 RX

SYM 12 lab3 11 0x00 L

#rel.data

0x100 A 5

0x104 A 9

0x108 A 11

.data

20 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

00 00 00 00 00 00 00 00 00 00 00 00 04 00 04 00

04 00 04 00 04 00 04 00 04 00 04 00 04 00 04 00

04 00 04 00 04 00 04 00 04 00 04 00

#rel.text.0

0x5034 A 11

.text.0

00 10 80 10 02 00 00 00 00 00 02 20 00 00 0b 21

00 18 c0 10 01 00 00 00 00 00 03 20 00 00 0c 21

00 20 e2 10 fa ff ff ff 00 00 04 20 00 00 0d 21

00 00 c0 03 00 00 00 00

#rel.text.2

0x18 A 5

.text.2

c0 10 02 30 80 18 03 31 c0 10 02 35 00 3a 06 39

c0 52 09 3a 00 00 c0 02 20 50 00 00

#rel.text.3

0x10 R 9

.text.3

00 00 04 20 00 20 10 10 00 00 04 10 00 00 f1 05

00 00 00 00

#end

**Test 3:**

ORG 0x0

.data

DD sp

DD 2 DUP ?

DD greska

DD tajmer

DD tastatura

DD 30 DUP ?

.text.0

out\_reg DEF 0x80

in\_reg DEF 0x84

.global START

START:

LOAD R0, #0

LOAD R1, #1

LOAD R2, #2

LOAD R3, #3

ADD R4, R3, R2

PUSH R1

PUSH R2

PUSH R3

MUL R6, R2, R3

POP R3

POP R2

POP R1

SUB R7, R6, R0

JMP START

.text.1

greska:

RET

.text.2

tastatura:

RET

.text.3

tajmer:

RET

.bss

sp:

DB 0x100 DUP ?

.end

**Izlaz 3:**

#TabelaSimbola

SEG 1 .data 1 0x00 0x90 RWO

SEG 2 .text.0 2 0x00 0x4c RX

SYM 3 out\_reg -1 0x80 L

SYM 4 in\_reg -1 0x84 L

SYM 5 START 2 0x00 G

SEG 6 .text.1 6 0x00 0x04 RX

SYM 7 greska 6 0x00 L

SEG 8 .text.2 8 0x00 0x04 RX

SYM 9 tastatura 8 0x00 L

SEG 10 .text.3 10 0x00 0x04 RX

SYM 11 tajmer 10 0x00 L

SEG 12 .bss 12 0x00 0x100 RW

SYM 13 sp 12 0x00 L

#rel.data

0x00 A 12

0x0c A 6

0x10 A 10

0x14 A 8

.data

00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

#rel.text.0

0x48 A 5

.text.0

00 00 80 10 00 00 00 00 00 08 80 10 01 00 00 00

00 10 80 10 02 00 00 00 00 18 80 10 03 00 00 00

80 18 04 30 00 00 01 20 00 00 02 20 00 00 03 20

c0 10 06 32 00 00 03 21 00 00 02 21 00 00 01 21

00 30 07 31 00 00 c0 02 00 00 00 00

#rel.text.1

.text.1

00 00 00 01

#rel.text.2

.text.2

00 00 00 01

#rel.text.3

.text.3

00 00 00 01

#rel.bss

.bss

00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

#end

**Test 4:**

.text.0

lab1:

XOR R1, R1, R2, R2

NOT R1, R2

STORE R0, #0x200

.text.1

INT R2

RET

PUSH R1 R2

POP R11

ASL R3, R3, R3

lab1:

MOD R5, R6, R7 NOT R8, R9

1kraj:

JMP R6

CALL lab1

.end

**Izlaz 4:**

#TabelaSimbola

SEG 1 .text.0 1 0x00 0x10 RX

SYM 2 lab1 1 0x00 L

SEG 3 .text.1 3 0x00 0x2c RX

#rel.text.0

.text.0

00 10 01 38

#rel.text.1

0x18 A 1

.text.1

00 00 02 00 00 00 00 01 00 00 0b 21 c0 18 03 39

00 00 c0 03 00 00 00 00

#end

**Greske 4:**

Labela lab1 vec postoji

Labela nije ispravna

Broj argumenata u liniji nije odgovarajuci

U STORE instrukciji ne sme biti neposredno adresiranje

Registar ne postoji

Broj argumenata u liniji nije odgovarajuci

**Test 5:**

ORG 0x100

.data

DD a

DD 8

lab:

DD 12

DB 16 DUP 1

a DEF 3

b DEF 4

.text

LOAD R1, a\*+b

ADD R1, R1, R1

.end

**Izlaz 5:**

#TabelaSimbola

SEG 1 .data 1 0x100 0x1c RWO

SYM 2 lab 1 0x108 L

SYM 3 a -1 0x03 L

SYM 4 b -1 0x04 L

SEG 5 .text 5 0x00 0x0c RX

#rel.data

.data

03 00 00 00 08 00 00 00 0c 00 00 00 01 01 01 01

01 01 01 01 01 01 01 01 01 01 01 01

#rel.text

.text

00 08 c0 10 00 00 00 00 40 08 01 30

#end

**Greske 5:**

Izraz je nekorektan