**Електротехнички факултет Београд**

**Двопролазни асемблер**

Студент:

Предраг Шијан 2013/0541

Београд, август 2019.

**Опис проблема:**

Задатак је направити двопролазни асемблер на програмском језику C++ који треба превести и покренути на оперативном систему Linux. Двопролазни асемблер који је потребно реализовати је детаљно описан унутар pdf фајлова postavka и prilog.

**Опис фајлова:**

Програм се састоји од следећих фајлова:

- simbol.h: унутар овог фајла се налази класа Simbol унутар које чувамо све потребне вредности за табелу симбола осим имена секције којој симбол припада

- relokacije.h: унутар овог фајла се налази класа Relokacija унутар које чувамо све потребне вредности за табелу реалокација

- sekcija.h и sekcija.cpp: унутар ових фајлова се налази класа Sekcija која осим што чува све потребне вредности секције за табелу симбола такође чува и векторе свих симбола и реалокација дефинисаних унутар те секције

- asembler.h и asembler.cpp: унутар ових фајлова се налази класа Asembler која има функције за први и други пролаз као и функцију за испис програма (уколико је програм успешно завршен)

- instrukcije.h и instrukcije.cpp: унутар ових фајлова имамо класу Instrukcije која садржи функције за обраду инструкција. Неке од функција ове класе дају тачан број бајтова које одређена инструкција заузима као и функције које обрађују сваку од тих инструкција

-main.cpp: унутар овог фајла се налази функција main за покретање програма и у њој правимо објекат класе и позивамо функције те класе.

- ulaz.txt: унутар ове текстуалне датотеке се налази програм који наш двопролазни асемблер треба да обради

**Упутство за покретање програма:**

Све фајлове убацити унутар једног фолдера на десктопу са произвољним именом (у овом случају то је фолдер Projekat). Након што смо то урадили треба отворити терминал и укуцати команду:

cd ./Desktop/Projekat

Ова наредба приступа фолдера унутар кога смо ставили све фајлове. Након тога командом:

g++-5 -o ProjekatSS main.cpp asembler.cpp instrukcije.cpp sekcija.cpp -std=c++11

која преводи наш програм и прави фајл ProjekatSS помоћу кога ћемо покренути програм. Наредбом:

./ProjekatSS "ulaz.txt"

покрећемо наш програм. У случају да се програм успешно изврши направиће се текстуална датотека izlaz.txt унутар које ћемо моћи видети резулат рада двопролазног асемблера. Уколико са друге стране дође до неке грешке на терминалу ће се исписати где се грешка налази да би могли да је уклонимо.

**Тестирање:**

У даљем тексту се налази садржај улазних датотека помоћу којих сам тестирао двопролазни асемблер, као и садржај излазних излазних датотека које су направљење након извршавања програма. Тестови су написани тако да тестирају све могуће начине адресирања, као и различите инструкције и директиве.

**ulaz1.txt**

.global a, b, c

.extern d

.text

a: addw r0, 10

movw r5, a

pushw b

.data

b: .skip 5

.word b, a

.byte 10, c, -19

e: .equ f, 20

.align 4

jne $kraj

.equ g, 59

halt

.section sek

c: addb [r7l], 5

mul r2, [r1]

kraj: ret

.end

**izlaz1.txt**

Tabela simbola:

Ime Sekcija Vrednost Vidljivost Redni broj

UND 0 l 0

d UND 0 g 11

.text text 0 l 1

a text 0 g 2

.data data 0 l 3

b data 0 g 4

e data 8 l 5

f data 20 l 6

g data 59 l 7

.sek sek 0 l 8

c sek 0 g 9

kraj sek 7 l 10

Relokacioni zapisi:

.ret.text

Offset Tip Vrednost

8 R\_386\_16 2

12 R\_386\_16 4

.ret.data

Offset Tip Vrednost

5 R\_386\_16 4

7 R\_386\_16 2

10 R\_386\_8 9

18 R\_386\_PC16 8

.ret.sek

Offset Tip Vrednost

Sekcije:

text

2C 20 00 0a 00 24 2A A0 00 00 8C A0 00 00

data

00 00 00 00 00 00 00 00 00 0a 00 ed 00 00 00 00 AC 8E 05 00 0C

sek

28 4E 00 05 3C 24 42 C4

**ulaz2.txt**

.extern skocko

.text

xchgw [r3], r6[25]

lab: subw r2, 32

xorb r0l, r6h[lab]

pab: notw r7

cmpw r3, 0

jeq $pab

.section kalimero, 111

pushw &lab

movw r3, 7

cmpw r3, 0

jeq $dalje

petlja: subw r3, 1

cmpw r3, 0

jeqw $dalje

jmp $petlja

dalje: call $skocko

.end

**izlaz2.txt**

Tabela simbola:

Ime Sekcija Vrednost Vidljivost Redni broj

UND 0 l 0

skocko UND 0 g 7

.text text 0 l 1

lab text 5 l 2

pab text 14 l 3

.kalimero kalimero 0 l 4

petlja kalimero 18 l 5

dalje kalimero 36 l 6

Relokacioni zapisi:

.ret.text

Offset Tip Vrednost

12 R\_386\_8 1

.ret.kalimero

Offset Tip Vrednost

2 R\_386\_16 1

38 R\_386\_PC16 7

Sekcije:

text

14 45 8C 19 00 34 24 00 20 00 68 20 6D 06 54 2E 4C 25 00 00 00 A4 8E f5 ff

kalimero

8C 00 05 00 24 25 00 07 00 4C 25 00 00 00 A4 8E 12 00 34 25 00 01 00 4C 25 00 00 00 A4 8E 04 00 9C 8E ee ff BC 8E fe ff

**ulaz3.txt**

.global lab1

.text

pocetak: movb [r4l], sph

divb r2h, 2

andb r1l[lab1], 7

testb \*116, 0

.bss

.skip 3

ac\_dc: .byte 7

.word 96

jmp $lab1

.section etf

lab1: orb spl, &pocetak

cmpb spl, 0

jgt $pocetak

pushb sph

iret

.end

**izlaz3.txt**

Tabela simbola:

Ime Sekcija Vrednost Vidljivost Redni broj

UND 0 l 0

.text text 0 l 1

pocetak text 0 l 2

.bss bss 0 l 3

ac\_dc bss 3 l 4

.etf etf 0 l 5

lab1 etf 0 g 6

Relokacioni zapisi:

.ret.text

Offset Tip Vrednost

9 R\_386\_8 6

.ret.bss

Offset Tip Vrednost

8 R\_386\_PC16 6

.ret.etf

Offset Tip Vrednost

3 R\_386\_8 1

10 R\_386\_PC16 1

Sekcije:

text

20 48 2D 40 25 00 02 58 62 04 00 07 70 A0 74 00 00 00

bss

00 00 00 07 60 00 9C 8E fe ff

etf

60 2C 00 00 48 2C 00 00 B4 8E fe ff 88 2D CC