\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Електротехнички факултет у Београду

Катедра за рачунарску технику и информатику

***Предмет****:* Системски софтвер (13Е113СС)

***Наставници***: др Драган Бојић, др Саша Стојановић

***Асистенти:*** Марко Мићовић, дипл. инж.

***Испитни рок:*** јун 2020.

***Датум:*** 6.6.2020.

***Кандидат:*** Марко Кујунџић

***Број индекса:*** 0418/2016

***Тема:*** Једнопролазни асемблер

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# Опис решења

Први део пројекта представља једнопролазни асемблер, док је други део пројекта интерпретативни емулатор који извршава асемблерске инструкције.

Једнопролазни асемблер ради на следећи начин:

У првом и једином пролазу кроз улазни фајл, скупљамо све симболе и убацујемо их у табелу симбола. После тога, попуњавамо места у коду где вредност симбола у тренутку проласка није била позната, и тиме остављамо конзистентан садржај машинског кода у секцијама, и правимо релокационе записе тамо где је потребно.

Повезивач и Пунилац, након обраде машинског кода уграђеног у секције и пуњења програма у меморију, остављају емулатору све што му је потребно за извршавање асемблерских инструкција.

# Структура решења

Сваки ред улазног фајла се издели на жетоне, који представљају тип инструкције или директиве, лабелу или операнд. Провером првог жетона, знамо да ли се ради о инструкцији, лабели или директиви, и у зависности од случаја, настављамо са даљом обрадом.

Инструкције и директиве се обрађују на следећи начин:

У случају да не садрже ни један симбол него само константе, инструкција или директива се може одмах обрадити и увезати у листу свих инструкција које генеришу садржај у коду секција.

У случају да имамо симбол, такође увезујемо инструкцију или директиву у исту листу, али је обрађујемо тек после првог пролаза, након што сазнамо вредности свих симбола.

Током парсирања и обрађивања жетона, раде се стандардне провере и долази до грешке у случају неке недозвољене ситуације, попут дељења нулом, непостојеће инструкције, непосредног адрецирања одредишног операнда итд.

Потом долази до разрешавања симбола дефинисаних директивом еквиваленције, који се чувају у посебној табели неизрачунљивих симбола. Проласком кроз ту табелу, рачунамо индексе класификације свих израза, па на основу њих добијамо вредности симбола и убацујемо их у табелу симбола.

На крају, проласком кроз листу инструкција, генеришемо машински код и за инструкције и директиве које као операнде имају симболе, и генеришемо релокационе записе на местима где је то потребно, да би Повезивач, а касније и Пунилац, знали како да повежу и учитају програм у меморију.

Излаз асемблера представља објектни фајл, који уједно представља и улаз емулатора.

Покретање програма на Linux x86 архитектури:

Програм се покреће на Linux оперативном систему, прослеђивањем одговарајућих параметара у зависности од тога да ли се покреће емулатор или асемблер.

Асемблер се покреће на следећи начин:

* assembler -o output\_file.o input\_file.s

Емулатор се покреће на следећи начин:

* emulator -place<section\_name>@<start\_address> input\_file\_1.o ... input\_file\_n.o

При покретању емулатора, може се, али и не мора специфицирати стартна адреса секција у меморији. Подразумевано, IV табела се учитава у меморију на адресу 0x0000. После ње, налази се системски стек, а после њега се налази простор за учитавање корисничког програма, све до адресе 0xFF00, на којој се налазе меморијски мапирани регистри.

Извршни фајл се прави коришћењем команди make clean и make, а након тога се може покренути или на горе поменути начин, или покретањем скрипте која аутоматски покреће све тестове написане за сам програм. Покретање се врши на следећи начин, из главног директоријума:

* ./script.sh

Због разлике у формату текстуалног фајла на оперативном систему Linux и оперативном систему Windows, у програму Notepad++ потребно је урадити следеће са улазним фајловима оба програма:

* Edit => EOL Conversion => Unix(LF)

Покретање и тестови – Асемблер

Тест 1:

push 100

Излаз 1:

Error: instruction cannot be located inside .und or .abs section.

Тест 2:

.section .text:

div $0x0, a

.section .data:

.skip 4

a: .byte a

Излаз 2:

Error: dividing by zero.

Тест 3:

.section .text:

mov (%psw), $0xA

Излаз 3:

Error: destination operand cannot be immediately addressed ->

mov (%psw), $0xa instruction.

Тест 4:

.section .text:

.equ a, b

.equ b, c

.equ c, d

mov %r1, %r2

Излаз 4:

Error: undefined symbol(s) or circular dependency detected -> .EQU directive.

Тест 5:

.global \_start

.section .text:

\_start:

jmp \*(%r1)

call \*\_start(%pc)

.section .rodata:

.skip 4

jne \*\_start(%pc)

Излаз 5:

/ \*\*\* Symbol Table \*\*\* /

NAME SECTION SCOPE SECTION EXTERN ABSOLUTE EQU DEPEND. VALUE SIZE ENTRY

.und .und LOCAL true false false false N/A 0 0 0

.abs .abs LOCAL true false false false N/A 0 0 1

.text .text LOCAL true false false false N/A 0 0X6 2

.rodata .rodata LOCAL true false false false N/A 0 0X8 3

\_start .text GLOBAL false false false false N/A 0 0 4

#.text

2C 42 24 6E FA FF

#.ret .text

OFFSET TYPE VALUE

#.rodata

00 00 00 00 3C 6E FE FF

#.ret .rodata

OFFSET TYPE VALUE

0X6 R\_386\_PC16 4

Тест 6:

.extern i

.section .data:

.equ j, i + 0x1

.equ k, 13

.section .data:

mov $j, k(%r6)

Излаз 6:

/ \*\*\* Symbol Table \*\*\* /

NAME SECTION SCOPE SECTION EXTERN ABSOLUTE EQU DEPEND. VALUE SIZE ENTRY

.und .und LOCAL true false false false N/A 0 0 0

.abs .abs LOCAL true false false false N/A 0 0 1

.data .data LOCAL true false false false N/A 0 0X7 2

i .und GLOBAL false true false false N/A 0 0 3

j .und GLOBAL false true false true i 0X1 0 4

k .abs LOCAL false false true true N/A 0XD 0 5

#.data

64 00 01 00 6C 0D 00

#.ret .data

OFFSET TYPE VALUE

0X2 R\_386\_16 3

Тест 7:

.global \_start

.section .text:

print:

push %r3

push %r5

mov %r0, %r3

jeq return

mov %r3, %r5

mod $10, %r5

div $10, %r3

mov %r3, %r0

call print

add $zero, %r5

mov %r5, data\_out

return:

pop %r5

pop %r3

ret

\_start:

jmp \_start

halt

.section iv\_table:

.word ivt\_0

.word ivt\_1

.word ivt\_2

.word ivt\_3

.word ivt\_4

.word ivt\_5

.word ivt\_6

.word ivt\_7

.equ data\_in, 0xFF02

.equ data\_out, 0xFF00

.equ timer\_cfg, 0xFF10

.equ zero, 48 # ascii value for zero character

.equ space, 32 # ascii value for space character

.equ new\_line, 10 # ascii value for newline character

.section entries:

ivt\_0:

mov $1, timer\_cfg

mov $\_start, %pc

ivt\_1:

halt

ivt\_2:

cmp $0, %r4

jeq exit\_1

cmp $0, %r5

jeq stop

mov %r5, %r0

call print

mov $space, data\_out

sub $1, %r5

jmp \*exit\_1(%pc)

stop:

mov $0, %r4

mov $zero, data\_out

mov $new\_line, data\_out

exit\_1:

iret

ivt\_3:

cmp $new\_line, data\_in # check if enter key is pressed

jeq start

mul $10, %r5

mov data\_in, %r3

sub $zero, %r3

add %r3, %r5

jmp exit\_2

start:

mov $1, %r4

exit\_2:

mov $0, data\_in

iret

ivt\_4: iret

ivt\_5: iret

ivt\_6: iret

ivt\_7: iret

.end

Излаз 7:

/ \*\*\* Symbol Table \*\*\* /

NAME SECTION SCOPE SECTION EXTERN ABSOLUTE EQU DEPEND. VALUE SIZE ENTRY

.und .und LOCAL true false false false N/A 0 0 0

.abs .abs LOCAL true false false false N/A 0 0 1

.text .text LOCAL true false false false N/A 0 0X33 2

iv\_table iv\_table LOCAL true false false false N/A 0 0X10 3

entries entries LOCAL true false false false N/A 0 0X7C 4

\_start .text GLOBAL false false false false N/A 0X2E 0 5

print .text LOCAL false false false false N/A 0 0 6

return .text LOCAL false false false false N/A 0X29 0 7

zero .abs LOCAL false false true true N/A 0X30 0 8

data\_out .abs LOCAL false false true true N/A 0XFF00 0 9

ivt\_0 entries LOCAL false false false false N/A 0 0 10

ivt\_1 entries LOCAL false false false false N/A 0XC 0 11

ivt\_2 entries LOCAL false false false false N/A 0XD 0 12

ivt\_3 entries LOCAL false false false false N/A 0X4A 0 13

ivt\_4 entries LOCAL false false false false N/A 0X78 0 14

ivt\_5 entries LOCAL false false false false N/A 0X79 0 15

ivt\_6 entries LOCAL false false false false N/A 0X7A 0 16

ivt\_7 entries LOCAL false false false false N/A 0X7B 0 17

data\_in .abs LOCAL false false true true N/A 0XFF02 0 18

timer\_cfg .abs LOCAL false false true true N/A 0XFF10 0 19

space .abs LOCAL false false true true N/A 0X20 0 20

new\_line .abs LOCAL false false true true N/A 0XA 0 21

exit\_1 entries LOCAL false false false false N/A 0X49 0 22

stop entries LOCAL false false false false N/A 0X36 0 23

start entries LOCAL false false false false N/A 0X6B 0 24

exit\_2 entries LOCAL false false false false N/A 0X70 0 25

#.text

4C 26 4C 2A 64 20 26 34 00 29 00 64 26 2A CC 00 0A 00 2A 84 00 0A 00 26 64 26 20 24 00 00 00 6C 00 30 00

2A 64 2A 80 00 FF 54 2A 54 26 10 2C 00 00 00 00

#.ret .text

OFFSET TYPE VALUE

0X9 R\_386\_16 2

0X1D R\_386\_16 2

0X30 R\_386\_16 5

#iv\_table

00 00 0C 00 0D 00 4A 00 78 00 79 00 7A 00 7B 00

#.ret iv\_table

OFFSET TYPE VALUE

0 R\_386\_16 4

0X2 R\_386\_16 4

0X4 R\_386\_16 4

0X6 R\_386\_16 4

0X8 R\_386\_16 4

0XA R\_386\_16 4

0XC R\_386\_16 4

0XE R\_386\_16 4

#entries

64 00 01 00 80 10 FF 64 00 00 00 2E 00 8C 00 00 00 28 34 00 49 00 8C 00 00 00 2A 34 00 36 00 64 2A 20 24

00 00 00 64 00 20 00 80 00 FF 74 00 01 00 2A 2C 6E 13 00 64 00 00 00 28 64 00 30 00 80 00 FF 64 00 0A 00

80 00 FF 08 8C 00 0A 00 80 02 FF 34 00 6B 00 7C 00 0A 00 2A 64 80 02 FF 26 74 00 30 00 26 6C 26 2A 2C 00

70 00 64 00 01 00 28 64 00 00 00 80 02 FF 08 08 08 08 08

#.ret entries

OFFSET TYPE VALUE

0X9 R\_386\_16 5

0X14 R\_386\_16 4

0X1D R\_386\_16 4

0X24 R\_386\_16 2

0X53 R\_386\_16 4

0X69 R\_386\_16 4

Тест 8:

.global \_start

.section .text:

\_start:

jmp \_start

.section iv\_table:

.word ivt\_0

.word ivt\_1

.word ivt\_2

.word ivt\_3

.word ivt\_4

.word ivt\_5

.word ivt\_6

.word ivt\_7

.equ data\_in, 0xFF02

.equ data\_out, 0xFF00

.equ timer\_cfg, 0xFF10

.equ new\_line, 10 # ascii value for newline character

.section entries:

ivt\_0:

mov $\_start, %pc

ivt\_1:

halt

ivt\_2:

cmp $0, %r4

jeq exit\_1

cmp $0, %r0

jeq timer

cmp $0, %r0

jne inc

timer:

mov %r3, timer\_cfg

mov $1, %r0

jmp label\_t

inc:

add $1, timer\_cfg

label\_t:

mov $116, data\_out

mov $105, data\_out

mov $109, data\_out

mov $101, data\_out

mov $114, data\_out

mov $32, data\_out

iret

ivt\_3:

cmp $new\_line, data\_in # check if enter key is pressed

jeq start

jmp exit\_2

start:

mov $1, %r4

exit\_1:

iret

exit\_2:

mov $0, data\_in

iret

ivt\_4: iret

ivt\_5: iret

ivt\_6: iret

ivt\_7: iret

.end

Излаз 8:

/ \*\*\* Symbol Table \*\*\* /

NAME SECTION SCOPE SECTION EXTERN ABSOLUTE EQU DEPEND. VALUE SIZE ENTRY

.und .und LOCAL true false false false N/A 0 0 0

.abs .abs LOCAL true false false false N/A 0 0 1

.text .text LOCAL true false false false N/A 0 0X4 2

iv\_table iv\_table LOCAL true false false false N/A 0 0X10 3

entries entries LOCAL true false false false N/A 0 0X82 4

\_start .text GLOBAL false false false false N/A 0 0 5

ivt\_0 entries LOCAL false false false false N/A 0 0 6

ivt\_1 entries LOCAL false false false false N/A 0X5 0 7

ivt\_2 entries LOCAL false false false false N/A 0X6 0 8

ivt\_3 entries LOCAL false false false false N/A 0X61 0 9

ivt\_4 entries LOCAL false false false false N/A 0X7E 0 10

ivt\_5 entries LOCAL false false false false N/A 0X7F 0 11

ivt\_6 entries LOCAL false false false false N/A 0X80 0 12

ivt\_7 entries LOCAL false false false false N/A 0X81 0 13

data\_in .abs LOCAL false false true true N/A 0XFF02 0 14

data\_out .abs LOCAL false false true true N/A 0XFF00 0 15

timer\_cfg .abs LOCAL false false true true N/A 0XFF10 0 16

new\_line .abs LOCAL false false true true N/A 0XA 0 17

exit\_1 entries LOCAL false false false false N/A 0X75 0 18

timer entries LOCAL false false false false N/A 0X21 0 19

inc entries LOCAL false false false false N/A 0X2F 0 20

label\_t entries LOCAL false false false false N/A 0X36 0 21

start entries LOCAL false false false false N/A 0X70 0 22

exit\_2 entries LOCAL false false false false N/A 0X76 0 23

#.text

2C 00 00 00

#.ret .text

OFFSET TYPE VALUE

0X2 R\_386\_16 5

#iv\_table

00 00 05 00 06 00 61 00 7E 00 7F 00 80 00 81 00

#.ret iv\_table

OFFSET TYPE VALUE

0 R\_386\_16 4

0X2 R\_386\_16 4

0X4 R\_386\_16 4

0X6 R\_386\_16 4

0X8 R\_386\_16 4

0XA R\_386\_16 4

0XC R\_386\_16 4

0XE R\_386\_16 4

#entries

64 00 00 00 2E 00 8C 00 00 00 28 34 00 75 00 8C 00 00 00 20 34 00 21 00 8C 00 00 00 20 3C 00 2F 00 64 26

80 10 FF 64 00 01 00 20 2C 00 36 00 6C 00 01 00 80 10 FF 64 00 74 00 80 00 FF 64 00 69 00 80 00 FF 64 00

6D 00 80 00 FF 64 00 65 00 80 00 FF 64 00 72 00 80 00 FF 64 00 20 00 80 00 FF 08 8C 00 0A 00 80 02 FF 34

00 70 00 2C 00 76 00 64 00 01 00 28 08 64 00 00 00 80 02 FF 08 08 08 08 08

#.ret entries

OFFSET TYPE VALUE

0X2 R\_386\_16 5

0XD R\_386\_16 4

0X16 R\_386\_16 4

0X1F R\_386\_16 4

0X2D R\_386\_16 4

0X6A R\_386\_16 4

0X6E R\_386\_16 4

Тест 9:

.extern symbol

.section .text:

pushb %r1h

xchgb (%r2), symbol

Излаз 9:

/ \*\*\* Symbol Table \*\*\* /

NAME SECTION SCOPE SECTION EXTERN ABSOLUTE EQU DEPEND. VALUE SIZE ENTRY

.und .und LOCAL true false false false N/A 0 0 0

.abs .abs LOCAL true false false false N/A 0 0 1

.text .text LOCAL true false false false N/A 0 0X7 2

symbol .und GLOBAL false true false false N/A 0 0 3

#.text

48 23 58 44 80 00 00

#.ret .text

OFFSET TYPE VALUE

0X5 R\_386\_16 3

Тест 10:

.global \_start

.section .text:

print:

push %r3

push %r5

mov %r0, %r3

jeq return

mov %r3, %r5

mod $10, %r5

div $10, %r3

mov %r3, %r0

call print

add $zero, %r5

mov %r5, data\_out

return:

pop %r5

pop %r3

ret

\_start:

jmp \_start

halt

.section iv\_table:

.word ivt\_0

.word ivt\_1

.word ivt\_2

.word ivt\_3

.word ivt\_4

.word ivt\_5

.word ivt\_6

.word ivt\_7

.equ data\_in, 0xFF02

.equ data\_out, 0xFF00

.equ timer\_cfg, 0xFF10

.equ zero, 48 # ascii value for zero character

.equ space, 32 # ascii value for space character

.equ new\_line, 10 # ascii value for newline character

.section entries:

ivt\_0:

mov $1, timer\_cfg

mov $\_start, %pc

ivt\_1:

halt

ivt\_2:

iret

ivt\_3:

cmp $new\_line, data\_in # check if enter key is pressed

jeq start

mul $10, %r5

mov data\_in, %r3

sub $zero, %r3

add %r3, %r5

jmp exit\_2

start:

mov %r5, %r4

loop:

cmp $0, %r5

jeq calc

sub $0x1, %r5

cmp $0x0, %r5

jeq calc

mul %r5, %r4

jmp loop

calc:

mov %r4, %r0

cmp $0, %r4

jeq exit\_3

call print

help:

mov $0, %r5

mov $new\_line, data\_out

exit\_2:

mov $0, data\_in

iret

exit\_3:

mov $zero, data\_out

jmp help

ivt\_4: iret

ivt\_5: iret

ivt\_6: iret

ivt\_7: iret

.end

Излаз 10:

/ \*\*\* Symbol Table \*\*\* /

NAME SECTION SCOPE SECTION EXTERN ABSOLUTE EQU DEPEND. VALUE SIZE ENTRY

.und .und LOCAL true false false false N/A 0 0 0

.abs .abs LOCAL true false false false N/A 0 0 1

.text .text LOCAL true false false false N/A 0 0X33 2

iv\_table iv\_table LOCAL true false false false N/A 0 0X10 3

entries entries LOCAL true false false false N/A 0 0X83 4

\_start .text GLOBAL false false false false N/A 0X2E 0 5

print .text LOCAL false false false false N/A 0 0 6

return .text LOCAL false false false false N/A 0X29 0 7

zero .abs LOCAL false false true true N/A 0X30 0 8

data\_out .abs LOCAL false false true true N/A 0XFF00 0 9

ivt\_0 entries LOCAL false false false false N/A 0 0 10

ivt\_1 entries LOCAL false false false false N/A 0XC 0 11

ivt\_2 entries LOCAL false false false false N/A 0XD 0 12

ivt\_3 entries LOCAL false false false false N/A 0XE 0 13

ivt\_4 entries LOCAL false false false false N/A 0X7F 0 14

ivt\_5 entries LOCAL false false false false N/A 0X80 0 15

ivt\_6 entries LOCAL false false false false N/A 0X81 0 16

ivt\_7 entries LOCAL false false false false N/A 0X82 0 17

data\_in .abs LOCAL false false true true N/A 0XFF02 0 18

timer\_cfg .abs LOCAL false false true true N/A 0XFF10 0 19

space .abs LOCAL false false true true N/A 0X20 0 20

new\_line .abs LOCAL false false true true N/A 0XA 0 21

start entries LOCAL false false false false N/A 0X2F 0 22

exit\_2 entries LOCAL false false false false N/A 0X6C 0 23

loop entries LOCAL false false false false N/A 0X32 0 24

calc entries LOCAL false false false false N/A 0X50 0 25

exit\_3 entries LOCAL false false false false N/A 0X74 0 26

help entries LOCAL false false false false N/A 0X60 0 27

#.text

4C 26 4C 2A 64 20 26 34 00 29 00 64 26 2A CC 00 0A 00 2A 84 00 0A 00 26 64 26 20 24 00 00 00 6C 00 30 00

2A 64 2A 80 00 FF 54 2A 54 26 10 2C 00 00 00 00

#.ret .text

OFFSET TYPE VALUE

0X9 R\_386\_16 2

0X1D R\_386\_16 2

0X30 R\_386\_16 5

#iv\_table

00 00 0C 00 0D 00 0E 00 7F 00 80 00 81 00 82 00

#.ret iv\_table

OFFSET TYPE VALUE

0 R\_386\_16 4

0X2 R\_386\_16 4

0X4 R\_386\_16 4

0X6 R\_386\_16 4

0X8 R\_386\_16 4

0XA R\_386\_16 4

0XC R\_386\_16 4

0XE R\_386\_16 4

#entries

64 00 01 00 80 10 FF 64 00 00 00 2E 00 08 8C 00 0A 00 80 02 FF 34 00 2F 00 7C 00 0A 00 2A 64 80 02 FF 26

74 00 30 00 26 6C 26 2A 2C 00 6C 00 64 2A 28 8C 00 00 00 2A 34 00 50 00 74 00 01 00 2A 8C 00 00 00 2A 34

00 50 00 7C 2A 28 2C 00 32 00 64 28 20 8C 00 00 00 28 34 00 74 00 24 00 00 00 64 00 00 00 2A 64 00 0A 00

80 00 FF 64 00 00 00 80 02 FF 08 64 00 30 00 80 00 FF 2C 00 60 00 08 08 08 08

#.ret entries

OFFSET TYPE VALUE

0X9 R\_386\_16 5

0X17 R\_386\_16 4

0X2D R\_386\_16 4

0X39 R\_386\_16 4

0X47 R\_386\_16 4

0X4E R\_386\_16 4

0X5A R\_386\_16 4

0X5E R\_386\_16 2

0X7D R\_386\_16 4

Тест 11:

.global \_start

.section .text:

\_start:

jmp \_start

.section iv\_table:

.word ivt\_0

.word ivt\_1

.word ivt\_2

.word ivt\_3

.word ivt\_4

.word ivt\_5

.word ivt\_6

.word ivt\_7

.equ data\_in, 0xFF02

.equ data\_out, 0xFF00

.section entries:

ivt\_0:

mov $\_start, %pc

ivt\_1:

halt

ivt\_2:

iret

ivt\_3:

mov data\_in, data\_out

mov $0, data\_in

iret

ivt\_4: iret

ivt\_5: iret

ivt\_6: iret

ivt\_7: iret

.end

Излаз 11:

/ \*\*\* Symbol Table \*\*\* /

NAME SECTION SCOPE SECTION EXTERN ABSOLUTE EQU DEPEND. VALUE SIZE ENTRY

.und .und LOCAL true false false false N/A 0 0 0

.abs .abs LOCAL true false false false N/A 0 0 1

.text .text LOCAL true false false false N/A 0 0X4 2

iv\_table iv\_table LOCAL true false false false N/A 0 0X10 3

entries entries LOCAL true false false false N/A 0 0X1A 4

\_start .text GLOBAL false false false false N/A 0 0 5

ivt\_0 entries LOCAL false false false false N/A 0 0 6

ivt\_1 entries LOCAL false false false false N/A 0X5 0 7

ivt\_2 entries LOCAL false false false false N/A 0X6 0 8

ivt\_3 entries LOCAL false false false false N/A 0X7 0 9

ivt\_4 entries LOCAL false false false false N/A 0X16 0 10

ivt\_5 entries LOCAL false false false false N/A 0X17 0 11

ivt\_6 entries LOCAL false false false false N/A 0X18 0 12

ivt\_7 entries LOCAL false false false false N/A 0X19 0 13

data\_in .abs LOCAL false false true true N/A 0XFF02 0 14

data\_out .abs LOCAL false false true true N/A 0XFF00 0 15

#.text

2C 00 00 00

#.ret .text

OFFSET TYPE VALUE

0X2 R\_386\_16 5

#iv\_table

00 00 05 00 06 00 07 00 16 00 17 00 18 00 19 00

#.ret iv\_table

OFFSET TYPE VALUE

0 R\_386\_16 4

0X2 R\_386\_16 4

0X4 R\_386\_16 4

0X6 R\_386\_16 4

0X8 R\_386\_16 4

0XA R\_386\_16 4

0XC R\_386\_16 4

0XE R\_386\_16 4

#entries

64 00 00 00 2E 00 08 64 80 02 FF 80 00 FF 64 00 00 00 80 02 FF 08 08 08 08 08

#.ret entries

OFFSET TYPE VALUE

0X2 R\_386\_16 5