

CS2308

Directives

09/27/16

.ascii

Prompt: .ascii "please > IN"
LABEL

in compile time compiler asks
for space for storing the string
compiler TELLS
prompt " string "

Load & Link the program

Place the string in the DATA

Segment Associates prompt with
the base address of the string

2

A : .WORD 100

int A = 100;

B : .space 4

int B;

Arr : .word 5, 8, 13, 21, 34

int Arr = { 5, 8, 13, 21, 34 };

Bnr : .space 20

int B[20]

→ place 50 in B

→ place A in \$to

Ex \$R, Loke (LA \$5, B)

Base address into \$R

3)

Li \$t0, 50

La \$t1, B \$t1 base of B

Sw \$t0, 0(\$t1)

La \$t1, A

Lw \$t0, 0(\$t1)

Arr .space 80

Arr[5] = 55

Lo \$t0, Arr

Move \$50,\$t0

Li \$t1, 55

addi \$50,\$50,20

Sw \$t1,0(\$t0)

La \$t0, Arr

to points to A.
Arr

Li \$t1, 55

Sw \$t1, 20(\$t0) static

C52318
09/29/16

Switch

word

Jumptable: 1000, 2000, 3000

JR \$6

La \$5, Jumptu

JR \$7
JR \$9 ↗

in 4th segment say after 5000

1000
2000
3000

5000 ← Jumptable

La \$5, Jumptable

\$5 ← 5000

LW \$6, 0(\$5) \$6 ← 1000

LW \$7, 4(\$5) \$7 ← 2000

add \$8, \$5, 8

LW \$8, 0(\$8) -300

2

Jump table .word case1, case2, default

-

-

-

-

-

case1 vs case 2 vs
case

case1:

-

- 500

-

-

case2:

-

520

-

-

20

-

fehlt:

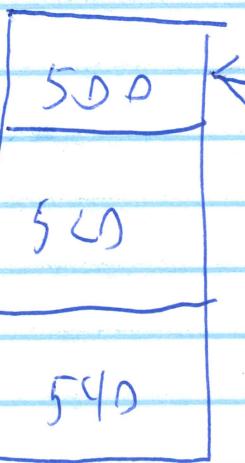
-

540

-

Jump table

0



500 <=> jump table

3

.data

X JT: .word cos1, cos2, cos3

.text

LA \$\$P, JT

base +0
+4
+8

501	JT
5L9	
540	

prompt Enter an int

0, 1, 2 Read int int \$t0

5scale

Righton \$t0 to be returned

D - 2

Check multiply \$t0 by 4

(byte offset) place in \$t1

→ add \$t1 to \$\$P place in \$t2
lw \$t2, 0(\$t2)
jr \$t2

4

Function calls

Using the convention and
or the stack

Convention \$a_{0..a₃} are arguments
O.W. use the stack (need more)

\$V₀, \$V₁ return value

O.W. use the stack

\$T₀, ... \$T₉ temporary reg.

Do not maintain value between function
call

\$S₀ - \$S₈ temporaries that do

Maintain values across functions

\$RA is the top level Return Address
More nesting? \Rightarrow stack

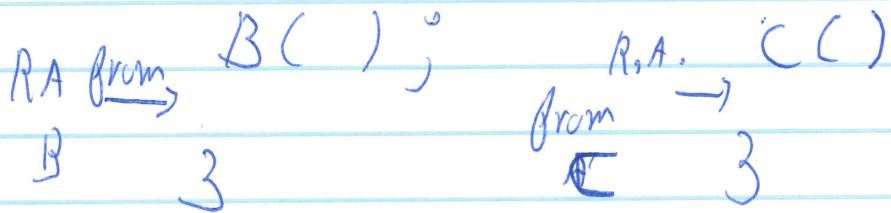
5

int < >;

void A, B, C

A() ;

B() ;



ORDER is LIFO \Rightarrow stack

Often only the stack is used

TOP A() ; Local variables on the stack



\leftarrow push R.A.; set arguments

B() JAL B

\leftarrow pop R.A.; get local variables



6

A:

use \$s

args into A

push \$RA

push needed \$ts

JAL B



POP RA

~~Arguments to A~~

use Vs as needed

- ~~JR \$RA~~ Done with B
C can use ts with no regard
A.

JR \$RA

C:

Assume C requires \$s

C push needed \$s before use
pop before return

JR \$RA

7 RA is pushed when the function starts

Poped when a Caller returns

The caller is responsible for
Save needed st

Caller is responsible for restore

Any needed ss

Argument, return value

are on V's ds or stack

Local

Temporary vars one on stack

Global vars global data segment