Copyright © 2021 Anthony J. Dos Reis

**5** Global, Dynamic Local, and Static Local Variables

Global Variables

1 ; ex0501.a Global variables

2

3 startup: bl main

4 halt

5 ;===========================================================

6 ; #include <stdio.h>

7 x: .word 2 ; int x = 2, y;

Default value of global variables is 0

8 y: .word 0

9

10 main: push lr ; int main()

11 push fp ; {

12 mov fp, sp

13

14 ld r0, x ; y = x;

15 st r0, y

16

17 ld r0, y ; printf("%d\n", y);

18 dout r0

29 nl

20

21 mov r0, 0 ; return 0;

22 mov sp, fp

23 pop fp

24 pop lr

25 ret

26 ; }

*Rule*: C global variables for which an initial value is not specified get the default initial value 0.

*Rule*: Global variables are accessed with the ld and st instructions.

Dynamic Local Variables

1 ; ex0502.a Dynamic local variables

2

3 startup: bl main

4 halt

5 ;===========================================================

6 ; #include <stdio.h>

7 f: push lr ; void f(int a)

8 push fp ; {

9 mov fp, sp

Creates and initializes x

10

11 mov r0, 1 ; int x = 1, y;

12 push r0

Creates y

13 sub sp, sp, 1

14

15 ldr r0, fp, -1 ; y = x + a;

16 ldr r1, fp, 2

17 add r0, r0, r1

18 str r0, fp, -2

19

20 ldr r0, fp, -2 ; printf("%d\n", y);

21 dout r0

22 nl

23

24 mov sp, fp ; }

25 pop fp

26 pop lr

27 ret

28 ;===========================================================

29 main: push lr ; int main()

30 push fp ; {

31 mov fp, sp

32

33 mov r0, 11 ; f(11);

34 push r0

35 bl f

36 add sp, sp, 1

37

38 mov r0, 0 ; return 0;

39 mov sp, fp

40 pop fp

41 pop lr

42 ret

43 ; }

sp (r6) ~ ~

garbage y

local variables

fp (r5) 1 x

saved fp

saved lr

11 a parameter

~ ~

*Rule*: Access parameters with positive offsets; access local variables with negative offsets.

24 mov sp, fp

sp (r6) ~ ~

12 y

local variables now off the stack

fp (r5) 1 x

saved fp saved fp now is top of the stack

saved lr

11 a parameter

~ ~

25 pop fp

26 pop lr

27 ret

*Rule*: A function creates its dynamic local variables on entry and destroys them on exit. The values of dynamic local variables are not retained between calls of a function.

Local variable (no default initialization):

int y;

Code is not

mov r0, 0 ; create and initialize y  
 push r0

In, it is

sub sp, sp, 1 ; reserve slot on stack for y

Static Local Variables

1 // ex0503.c

2 #include <stdio.h>

3 int x = 5;

4 void f()

5 {

6 static int x;

7 printf("%d\n", x); // displays 0

8 }

9 //===================

10 void g()

11 {

12 printf("%d\n", x); // displays 5

13 }

14 //===================

15 int main()

16 {

17 static int x = 3;

18 printf("%d\n", x); // displays 3

19 f();

20 g();

21 return 0;

22 }

static Local Variables Created with .word Directives

Incorrect!!!

x: .word 5 ; global x

x: .word 0 ; x in f

x: .word 3 ; x in main

Correct!!!

1. “@s” (“@” indicates compiler generated label )
2. a sequence number (0, 1, 2, …)
3. an underscore
4. the name of the variable in the C program

@s0\_x: .word 0 ; x in f

@s1\_x: .word 3 ; x in main

ld r0, @s0\_x ; load the x in f  
 dout r0  
 nl

The .word directive for the global x, on the other hand, is labeled with x:

x: .word 5 ; global x

1 ; ex0503.a Static local variables

2

3 startup: bl main

4 halt ; back to operating system

5 ;===========================================================

6 ; #include <stdio.h>

7 x: .word 5 ; int x = 5;

No code for the declaration

of a static local variable

8 f: push lr ; void f()

9 push fp ; {

10 mov fp, sp

11

12 ; static int x;

13

14 ld r0, @s0\_x ; printf("%d\n", x);

15 dout r0

16 nl

17

18 mov sp, fp ; }

19 pop fp

20 pop lr

21 ret

22 ;===========================================================

23 g: push lr ; void g()

24 push fp ; {

25 mov fp, sp

26

27 ld r0, x ; printf("d\n", x);

28 dout r0

Accessing global variable using C-level name

29 nl

30

31 mov sp, fp ; }

32 pop fp

33 pop lr

34 ret

35 ;===========================================================

36 main: push lr ; int main()

No code for the declaration

of a static local variable

37 push fp ; {

38 mov fp, sp

39

40 ; static int x = 3;

41

42 ld r0, @s1\_x ; printf("%d\n", x);

43 dout r0

44 nl

45

46 bl f ; f();

47

48 bl g ; g();

49

50 mov r0, 0 ; return 0;

51 mov sp, fp

52 pop fp

53 pop lr

54 ret

Static local variables

created with .word

directives. Labels start with unique prefix.

55 ; }

56 @s0\_x: .word 0

57 @s1\_x: .word 3

ex0503.a

o

C marks the beginning of the machine code (no header for this program) from the machine code

C

Startup

code

5 x (global x)

f

g

main

0 @s0\_x (static local x in f)

3 @s1\_x (static local x in main)