

Mixing C and Assembly

Defining in assembly and using in C

The assembly code which defines functions and variables:

```
# define a global variable asm_var and export it so that C code can use it.
.data
.global asm_var
asm_var:
.long

# define a function asm_fun and export it so that C code can use it.
.text
.global asm_fun
asm_fun:
    ret;
```

C code:

```
void asm_fun();
extern int asm_var;

int x;
void main()
{
    asm_fun();
    asm_var++;
}
```

Points to understand:

- Assembly code can define variables and functions which can be used in C program.
- The assembly code must export variables and functions which will be used from C code. In this example .global is used to export asm_var and asm_fun.
- The C code should declare variables and function before using. These declarations are similar to what is done in the header file. The declaration can be done in c source files or c header files.

Defining in C and using in assembly

C code

```
void fun()
{
}
```

Assembly code:

```
.text
.global main
main:
    call fun
    ret;
```

Points to understand:

- Functions and variables defined in C are always exported unless you use the static keyword. You do not have to export each symbol as you do in assembly.
- The assembly can call function or use variables as if they are defined in assembly code itself.

Mixing assembly inside C code

There are few things that you can not do in pure C code at all. For those things you have to use assembly. For example if you want to raise a hardware exception, then you need to write assembly instructions. C does not provide any syntax to raise hardware exceptions. To avoid writing code in separate assembly files and call from C code, C provides a way to insert assembly code inside a C code. This is done by asm keyword.

Here is an example:

```
int main()
{
    asm("int $3");
    return 0;
}
```

Assembly which is generated from this C code will be:

```
.text
.globl main
.type    main, @function
main:
    pushl %ebp
    movl  %esp, %ebp
    int $3
    movl  $0, %eax
    popl  %ebp
    ret
```

◄ [Memory Allocation for Structure \(/cin/structure.html\)](/cin/structure.html)

[up \(/cin/cin.html\)](/cin/cin.html)

[Miscellaneous > \(/cin/misc.html\)](/cin/misc.html)

Do you collaborate using whiteboard? Please try Lekh Board - An Intelligent Collaborate Whiteboard App (<https://lekh.app>)

Copyright © 2012 www.avabodh.com. All rights reserved.