

DESIGNER'S NOTEBOOK



Sharing Header Files in C and Assembly

Contributed by Alan Davis

Design Problem

Sometimes it is useful to be able to define named constants that can be used in both C and assembly language.

One method is to have separate header files that define the same symbols: a C include file with `#define` directives, and an assembler include file with `.set` or `.asg` directives. But can you have a single, shared header file that defines the symbols once for both C and assembler?

Solution

The file shown in Figure 1 can be used normally as a C include file (ASMDEFS not defined). It can also be used to generate an assembler include file: compile with ASMDEFS defined and use `-k` to keep the output:

```
c130 -dASMDEFS -k defs.h
```

```
#define PI 3.14
#define E 2.72
#ifdef ASMDEFS /* IF DEFINED, CREATE .asg DIRECTIVES */
#define ASM_ASG(sym) asm("\t.asg\t" VAL(sym) ", " #sym)
#define VAL(sym)      #sym
ASM_ASG(PI);
ASM_ASG(E);
#endif /*ASMDEFS*/
```

Figure 1. Input file `defs.h`

The output is the file `defs.asm`, which contains `.asg` directives for your symbols. See Figure 2.

```
; ... <compiler-generated header stuff> ...
.asg 3.14,PI
.asg 2.72,E
```

Figure 2. Output file `defs.asm`

You can then `.include` this file in your assembly modules. The same technique can be used to create `.set` directives rather than `.asg`.

Here's how it works: The `ASM_ASG` macros in `defs.h` expand to `asm` statements containing the `.asg` directives. The trick is in generating both the name and the value of the argument symbol. `ASM_ASG` accomplishes this with ANSI C's new stringize operator, `#`. The last expression in `ASM_ASG`'s definition, `#sym`, simply makes a string out of the argument without expanding it. Thus, `#PI` becomes `"PI"`. The second expression in `ASM_ASG`'s definition calls another macro, `VAL`, which, in turn, stringizes its argument. But in passing `sym` to `VAL`, `PI` is expanded (to `3.14`), so `VAL` returns `"3.14"`. The result:

```
asm("\t.asg\t" "3.14" ", " "PI")
```

In ANSI C, adjacent strings are concatenated, so this compiles down to a simple `.asg` directive in `defs.asm`.