Examples of Pseudo-Operations B

This chapter shows some examples of ways to use various pseudo-ops.

Example 1

This example shows how to use the following pseudo-ops to specify the bindings of variables in C:

`common, .global, .local, .weak`

The following C definitions/declarations:

```
int     foo1 = 1;
#pragma weak foo2 = foo1
static  int foo3;
static  int foo4 = 2;
```

can be translated into the following assembly code:

Code Example B-1

```
    .pushsection".data"

    .globalfoo1! int foo1 = 1
    .align4
foo1:
    .word0x1
    .type   foo1,#object   ! foo1 is of type data object,
    .size   foo1,4         ! with size = 4 bytes

    .weak       foo2       ! #pragma weak foo2 = foo1
    foo2 = foo1

    .local      foo3       ! static int foo3
    .common     foo3,4,4

    .align      4          ! static int foo4 = 2
   foo4:
    .word       0x2
    .type       foo4,#object
    .size       foo4,4

    .popsection
```

Example 2

This example shows how to use the pseudo-op `.ident` to generate a string in the `.comment` section of the object file for identification purposes.

```
.ident  "acomp: (CDS) SPARCompilers 2.0 alpha4 12 Aug 1991"
```

Example 3

The pseudo-ops shown in this example are `.align`, `.global`, `.type`, and `.size`.

The following C subroutine:

```
int sum(a, b)
    int a, b;
{
    return(a + b);
}
```

can be translated into the following assembly code:

```
    .section    ".text"


    .global     sum

    .align      4

sum:

    retl
    add     %o0,%o1,%o0    ! (a + b) is done in the
                           ! delay slot of retl

    .type   sum,#function  ! sum is of type function
    .size   sum,.-sum      ! size of sum is the diff
                           ! of current location
                           ! counter and the initial
                           ! definition of sum
```

Example 4

The pseudo-ops shown in this example are `.section`, `.ascii`, and `.align`. The example calls the `printf` function to output the string `"hello world"`.

```
    .section   ".data1"
    .align  4
.L16:
    .ascii   "hello world\n\0"

    .section   ".text"
    .global    main
main:
    save   %sp,-96,%sp
    set    .L16,%o0
    call   printf,1
    nop
    restore
```

Example 5

This example shows how to use the `.volatile` and `.nonvolatile` pseudo-ops to protect a section of handwritten asembly code from peephole optimization.

```
.volatile
t       0x24
std     %g2, [%o0]
retl
nop
.nonvolatile
```