SPARC Demo Programs

```
! myHello.s
!
! Harry Porter - 12/23/97
!
! This program demonstrates the basics of assembling and running a SPARC
! assembly language program.  It prints out a message when run.
!
        .data                           ! Data will go into 'data' segment
arg:    .asciz  "Hello world!!!\n"       ! A string argument to printf

        .text                           ! Code will go into 'text' segment
        .global main                    ! Make this symbol externally known

main:   save    %sp,-(64+4+24)&-8,%sp   ! Set up a new activation record
        set     arg,%o0                 ! Move arg1 (ptr to string) into %o0
        call    printf                  ! Call printf
        nop                             ! Delay slot is wasted
        ret                             ! Return to caller after
        restore                         ! .  restoring the stack
```

To assemble and run this program, type:

```
gcc —g -c myHello.s
gcc -g myHello.o -o myHello
myHello
```

To see what the "C" compiler produces, type this:

```
gcc -S goodbye.c
more goodbye.s
```

SPARC Demo Programs

```
! myFloat.s
!
! Harry Porter - 1/13/00
!
! This program demonstrates floating point computation.  It also
! demonstrates the preferred style of commenting.
!
! This program performs this function:
!
!        main () {
!           double x = 12.34;
!           double y = 10;
!           printf ("Result = %g\n", x + y);
!        }
!
! Frame layout:
!    %fp-24  ==> y    (a floating double)
!    %fp-16  ==> x    (a floating double)
!    %fp-8   ==> temp (a floating double)
! (Size of local storage = 3*8 = 24 bytes)
!
         .text

strArg: .asciz  "Result = %g\n"          ! String constant

         .align 8
xConst: .double 0r+1234.0e-2             ! Double constant = 12.34
yConst: .double 0r+1.00E1                ! Double constant = 10

         .global main                     ! Begin main function
main:   save %sp,-(64+4+24+0+24)&-8,%sp ! .

         sethi %hi(xConst),%o0            ! Initialize x (%fp-16)
         ldd [%o0+%lo(xConst)],%o2        ! .   from stored constant xConst
         std %o2,[%fp-16]                 ! .

         sethi %hi(yConst),%o0            ! Initialize y (%fp-24)
         ldd [%o0+%lo(yConst)],%o2        ! .   from stored constant yConst
         std %o2,[%fp-24]                 ! .

         ldd [%fp-16],%f2                 ! Add x+y, storing result in
         ldd [%fp-24],%f4                 ! .   temp var (%fp-8)
         faddd %f2,%f4,%f6                ! .
         std %f6,[%fp-8]                  ! .

         ldd [%fp-8],%o2                  ! Load value of temp
         mov %o2,%o1                      ! .   into %o1 and %o2
         mov %o3,%o2                      ! .
         set strArg,%o0                   ! Load addr of strArg into %o0
         call printf                      ! Call printf
         nop                              ! .

         ret                              ! Return from "main"
         restore                          ! .
```