Princeton University
COS 217: Introduction to Programming Systems
SPARC Subroutine Calling Conventions

```
When subroutine f calls subroutine g...

In f:

   (1) Store actual parameters 1 to 6 in registers %o0-%05.

   (2) Store actual parameters 7 and above in memory locations %sp+92, %sp+96, ...

   (3) Execute "call g".

         Store %pc in register %o7.
            Register %o7 thus stores the address of the call instruction.
         Jump to the instruction at label g.

   (4) But before executing the "jumped to" instruction, execute the delay instruction
       that follows the "call" instruction.

In g:

   (5) Execute "save %sp, -N, %sp".

         Compute the sum of -N and the current value of %sp (%o6).
         Slide the register window forward.
            Create a fresh set of %l0-%l7 and %o0-%o7 registers.
            The former %o0-%o7 registers are now known as the %i0-%i7 registers.
               Note:  The former %o6 register now known as %i6.
                  That is, the former %sp register is now known as %fp.
                  Thus %fp marks the previous top of the stack.
            [Note: Saves oldest register window in stack if necessary.]
         Store the sum (computed above) in %sp.
            Thus a new stack frame is pushed onto the stack.

   (6) Use the %i0-%i5 and %fp+92, %fp+96, ... to compute return value(s).

   (7) Store return values in registers %i0-%i5.

   (8) Execute "ret".

         Jump to %i7 + 8
            Jump to the instruction after the delay instruction after the call
            instruction.

   (9) But before executing the "jumped to" instruction, execute the delay instruction
       that follows the "ret" instruction, that is, "restore".

         Slide the register window backward.
            The former %i0-%i7 registers are now known as %o0-%o7.
               Note:  The former %i6 register is now known as %o6.
                  That is, the former %fp register is now known as %sp.
                  Thus a stack frame is popped from the stack.
            Restore the old set of registers %l0-%l7 and %i0-%i7.
            [Note: Restores current register window from stack if necessary.]

In f:

   (10) Retrieve g's return values from registers %o0-%o5.
```