

Princeton University

COS 217: Introduction to Programming Systems

SPARC “Leaf” Subroutine Calling Conventions

A subroutine can be a leaf subroutine only if it need not have its own stack frame.
That is, only if it calls no other subroutines and defines no local variables in memory.

A leaf subroutine may use only registers %o1-%o5 and %g0-%g1.

The calling subroutine need not know that the called subroutine is a leaf subroutine.

When subroutine *f* calls leaf subroutine *g*...

In *f*:

- (1) Store actual parameters 1 to 6 in registers %o0-%o5.
- (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) Use the %o0-%o5 and %sp+92, %sp+96, ... to compute return value(s).
- (6) Store return values in registers %o0-%o5.
- (7) Execute “retl” (return from leaf).

 Jump to %o7 + 8
 Jump to the instruction after the delay instruction after the call instruction.
- (8) But before executing the “jumped to” instruction, execute the delay instruction that follows the “retl” instruction, typically “nop”.

In *f*:

- (9) Retrieve *g*'s return values from the registers %o0-%o5.