

Calling Conventions

64-bit Conventions

Passing Arguments

User level functions are passed arguments through the following registers in order:

- `%rdi, %rsi, %rdx, %rcx, %r8, %r9`

System calls are passed arguments through:

- `%rdi, %rsi, %rdx, %r10, %r8, %r9`

Floating point arguments are passed through `%xmm0-%xmm7`. The system call number is passed through `%rax`.

Callee/Caller Saved Registers

Preserved:

- `%rbp, %rbx, %rsp, %r12-15`

Volatile:

- `%rax, %rcx, %rdx, %rdi, %rsi, %r8-11`

Floating point registers are all caller saved.

Return Values

Integers return values are returned through `%rax`, floating point return values are returned through `%xmm0`.

32-bit Conventions

Passing Arguments

User level functions are passed arguments through the stack. Arguments are pushed onto the stack in the order of last to first.

System calls are passed arguments through:

- `%ebx, %ecx, %edx, %esi, %edi, %ebp`

For system calls with more than six arguments `%ebx` must contain the memory location of the arguments. Floating point arguments are passed through `%st0-%st7`. The system call number is passed through `%eax`.

Callee/Caller Saved Registers

Preserved:

- `%ebp, %ebx, %edi, %esi, %esp`

Volatile:

- `%eax, %ecx, %edx`

Floating point registers are all caller saved.

Return Values

Integers return values are returned through `%eax`, floating point return values are returned through `%st0`.
