Basics & Sample Programs

Here are some sample programs. System calls are made using the syscall instruction on an x86-64 version of GNU/Linux as opposed to using int 0x80 on an x86 version of GNU/Linux. All the programs are in long mode. The list of system calls can be found in /usr/include/asm-x86_64/unistd.h.

Calling Convention

System Calls

1. The kernel or system call interface uses registers RDI, RSI, RDX, R10, R8, R9 for passing arguments in that order. A maximum of 6 parameters can be passed.
2. The kernel destroys registers RCX and R11.
3. The number of the system call is passed in the register RAX.
4. No argument is passed on the stack.
5. The return value is placed in RAX. A value in the range -1 to -4095 (0xFFFFFFFF to 0xFFFF0000).
6. In 32-bit mode, GNU/Linux supports 6 arguments in the system call and they are passed in the registers EBX, ECX, EDX, ESI, EDI and EBP with the system call number in EAX.

Function Calls

1. For a complete list of the registers that should be used for passing parameters, and for return values, refer the x86-64 ABI .
2. The integer and pointer arguments are passed in the registers RDI, RSI, RDX, RCX, R8, R9 in that order.
3. The registers XMM0-XMM7 are used to pass the single and double precision floating point arguments. Rest of the arguments might have to be passed on the stack
4. The RAX register should hold the number of SSE registers (XMM0-XMM7) that are used in the passing of arguments.
5. The registers RBX, RSP, RBP, R12-R15 are callee-saved registers and are preserved across function calls.
6. RBX is the optional base pointer and RBP is the optional frame pointer.
7. R11 is the temporary register used by the Procedure Linkage Table and R10 is used to pass a function's static chain pointer.
8. Integer or pointer type return values are returned in RAX and RDX.
9. Floating point return values are returned in XMM0 and XMM1. Long double precision values are returned in ST(0) and ST(1).