

Basic Instructions

Mnemonic	Purpose
mov dest,src	Move data between registers, load immediate data into registers, move data between registers and memory.
push src	Insert a value onto the stack. Useful for passing arguments, saving registers, etc
pop dest	Remove the top most value from the stack. Equivalent to "mov dest, [rsp]; add 8, rsp"
call func	Push the address of the next instruction and start executing "func"
ret	Pop the return program counter and jump there. Ends subroutine
add dest, src	$\text{dest} = \text{dest} + \text{src}$
sub dest, src	$\text{dest} = \text{dest} - \text{src}$
mul src	Multiply rax and src as unsigned integers, and put the result in rax.
div src	Divide rax by src and put the ration into rax, and the remainder into rdx. On operation rdx must be zero or you will get a SIGFPE
shr val, bits	Bitshift a value to the right by a constant, or the low 8 bits of rcx ("cl")
jmp label	Go to the instruction label. Skips anything else in the way.
cmp a,b	Compare two values. Sets flags that are used by the conditional jumps
j<label>	Go to label if previous comparison came out as less-than. Other conditionals are available: jle (<=), je (==), jge(>=), jg (>), jne (!=). There are also unsigned comparisons: jb (<), jbe (<=), ja (>), jae (>=)
and dest, src	$\text{dest} = \text{dest} \& \text{src}$
or dest, src	$\text{dest} = \text{dest} \text{src}$
xor dest, src	$\text{dest} = \text{dest} \wedge \text{src}$
inc src	$\text{src} = \text{src} + 1$
dec src	$\text{src} = \text{src} - 1$
syscall n	Invoke operating system routine n. Same effect as int n and is available for 64-bit ASM. NOTE: syscall will overwrite the RCX register!! It is also the recommended way to make system calls in 64-bit ASM.
int val	Interrupt. Used to perform a system call. Used in 32-bit ASM.
loop <label>	Used to create looping code. Uses RCX to keep track of the number of loops. On each loop RCX is decremented and if it's 0 the loop ends, otherwise it jumps to specified label.