- addresses refer to bytes so two hex numbers == one incremented address
  - each hex number == 4 bits
  - 2 = 1 byte
- main function allocates hidden local variable on stack for return value
- negative values from base pointer address local variables within the current stack frame
- positive values from the base pointer are formal arguments pushed to stack before function call

Label Issue

```
movq label(%rip), register -> loads the value of the label in
leaq label(%rip), register -> loads the address of the label in
```

- remember segment/section .data == .section __DATA,__data, etc.
- syscall is default way of entering kernel mode on x86-64. This instruction is not available in 32 bit modes of operation on Intel processors
- sysenter is an instruction most frequently used to invoke system calls in 32 bit modes of operation
  - is similar to syscall, a bit more difficult to use though, but that is kernel's concern
- int 0x80 is a legacy way to invoke a system call and should be avoided
- preferable way to invoke a system call is to use VDSO
  - VDSO
    - a part of memory mapped in each process address space that allows for use of system calls more efficiently
      - e.g. by not entering kernel mode in some cases at all
    - also takes care of more difficult handling of syscall or sysenter instructions (compared to the legacy int 0x80 way)
- resd -> reserve double in bss