<http://blog.csdn.net/21cnbao/article/details/7385161>

(gdb) disassemble /rm

Dump of assembler code for function main:

{

0x0000000000400521 <main+0>: 55 push %rbp

0x0000000000400522 <main+1>: 48 89 e5 mov %rsp,%rbp

foo(2, 5);

0x0000000000400525 <main+4>: be 05 00 00 00 mov $0x5,%esi

0x000000000040052a <main+9>: bf 02 00 00 00 mov $0x2,%edi

0x000000000040052f <main+14>: e8 d2 ff ff ff callq 0x400506 <foo>

return 0;

0x0000000000400534 <main+19>: b8 00 00 00 00 mov $0x0,%eax

}

0x0000000000400539 <main+24>: c9 leaveq

0x000000000040053a <main+25>: c3 retq

End of assembler dump

disassemble命令的/m指示显示汇编指令的同时，显示相应的程序源码；/r指示显示十六进制的计算机指令(raw instruction)。

以上输出每行指示一条汇编指令，除程序源码外共有四列，各列含义为：

1. **0x0000000000400521**: 该指令对应的虚拟内存地址
2. **<main+0>**: 该指令的虚拟内存地址偏移量
3. **55**: 该指令对应的计算机指令
4. **push %rbp**: 汇编指令

**4.汇编style：**

基本的流程走完了，该引入汇编了。

i代表指令（instruction）

不很确定，至少你不能用instruction代替i，至少，先理解为汇编的意思。

前边的指令加上i就显示了汇编代码，例如：

n(ext)i

s(tep)i

要想一步一步看汇编代码和执行过程，

(gdb)ni

(gdb)si

是必不可少的，不过你可以用回车表示继续使用上一次的命令。

.