Linux 下 objdump 使用方法

linux 下 objdump 命令常见用法举例:

objdump -x obj:以某种分类信息的形式把目标文件的数据组成输出；<可查到该文件的的所有动态库>

objdump -t obj:输出目标文件的符号表（）

objdump -h obj:输出目标文件的所有段概括（）

objdump -j ./text/.data -S obj:输出指定段的信息（反汇编源代码）

objdump -S obj:输出目标文件的符号表（） 当 gcc -g 时打印更明显

objdump -j .text -Sl stack1 | more

-S 尽可能反汇编出源代码，尤其当编译的时候指定了-g 这种调试参数时，

效果比较明显。隐含了-d 参数。

-l 用文件名和行号标注相应的目标代码，仅仅和-d、-D 或者-r 一起使用

使用-ld 和使用-d 的区别不是很大，在源码级调试的时候有用，要求

编译时使用了-g 之类的调试编译选项。

-j name 仅仅显示指定 section 的信息

如何使用 linux 下 objdump 命令对任意一个二进制文件进行反汇编？

可以使用如下命令：

objdump -D -b binary -m i386 a.bin

-D 表示对全部文件进行反汇编，-b 表示二进制，-m 表示指令集架构，a.bin 就是我们要反汇编的二进制文件

objdump -m 可以查看更多支持的指令集架构，如 i386:x86-64，i8086 等

另外上面的所有 objdump 命令的参数同样适用于 arm-linux-objdump。

同时我们也可以指定 big-endian 或 little-endian（-EB 或-EL），我们可以指定从某一个位置开始反汇编等。

objdump 命令是 Linux 下的反汇编目标文件或者可执行文件的命令，它还有其他作用，下面以 ELF 格式可执行文件 test 为例详细介绍：

objdump -f test　显示 test 的文件头信息

objdump -d test　反汇编 test 中的需要执行指令的那些 section

objdump -D test　与-d 类似，但反汇编 test 中的所有 section

objdump -h test　显示 test 的 Section Header 信息

objdump -x test　显示 test 的全部 Header 信息

objdump -s test　除了显示 test 的全部 Header 信息，还显示他们对应的十六进制文件代码

输出到 txt 文件 objdump -s test.so>test.txt

同时可以用命 nm,strace,gdb.