定义数据用.section .data

如.section .data

Output:

.ascii “hello, world”

定义代码段用.section .text

同时需要一个代码入口.globl \_start（这种情况是用as和ld时），而直接用gcc时需要将\_start换成main

输出时，用系统调用 int $0x80，其相应寄存器的值：

Eax:功能用，固定为4

Ebx:文件描述符，1表示标准输入

Ecx:输出的字符串地址

Edx:字符串的长度

调试汇编程序：

用as –gstabs –o asm.o asm.s

ld –o asm asm.o

调用printf

在ld中用ld –dynamic-linker /lib/ld-linux-so.2 –o asm –lc asm.o

用gcc就不用指定动态链接，便是要将入口改为main

数据段中定义数据：

|  |  |
| --- | --- |
| **Directive** | **Data Type** |
| .ascii | Text |
| .asciz | Null terminated text string |
| .byte | Byte |
| .double | Double precision floating |
| .float | Single precision floating |
| .int | 32-bit integer number |
| .long | 32-bit integer number (same as .int) |
| .octa | 16-byte integer number |
| .quad | 8-byte integer number |
| .short | 16-bit integer number |
| .single | Single precision floating |

静态符号的定义

.equ factor, 3

.equ LINUX\_SYS\_CALL, 0x80

.bss段变量定义：

.comm Declares a common memory area for data that is not initialized

.lcomm Declares a local common memory area for data that is not initialized

.section .bss

.lcomm buffer, 10000