这里面的三个函数openlog， syslog， closelog是一套系统日志写入接口。

首先系统里应该具有syslog等程序，ubuntu下可以apt-get install sysklogd安装。（这里选用的sysklogd， 还有rsyslog，syslog-ng等日志软件）

通常syslog守护进程读取三种格式的记录消息。此守护进程在启动时读一个配置文件。一般来说，其文件名为/etc/syslog.conf，该文件决定了不同种类的消息应送向何处。例如，紧急消息可被送向系统管理员，并在控制台上显示，而警告消息则可记录到一个文件中。该机制提供了syslog函数，其调用格式如下

```
1. #include <syslog.h> //头文件
2. void openlog (char*ident, int option, int facility);
3. void syslog(int priority, char*format,.....);
4. void closelog();
```

调用openlog是可选择的。如果不调用openlog，则在第一次调用syslog时，自动调用openlog。调用closelog也是可选择的，它只是关闭被用于与syslog守护进程通信的描述符。调用openlog 使我们可以指定一个ident，以后， 此ident 将被加至每则记录消息中。ident 一般是程序的名称（例如 ，cron ，ine 等）。

简单的程序实例

```
1. #include <syslog.h>
2.
3. int main(int argc, char **argv) {
4.     openlog("zooyo", LOG_CONS | LOG_PID, 0);
5.     syslog(LOG_INFO,
6.             "This is a syslog test message generated by program %sn",
7.             argv[0]);
8.     closelog();
9.     return 0;
10. }
```

编译生成可执行程序后，运行一次程序将向/var/log/message文件添加一行信息如下：

Mar 2 16:17:52 ubuntu zooyo[1380]: This is a syslog test message generated by program ./a.out

openlog函数：

第一个参数ident将是一个标记，ident所表示的字符串将固定地加在每行日志的前面以标识这个日志，通常就写成当前程序的名称以作标记。
第二个参数option是下列值取与运算的结果：
LOG_CONS　　直接写入系统控制台，如果有一个错误，同时发送到系统日志记录。
LOG_NDELAY　　立即打开连接（通常，打开连接时记录的第一条消息）。
LOG_NOWAIT　　不要等待子进程，因为其有可能在记录消息的时候就被创建了（GNU C库不创建子进程，所以该选项在Linux上没有影响。）
LOG_ODELAY　　延迟连接的打开直到syslog函数调用。（这是默认情况下，需要没被指定的情况下。）
LOG_PERROR　　（不在SUSv3情况下）同时输出到stderr（标准错误文件）。
LOG_PID　　包括每个消息的PID。

第三个参数facility是用来指定记录消息程序的类型。它让指定的配置文件，将以不同的方式来处理来自不同方式的消息。
它的值可能为 LOG_KERN、LOG_USER、LOG_MAIL、LOG_DAEMON、LOG_AUTH、LOG_SYSLOG、LOG_LPR、LOG_NEWS、LOG_UUCP、LOG_CRON 或 LOG_AUTHPRIV。

LOG_AUTH　——认证系统：login、su、getty等
LOG_AUTHPRIV　——同LOG_AUTH，但只登录到所选择的单个用户可读的文件中
LOG_CRON　　——cron守护进程
LOG_DAEMON　　——其他系统守护进程，如routed
LOG_FTP　——文件传输协议：ftpd、tftpd
LOG_KERN　——内核产生的消息
LOG_LPR　——系统打印机缓冲池：lpr、lpd
LOG_MAIL　——电子邮件系统
LOG_NEWS　——网络新闻系统
LOG_SYSLOG　——由syslogd（8）产生的内部消息
LOG_USER　——随机用户进程产生的消息
LOG_UUCP　——UUCP子系统
LOG_LOCAL0~LOG_LOCAL7　——为本地使用保留

Syslog为每个事件赋予几个不同的优先级：

LOG_EMERG ——紧急情况

LOG_ALERT ——应该被立即改正的问题，如系统数据库破坏

LOG_CRIT ——重要情况，如硬盘错误

LOG_ERR ——错误

LOG_WARNING ——警告信息

LOG_NOTICE ——不是错误情况，但是可能需要处理

LOG_INFO ——情报信息

LOG_DEBUG ——包含情报的信息，通常旨在调试一个程序时使用