纯C日志函数库--zlog上手指南

zlog是一个高性能、线程安全、灵活、概念清晰的纯C日志函数库。作者编写的初衷是因为log4c不给力，而log4j、log4cxx等log4系列的日志函数库在概念上不正确。zlog继承了unix操作系统中syslog的伟大传统，将日志系统的分类、级别、输出方向、输出格式完全解耦。

**1.安装**

下载 <https://github.com/downloads/HardySimpson/zlog/zlog-latest-stable.tar.gz>

解压, 安装

$ tar -zxvf zlog-latest-stable.tar.gz

$ cd zlog-x.x.x/

$ ./configure --enable-test # 把测试程序和案例也编译下，反正不安装

$ make

$ sudo make install

# 默认安装在/usr/local/下，包括libzlog.so和zlog.h

把测试程序和案例也编译下，反正不安装

默认安装在/usr/local/下，包括libzlog.so和zlog.h

# add one line

$ sudo vi /etc/ld.so.conf

/usr/local/lib

$ sudo ldconfig

在你的程序运行之前，保证libzlog.so在系统的动态链接库加载器可以找到的目录下。上面的命令适用于linux，别的系统自己想办法。

**2.介绍一下配置文件**

zlog里面有三个重要的概念,category,format，rule

分类(Category)用于区分不同的输入，代码中的分类变量的名字是一个字符串，在一个程序里面可以通过获取不同的分类名的category用来后面输出不同分类的日志，用于不同的目的。

格式(Format)是用来描述输出日志的格式，比如是否有带有时间戳， 是否包含文件位置信息等，上面的例子里面的格式simple就配置成简单的用户输入的信息+换行符。

规则(Rule)则是把分类、级别、输出文件、格式组合起来，决定一条代码中的日志是否输出，输出到哪里，以什么格式输出。简单而言，规则里面的分类字符串和代码里面的分类变量的名字一样就匹配，当然还有更高级的纲目分类匹配。规则彻底解耦了各个元素之间的强绑定，例如log4j就必须为每个分类指定一个级别（或者从父分类那里继承），这在多层系统需要每一层都有自己的级别要求的时候非常不方便。

现在试着写配置文件，配置文件名无所谓，放在哪里也无所谓，反正在zlog\_init()的时候可以指定

$ cat /etc/zlog.conf

[formats]

simple = "%m%n"

[rules]

my\_cat.DEBUG >stdout; simple

在目前的配置文件的例子里面，可以看到my\_cat分类，>=debug等级的日志会被输出到stdout(标准输出)，并且输出的格式是simple这个格式，也就是用户输入信息+换行符。如果要输出到文件并控制文件大小为1兆，规则的配置应该是

my\_cat.DEBUG "/var/log/aa.log", 1M; simple

**3.在代码中使用**

$ vi test\_hello.c

#include <stdio.h>

#include "zlog.h"

int main(int argc, char\*\* argv)

{

int rc;

zlog\_category\_t \*c;

rc = zlog\_init("/etc/zlog.conf"); /\* 路径要和上面创建的一致，可以不放在/etc下 \*/

if (rc) {

printf("init failed\n");

return -1;

}

c = zlog\_get\_category("my\_cat");

if (!c) {

printf("get cat fail\n");

zlog\_fini();

return -2;

}

ZLOG\_INFO(c, "hello, zlog");

zlog\_fini();

return 0;

}

**4.编译、然后运行!**

$ cc -c -o test\_hello.o test\_hello.c -I/usr/local/include

$ cc -o test\_hello test\_hello.o -L/usr/local/lib -lzlog

$ ./test\_hello

hello, zlog

**5.高级功能**

 \*  syslog分类模型，比log4j模型更加直接了当

 \*  日志格式定制，类似于log4j的pattern layout

 \*  多种输出，包括动态文件、静态文件、stdout、stderr、syslog、用户自定义输出函数

 \*  运行时手动或自动刷新配置（同时保证安全）

 \*  高性能，在我的笔记本上达到72'000条日志每秒, 大概是syslog(3)配合rsyslogd的200倍速度

 \*  用户自定义等级

 \*  多线程和多进程环境下保证安全转档

 \*  精确到微秒

 \*  简单调用包装dzlog（一个程序默认只用一个分类）

 \*  MDC，线程键-值对的表，可以扩展用户自定义的字段

 \*  自诊断，可以在运行时输出zlog自己的日志和配置状态

 \*  不依赖其他库，只要是个POSIX系统就成(当然还要一个C99兼容的vsnprintf)

**6.相关链接：**

下载: <https://github.com/downloads/HardySimpson/zlog/zlog-latest-stable.tar.gz>   
  
zlog使用手册: <https://github.com/downloads/HardySimpson/zlog/UsersGuide-CN.pdf>

问题讨论区： <https://github.com/HardySimpson/zlog/issues>   
  
英文主页: <http://hardysimpson.github.com/zlog>   
  
中文主页: <http://www.oschina.net/p/zlog>   
  
作者博客: <http://my.oschina.net/HardySimpson/blog>   
  
作者email: HardySimpson1984@gmail.com