

平凡的程序员

喜欢编程、算法、虚拟机、搜索引擎、编译器、网络、游戏设计、操作系统等所有计算机中好玩的东西

目录视图

摘要视图

RSS 订阅

个人资料



feixiaoxing



访问： 1555211 次
积分： 19072 分
排名： 第111 名

原创： 296 篇 转载： 0 篇
译文： 0 篇 评论： 2085 条

文章搜索

文章分类

- 代码测试 (4)
- C/C++ (31)
- 数据结构和算法 (65)
- 多线程编程 (20)
- c语言和设计模式 (26)
- 搜索引擎的那些事 (8)
- Linux开发 (17)
- 操作系统 (28)
- 随想录 (62)
- 第三方库编译 (3)
- linux驱动编写 (5)
- verilog学习记 (4)
- 从skyye学习arm (5)
- fltk学习记 (17)

文章存档

- 2014年06月 (3)
- 2014年05月 (16)
- 2014年04月 (2)
- 2014年03月 (1)

有奖征资源，博文分享有内涵 5月推荐博文汇总 大数据读书汇--获奖名单公布 2014 CSDN博文大赛

linux下的C语言开发（动态库）

分类：Linux开发 2012-01-16 01:18 7389人阅读 评论(3) 收藏 举报

linux 语言 c windows dll api

【声明：版权所有，欢迎转载，请勿用于商业用途。联系信箱：feixiaoxing @163.com】

动态链接库不是linux独有的特性，在windows下面也存在这样的特性。一般来说，windows下面的动态连接库是以*.dll作为结尾的，而linux下面的动态连接库是以*.so结尾的。和静态链接库相比，动态连接库可以共享内存资源，这样可以减少内存消耗。另外，动态连接是需要经过操作系统加载器的帮助才能被普通执行文件发现的，所以动态连接库可以减少链接的次数。有了这个特点，我们就不难发现为什么很多软件的补丁其实都是以动态库发布的。

那么，在Linux上动态库是怎么生成的呢？

[cpp]

```
01. #include "test.h"
02.
03. int add(int a, int b)
04. {
05.     return a + b;
06. }
```

头文件格式，

[cpp]

```
01. #ifndef _TEST_H
02. #define _TEST_H
03.
04. int add(int a, int b);
05. #endif
```

此时如果我们想要生成动态库，要做的工作其实非常简单，输入gcc -shared -fPIC -o libtest.so test.c即可。回车后输入ls，我们就可以发现当前目录下面出现了libtest.so文件。

[cpp]

```
01. #include <stdio.h>
02. #include "test.h"
03.
04. int main()
05. {
06.     printf("%d\n", add(2, 3));
07.     return 1;
08. }
```

在上面的代码当中，我们发现使用到了add函数，那么此时如何才能生成一个执行文件呢？也很简单，输入gcc hello.c -o hello ./libtest.so。然后输入./hello，此时可以验证一下执行文件运行是否正确。在编写静态库的时候，我说过静态库是汇编链接到执行文件当中的，而动态库不会。朋友们可以做个小实验，删除libtest.so，然后输入./hello。此时大家可以看看系统有没有错误返回？

这个时候，有的朋友就会问了，那在windows下面dll应该怎么编写呢？其实也不难，只要在test.h上面稍作改

2014年02月 (2)

展开

阅读排行

- linux下的C语言编程（总 (50299)
- 一步一步写算法（之 算法 (47193)
- 一步一步写算法（之hasl (43308)
- 用汇编的眼光看C++（之 (29959)
- 多线程的那点儿事（基础 (28561)
- linux下的C语言开发（网 (23823)
- 多线程的那点儿事（之大 (23799)
- 一步一步写算法（之单向 (23744)
- 一步一步写算法（之快速 (23255)
- 随想录（程序员和收入） (22290)

推荐文章

最新评论

- 一步一步写算法（之hash表）
Brightziu: data % 10 这个为啥要取余数呢？
- 多线程的那点儿事（基础篇）
EbowTang: 草，牛逼了！我就邪恶的全盘学习了！
- 随想录（移动app下的生活）
zistym: 为什么民营企业的app要比国营公司的app好用得多？像农业银行的app, 太烂了。国营的基本是服务领...
- 随想录（从apple的swift语言说走w8708812: 博主，你好，我从去年开始拜读博主的文章了，收获很大，“我们要记住技术是为业务服务的，只有真正给客户创...
- linux下的C语言开发（动态库）
AmelieDan: 简单，干练
- 多线程的那点儿事（之数据互斥）
springontime: 或者我认为应该写成while(flag && (turn == index))
- C语言和设计模式（继承、封装、qq584253394: 恩。其实都是主要靠指针来实现抽象了。
- 一步一步写算法（之A*算法）
ios-开发者: 不知道 那两个函数传入的value数组从哪里来得，想请教
- 用汇编的眼光看C++（之虚函数）
xueerfei: LZ这块是不是有点小问题，调用顺序应该是：edx -> [, ecx里的内容是ptr，所以是两...
- 一步一步写算法（之洗牌算法）
wocaonilaozi: 楼主的程序每次运行后的结果都是一样的，因为他使用的是系统默认的种子，rand()使用前要有种子；应该...

友情链接

- 酷壳
- 云风
- 李先静
- 刘未鹏
- 高德纳
- 孟岩
- 潘爱民
- 周伟明
- gamers
- codeproject

变即可。其他的步骤和静态库的操作是基本类似的。

[cpp]

```
01. #ifndef _TEST_H
02. #define _TEST_H
03.
04. #ifdef USR_DLL
05. #define DLL_API __declspec(dllexport)
06. #else
07. #define DLL_API __declspec(dllimport)
08. #endif
09.
10. DLL_API int add(int a, int b);
11.
12. #endif
```

- 上一篇 linux下的C语言开发（静态库）
- 下一篇 linux下的C语言开发（定时器）

更多 0

顶 13 踩 1

主题推荐

- c语言
- 库
- linux
- 操作系统
- 内存

猜你在找

- 多线程的那点儿事（之多线程调试）
- Linux驱动编写（块设备驱动代码）
- 一步一步写算法（之二叉树广度遍历）
- 搜索引擎的那些事（网页下载）
- 使用socket的Linux上的C语言文件传输顺序服务器和
- 一步一步写算法（之双向链表）
- 一步一步写算法（之爬楼梯）
- linux驱动面试题目汇总
- 【Boost】boost::string_algo详解7——join的应用
- 点评圈内云战略——东软篇

查看评论

3楼 AmelieDan 2014-05-29 22:35发表



简单，干练

2楼 Code_Fly 2013-08-06 11:29发表



入门，很基础，但很实用。

1楼 angle555945 2012-01-17 01:51发表



学习了

您还没有登录,请[\[登录\]](#)或[\[注册\]](#)

* 以上用户言论只代表其个人观点，不代表CSDN网站的观点或立场

核心技术类目

- 全部主题
- Java
- VPN
- Android
- iOS
- ERP
- IE10
- Eclipse
- CRM
- JavaScript
- Ubuntu
- NFC
- WAP
- jQuery
- 数据库
- BI
- HTML5
- Spring
- Apache
- Hadoop
- .NET
- API
- HTML
- SDK
- IIS
- Fedora
- XML
- LBS
- Unity
- Splashtop
- UML
- components
- Windows Mobile
- Rails
- QEMU
- KDE
- Cassandra
- CloudStack
- FTC
- coremail
- OPhone
- CouchBase
- 云计算
- iOS6
- Rackspace
- Web App
- SpringSide
- Maemo
- Compuware
- 大数据
- apttech
- Perl
- Tornado
- Ruby

华清远见

51cto

IT 人生

raw-os

[Hibernate](#) [ThinkPHP](#) [Spark](#) [HBase](#) [Pure](#) [Solr](#) [Angular](#) [Cloud Foundry](#) [Redis](#) [Scala](#)
[Django](#) [Bootstrap](#)

[公司简介](#) | [招贤纳士](#) | [广告服务](#) | [银行汇款帐号](#) | [联系方式](#) | [版权声明](#) | [法律顾问](#) | [问题报告](#) | [合作伙伴](#) | [论坛反馈](#)
[网站客服](#) [杂志客服](#) [微博客服](#) webmaster@csdn.net 400-600-2320

京 ICP 证 070598 号

北京创新乐知信息技术有限公司 版权所有

江苏乐知网络技术有限公司 提供商务支持

Copyright © 1999-2014, CSDN.NET, All Rights Reserved 