

# Make Progress Everyday!

☰ 目錄視圖 ☰ 摘要視圖 RSS 訂閱

個人資料



adcxf

訪問：172781次

積分：3421分

排名：第1176名

原創：110篇 轉載：268篇

譯文：1篇 評論：101條

文章搜索

文章分類

- [C#/Visual C# \(2\)](#)
- [C/C++/VC \(114\)](#)
- [Java/JS/JSP/2ee等\(4\)](#)
- [Linux/unix/shell \(66\)](#)
- [TCP/IP協議棧研究\(4\)](#)
- [Windows系統\(3\)](#)
- [健康文摘\(7\)](#)
- [其他雜項\(29\)](#)
- [分佈式開發\(26\)](#)
- [前端開發\(1\)](#)
- [投資理財\(2\)](#)
- [數據庫原理及技術\(4\)](#)
- [數據結構和算法\(56\)](#)
- [生活/休閒/娛樂\(12\)](#)
- [網絡原理及編程\(23\)](#)
- [網絡安全/防火牆/IDS \(7\)](#)
- [設計模式/軟工/UML \(13\)](#)
- [高質量編程\(11\)](#)

文章存檔

- [2012年02月\(2\)](#)
- [2012年01月\(1\)](#)
- [2011年09月\(1\)](#)
- [2011年08月\(1\)](#)
- [2011年06月\(1\)](#)

展開

閱讀排行

新版論壇系列介紹之二——功能介紹篇 公告：CSDN博客頻道博客搬家功能上線！ CSDN產品客服新浪微博正式上線公告  
中國最大規模移動開發者高水平盛會 沒有重量只有質量：iPad版《程序員雜誌》應用上線 “第一次親密接觸”——有獎徵文活動

## linux c++動態鏈接庫so編寫

分類：C/C++/VC Linux/unix/shell

2011-04-11 17:07

983人閱讀

評論 (1)

收藏

舉報

Linux下的動態鏈接庫是.so文件，即：Shared Object，下面是一個簡單的例子說明如何寫.so以及程序如何動態載入.so中的函數和對象。

testso.h:

```
#ifndef _TESTSO_H
#define _TESTSO_H
extern "C" {
    int myadd(int a, int b);
    typedef int myadd_t(int, int); // myadd function type
}
#endif // _TESTSO_H
```

testso.cpp:

```
#include "testso.h"

extern "C"
int myadd(int a, int b)
{
    return a + b;
}
```

編譯so：

```
g++ -shared -fPIC -o testso.so testso.cpp
```

注意，-shared參數和-fPIC參數非常重要：

-shared告訴gcc要生成的是動態鏈接庫；

-fPIC告訴gcc生成的代碼是非位置依賴的，方面的用於動態鏈接。

在主程序裡調用這個動態鏈接庫：

main.cpp:

```
#include
#include
// for dynamic library函數

#include "testso.h"

void print_usage(void)
{
    printf("Usage: main SO_PATH/n");
}
```

[gettimeofday函數](#) (6200)  
[ubuntu gtk安裝](#) (6013)  
[回溯法介紹](#) (4732)  
[如何成爲一個優秀的程序](#) (4390)  
[boost bind使用指南](#) (4141)  
[rsync使用指南](#) (3988)  
[linux內核調試指南](#) (3968)  
[ulimit用法](#) (3619)  
[expr命令用法-實例講解](#) (3210)  
[svn使用簡單教程](#) (3200)

#### 評論排行

[如何成爲一個優秀的程序](#) (32)  
[回溯法介紹](#) (5)  
[Windows語音編程初步](#) (5)  
[boost bind使用指南](#) (5)  
[黑客白皮書:如何成爲一名](#) (4)  
[設計包含min函數的棧](#) (4)  
[小菜編程成長記系列](#) (3)  
[strcpy實現](#) (3)  
[迭代器失效小結](#) (3)  
[回溯法實現](#) (2)

#### 推薦文章

\* [超敏捷 Java 开发框架](#)  
[ServcieFramework](#) 作者祝海林专访  
\* [.Net企业级应用架构设计之设计原则和模式](#)  
\* [Dojo.grid.DataGrid 编程篇\(1\)](#)  
\* [树形结构的处理——组合模式 \(一\)](#)  
\* [用户控件中动态加入脚本引用](#)  
\* [Linux Shell函数返回值](#)  
\* [Class文件分析](#)

#### 最新評論

[boost bind使用指南](#)  
semiconductor3: "有函數: void f(int & x) { ++x; } 然後: int n = 0; bind...  
[小菜編程成長記系列](#)  
sanguorewrite: 用例錯誤，不是好教程  
[小菜編程成長記系列](#)  
ccppw: 做爲一個教材還不錯。各方面都寫到了。  
[小菜編程成長記系列](#)  
sanguorewrite: 原來一個小小的計算器也可以寫出這麼脫褲子放屁代碼的代碼  
[回溯法介紹](#)  
ying\_24ying: 挺好的。謝謝。要是能舉些例子就更好了。  
[boost bind使用指南](#)  
wojiushi3344: 寫得很好，頂一個！！  
[設計包含min函數的棧](#)  
zenme: @Thinkbig:直接保存最小值的話，當最小值被pop出去之後，就需要遍歷才能找回最小值。m\_da...  
[GFS, HDFS, Blob File System](#)  
caeser1234: 不清楚GFS 裡面允許,current client write to the same file ...  
[mmap共享內存](#)  
kasagawa: ding~  
[如何成爲一個優秀的程序員](#)  
copy\_zn: 好

```
int main(int argc, char *argv[])
{
    if (2 != argc) {
        print_usage();
        exit(0);
    }

    const char *soname = argv[1];

    void *so_handle = dlopen(soname, RTLD_LAZY); // 載入.so文件
    if (!so_handle) {
        fprintf(stderr, "Error: load so `%s' failed.\n", soname);
        exit(-1);
    }

    dlerror(); // 清空錯誤信息
    myadd_t *fn = (myadd_t*)dlsym(so_handle, "myadd"); // 載入函數
    char *err = dlerror();
    if (NULL != err) {
        fprintf(stderr, "%s\n", err);
        exit(-1);
    }

    printf("myadd 57 + 3 = %d\n", fn(57, 3)); // 調用函數

    dlclose(so_handle); // 關閉so句柄
    return 0;
}
```

編譯主程序：

**g++ main.cpp -o main -ldl**

注意：要加上-ldl

好了，上面就是如何寫和調用動態庫中的C函數。

對於C++中的類，不能直接導出，需要通過繼承的方式才能從動態鏈接庫中導出：

=====

testso.h:

```
#ifndef _TESTSO_H
#define _TESTSO_H
```

// 只能通過基類調用，因此需要先定義一個基類，然後在create中生成真正需要生成的對象。

```
class Base
{
public:
    int a, b;

    virtual int add(void)
    {
        return -1;
    }
};
```

```
class A : public Base
```

電子書

得益網

技術博客

[absurd](#)

[C++的羅浮宮](#)

[許式偉專欄](#)

[陳皓專欄](#)

[編程中編譯人生](#)

[C++ 有價值blog索引](#)

[Kendiv \(安全\)](#)

[榮耀](#)

[STL中文站](#)

[C++之父](#)

[C++參考大全](#)

[ithao123](#)

[龍互中華](#)

[創的技術博客](#)

[寒星軒](#)

[k-eckel](#)

技術網站

[codeproject](#)

[codeguru](#)

[csdn](#)

[sourceforge](#)

[cplusplus](#)

文學

[中國帝王](#)

```
{
public:
    int add(void);
};

extern "C" {
    Base* create(void);
    void destroy(Base *p);

    typedef Base* create_t(void); // create factory
    typedef void destroy_t(Base*); // destroy
}

#endif // _TESTSO_H

testso.cpp:

#include "testso.h"

int A::add(void)
{
    return a + b;
}

extern "C"
{
    Base* create(void) // 注意：create函數必須返回Base的對象，不能直接返回A的
                        // 對象，否則後面調用A::add()的時候會提示錯誤。
    {
        return new A;
    }

    void destroy(Base *p)
    {
        if (p) delete p;
    }
}

main.cpp: // 這裡需要注意

#include

#include
#include
#include "testso.h"

void print_usage(void)
{
    printf("Usage: myso SO_PATH/n");
}

int main(int argc, char *argv[])
{
    if (2 != argc) {
        print_usage();
        exit(0);
    }
}
```



```

}

const char *soname = argv[1];

void *so_handle = dlopen(soname, RTLD_LAZY);
if (!so_handle) {
    fprintf(stderr, "Error: load so `%s' failed.\n", soname);
    exit(-1);
}

dlerror();
create_t *create = (create_t*) dlsym(so_handle, "create");
if (NULL != err) {
    fprintf(stderr, "%s/n", err);
    exit(-1);
}
Base *pa = create();

pa->a = 57;
pa->b = 3;
printf("A.add(57, 3)=%d/n", pa->add()); // 注意，這里通過虛函數實現了
// 對A::add()的調用。

destory_t *destory = (destory_t*) dlsym(so_handle, "destory");
if (NULL != err) {
    fprintf(stderr, "%s/n", err);
    exit(-1);
}
destory(pa);
pa = NULL;

dlclose(so_handle);

printf("DONE!/n");
return 0;
}

```

上一篇：[有用的性能數據，請記住它吧~~](#)

分享到：

下一篇：[hadoop運行痕跡~~](#)



[查看評論](#)

1樓 [adcxf](#) 2011-04-11 17:10發表



[e01]

您還沒有登錄,請[登錄](#)或[註冊](#)

\* 以上用戶言論只代表其個人觀點，不代表CSDN網站的觀點或立場

专区推荐内容

web上的分页分析  
HTML5手机游戏将迎美好未来  
当智能手机遇到了NFC  
Android已经开始领跑？  
多线程知识点整理  
最好的HTML 5编码教程和参考...

<< >>

更多招聘职位

我公司职位也要出现在这里

【美世咨询】诚聘互联网应用程序支持专员等热门职位  
【广州瀚信】高薪诚聘C/C++ 初级开发工程师、C/C++ 中级开发工程师等  
【巨人网络】急聘Flash以及Web前台开发人才  
【网易有道 北京】诚聘研发/前端/web/测试开发工程师  
【CSDN】高薪急聘PHP开发/UI设计/网站编辑/社区运营等

[公司简介](#) | [招贤纳士](#) | [广告服务](#) | [银行汇款帐号](#) | [联系方式](#) | [版权说明](#) | [法律顾问](#) | [问题报告](#)

京 ICP 證 070598 號

北京創新樂知信息技術有限公司版權所有

✉ 聯繫郵箱：webmaster@csdn.net

Copyright © 1999-2012, CSDN.NET, All Rights Reserved

