

MYBLOG

Unix、Linux、TCP/IP、Oracle、Java

1 04, 2005

LINUX下創建靜態、動態庫

作者fdrtnos 15:55 | [靜態鏈接網址](#) | [最新回复\(0\)](#) | [引用\(0\)](#) | [Linux](#)   

Linux下創建靜態、動態庫

源碼

```
-----a.cpp-----
#include <stdio>
int a(int i)
{
    std::printf("in a( int a) %dn", i);
    return 0;
}

-----b.cpp-----
#include <iostream>
int b(char *s)
{
    std::cout<<" in int b(char *s): "<<s<<std::endl;
    return 0;
}

-----main.cpp-----
#include <stdio>
using namespace std;
int a(int i);
int b(char *s);
int main()
{
    a(5);
    b("ok");
    printf("in mainn");
    return 0;
}
```

靜態庫

1、編譯生成ao bo文件

```
# g++ -c a.cpp b.cpp
```

2、生成archive文件libtest.a c-創建r-將文件加入libtest.a

```
#ar rc libtest.a ao bo
```

3、指定靜態庫libtest.a進行編譯

搜索

搜索

最新發表

Linux環境下的高級隱藏技術

文章分類

tec [2] ([rss](#))
Oracle [3] ([rss](#))
海闊天空 [1] ([rss](#))
Java [5] ([rss](#))
TCP/IP [3] ([rss](#))
UNIX [3] ([rss](#))
Linux [10] ([rss](#))

文章歸檔

四月**2005** [7]
三月**2005** [20]

網站鏈接

Java
Linux
linux技術
oracle home
Oracle
TCP/IP
Unix

新聞聚合

關於
資源中心
RSS 0.90
RSS 1.0
RSS 2.0
Atom 0.3

```
# g++ -o main main.cpp libtest.a
```

運行結果

```
# ./a.out
in a(int a) 5
in int b(char *s): ok
in main
```

對於libtest.a直接將裡面的代碼拷貝出來相當於靜態編譯

動態庫

1、編譯生成a.o 文件-fPIC生成.o文件時用相對地址實現代碼位置無關

```
# g++ -fPIC a.cpp b.cpp
```

2、生成動態庫libtest.so

```
# g++ -shared -o libtest.so a.o b.o
```

3、指定動態庫libtest.so進行編譯

```
# g++ main.cpp ./libtest.so
```

查看動態連接庫

```
# ldd a.out
        ./libtest.so => ./libtest.so (0x40014000)
        libstdc++-libc6.2-2.so.3 => /usr/lib/libstdc++-
libc6.2-2.so.3 (0x4002e000)
        libm.so.6 => /lib/i686/libm.so.6 (0x40071000)
        libc.so.6 => /lib/i686/libc.so.6 (0x42000000)
        /lib/ld-linux.so.2 => /lib/ld-linux.so.2
(0x40000000)
        /lib/ld-linux.so.2 => /lib/ld-linux.so.2
(0x40000000)
```

其他

1、動態連接庫3時如果

```
# g++ main.cpp libtest.so
```

編譯可通過但運行出錯在默認庫路徑下找不到libtest.so庫

```
# ./a.out
./a.out: error while loading shared libraries: libtest.so:
cannot open shared object file: No such
file or directory
```

查看連接庫

```
# ldd a.out
        libtest.so => not found
        libstdc++-libc6.2-2.so.3 => /usr/lib /libstdc++-
libc6.2-2.so.3 (0x4002c000)
        libm.so.6 => /lib/i686/libm.so.6 (0x4006f000)
        libc.so.6 => /lib/i686/libc.so.6 (0x42000000)
        /lib/ld-linux.so.2 => /lib/ld-linux.so.2
(0x40000000)
```

2、方便使用可以將靜態庫libtest.a或者動態庫libtest.so拷貝到默認庫的查找路徑裡/lib或者/usr/lib裡，編譯時直接

```
#g++ main.cpp -ltest
```

-l自動進行庫名稱擴展

```
# ldd a.out
libtest.so => /usr/lib/libtest.so (0x4002b000)
libstdc++-libc6.2-2.so.3 => /usr/lib/libstdc++-
libc6.2-2.so.3 (0x4002e000)
libm.so.6 => /lib/i686/libm.so.6 (0x40071000)
libc.so.6 => /lib/i686/libc.so.6 (0x42000000)
/lib/ld-linux.so.2 => /lib/ld-linux.so.2
(0x40000000)
```

靜態情況

```
# g++ main.cpp -ltest
# ldd a.out
libstdc++-libc6.2-2.so.3 => /usr/lib /libstdc++-
libc6.2-2.so.3 (0x4002b000)
libm.so.6 => /lib/i686/libm.so.6 (0x4006f000)
libc.so.6 => /lib/i686/libc.so.6 (0x42000000)
/lib/ld-linux.so.2 => /lib/ld-linux.so.2
(0x40000000)
```

3、靜態庫中2生成archive後可能需要ranlib libtest.a 將內容的索引寫入libtest.a中供ld連接時使用

4、-l選項要放在所編譯文件的後面，在前面會出錯

```
# g++ -ltest main.cpp
/tmp/ccbQK3kY.o: In function `main':
/tmp/ccbQK3kY.o(.text+0xc ): undefined reference to `a(int)'
/tmp/ccbQK3kY.o(.text+0x1c): undefined reference to `b(char
*)'
collect2: ld returned 1 exit status
```

最新回复

發表評論

標題:

稱呼:

郵箱地址（可選）:

個人主頁（可選）:

發表評論:



發表