

LINUX动态链接库高级应用 PDF转换可能丢失图片或格式，  
建议阅读原文

[https://www.100test.com/kao\\_ti2020/252/2021\\_2022\\_LINUX\\_E5\\_8](https://www.100test.com/kao_ti2020/252/2021_2022_LINUX_E5_8)

[A\\_A8\\_E6\\_80\\_c103\\_252937.htm](https://www.100test.com/kao_ti2020/252/2021_2022_LINUX_E5_8) 一、编写合格的动态链接库头文件 C语言的头文件,可供一个或多个程序引用,里面一般定义程序所需的常量,自定义类型及函数原型说明等.其中的函数原型说明,则供编译器检查语法,用于排除引用参数时类型不一致的错误.只有编写合格的动态链接库头文件,程序员才能正确使用动态链接库内的函数. 动态链接库头文件要采用C语言标准格式,其中的动态函数原型定义,不必象上文介绍的那样用(\*动态函数名)的描述形式.请看下面的例子:(每行开始的数字为所在行行号,为笔者添加,供注解使用) 1 /\* adatetime.h : 纵横软件制作中心雨亦奇(zhsoft@371.net)编写, 2002-03-06. \*/ 2 3 #ifndef \_\_DATETIME\_H 4 5 #define \_\_DATETIME\_H 6 7 /\* 日期结构 \*/ 8 typedef struct 9 { 10 int year. 11 int mon. 12 int day. 13 }DATETYPE. 14 15 /\* 时间结构 \*/ 16 typedef struct 17 { 18 char hour. 19 char min. 20 char sec. 21 }TIMETYPE. 22 23 int getdate(DATETYPE \*d). /\* 取当前日期 \*/ 24 int gettime(TIMETYPE \*t). /\* 取当前时间 \*/ 25 26 #endif 27 注:与上文的datetime.h文件比较,从该头文件第23,24行可以看到,动态函数getdate,gettime的原型定义改变了,不再使用(\*getdate),(\*gettime)的格式了(这种格式使用较为罗嗦).二、正确编译与命名动态链接库 为了让GCC编译器生成动态链接库,编译时须加选项-shared.(这点须牢记) LINUX系统中,为了让动态链接库能被系统中其它程序共享,其名字应符合“lib\*.so\*”这种格式.如果某个动态链接库不符合此格式,则LINUX的动

态链接库自动装入程序(ld.so)将搜索不到此链接库,其它程序也无法共享之. 格式中,第一个\*通常表示为简写的库名,第二个\*通常表示为该库的版本号.如:在我的系统中,基本C动态链接库的名字为libc.so.6,线程 pthread动态链接库的名字为libpthread.so.0等等.本文例子所生成的动态链接库的名字为libmy.so,虽没有版本号,但也符合所要求的格式.生成该动态链接库的维护文件makefile-lib内容如下:

```
1 # makefile : 纵横软件制作中心雨亦奇编写, 2002-03-07.
2 3 all : libmy.so
4 5 SRC = getdate.c gettime.c
6 7 TGT = $(SRC:.c=.o)
8 9 $(SRC) : adatetime.h
10 @touch $@
11 12 %.o : %.c
13 cc -c $?
14 15 # 动态链接库(libmy.so)生成
16 libmy.so : $(TGT)
17 cc -s -shared -o $@ $(TGT)
18
```

运行命令: `$ make -f makefile-lib $` 即生成libmy.so库.

注: 维护文件中,第17行用-shared选项以生成动态链接库,用-s选项以去掉目标文件中的符号表,从而减小文件长度.

### 三、共享动态链接库

#### 3.1 动态链接库配置文件

为了让动态链接库为系统所使用,需要维护动态链接库的配置文件--/etc/ld.so.conf.此文件内,存放着可被LINUX共享的动态链接库所在目录的名字(系统目录/lib,/usr/lib除外),各个目录名间以空白字符(空格,换行等)或冒号或逗号分隔.一般的LINUX发行版中,此文件均含一个共享目录/usr/X11R6/lib,为X window窗口系统的动态链接库所在的目录.下面看看我的系统中此文件的内容如何:

```
# cat /etc/ld.so.conf /usr/X11R6/lib /usr/zzz/lib #
```

由上可以看出,该动态库配置文件中,增加了一个/usr/zzz/lib目录.这是我自己新建的共享库目录,下面存放我新开发的可供系统共享的动态链接库.

#### 3.2 动态链接库管理命令

为了让动态链接库为系统所共享,还需运行动态链接库的管理命令--ldconfig.此执行程序存放

在/sbin目录下。ldconfig命令的用途,主要是在默认搜寻目录(/lib和/usr/lib)以及动态库配置文件/etc/ld.so.conf内所列的目录下,搜索出可共享的动态链接库(格式如前介绍,lib\*.so\*),进而创建出动态装入程序(ld.so)所需的连接和缓存文件。缓存文件默认为/etc/ld.so.cache,此文件保存已排好序的动态链接库名字列表。ldconfig通常在系统启动时运行,而当用户安装了一个新的动态链接库时,就需要手工运行这个命令。ldconfig命令行用法如下: ldconfig [-v|--verbose] [-n] [-N] [-X] [-f CONF] [-C CACHE] [-r ROOT] [-l] [-p|--print-cache] [-c FORMAT] [--format=FORMAT] [-V] [-?|--help|--usage] path... ldconfig可用的选项说明如下: (1) -v或--verbose: 用此选项时,ldconfig将显示正在扫描的目录及搜索到的动态链接库,还有它所创建的连接的名字。 (2) -n: 用此选项时,ldconfig仅扫描命令行指定的目录,不扫描默认目录(/lib,/usr/lib),也不扫描配置文件/etc/ld.so.conf所列的目录。 (3) -N: 此选项指示ldconfig不重建缓存文件(/etc/ld.so.cache)。若未用-X选项,ldconfig照常更新文件的连接。 (4) -X: 此选项指示ldconfig不更新文件的连接。若未用-N选项,则缓存文件正常更新。 (5) -f CONF: 此选项指定动态链接库的配置文件为CONF,系统默认为/etc/ld.so.conf。 (6) -C CACHE: 此选项指定生成的缓存文件为CACHE,系统默认的是/etc/ld.so.cache,此文件存放已排好序的可共享的动态链接库的列表。 (7) -r ROOT: 此选项改变应用程序的根目录为ROOT(是调用chroot函数实现的)。选择此项时,系统默认的配置文件的/etc/ld.so.conf,实际对应的为 ROOT/etc/ld.so.conf。如用-r /usr/zzz时,打开配置文件/etc/ld.so.conf时,实际打开的是/usr/zzz/etc/ld.so.conf文件。用此选项,可以大大增加动态链接

库管理的灵活性. (8) -l: 通常情况下,ldconfig搜索动态链接库时将自动建立动态链接库的连接.选择此项时,将进入专家模式,需要手工设置连接.一般用户不用此项. (9) -p或--print-cache: 此选项指示ldconfig打印出当前缓存文件所保存的所有共享库的名字. (10) -c FORMAT 或 --format=FORMAT: 此选项用于指定缓存文件所使用的格式,共有三种:old(老格式),new(新格式)和compat(兼容格式,此为默认格式). (11) -V: 此选项打印出ldconfig的版本信息,而后退出. (12) -? 或 --help 或 --usage: 这三个选项作用相同,都是让ldconfig打印出其帮助信息,而后退出. 100Test 下载频道开通, 各类考试题目直接下载。详细请访问 [www.100test.com](http://www.100test.com)