

一、新建主目录libhello

二、在主目录libhello下,新建目录lib, m4, build-aux.

其中 m4 与 build-aux目录也可不创建,主要功能是存放编译过程需要的依赖文件,让目录结构不那么乱.

三、在lib目录下编写print_hello.c 与 print_hello.h 文件

1. print_hello.c 内容如下:

```
#include <stdio.h>

int print_hello(void)
{
    printf("hello world\n");
    return 0;
}
```

2.print_hello.h 内容如下:

```
#ifndef PRINT_HELLO_H
#define PRINT_HELLO_H

void print_hello(void);

#endif
```

四、在lib目录下编写Makefile.am文件

```
lib_LTLIBRARIES = libhello.la          //定义生成动态库
libhello_la_SOURCES = print_hello.c print_hello.h //生成动态库所需的源文件
libhello_la_LDFLAGS = -version-info 1:0:0 //定义动态库版本
```

五、在libhello目录下编写Makefile.am文件

```
SUBDIRS = lib          //编译lib子目录
ACLOCAL_AMFLAGS = -I m4 //编译时需要用到额外的库
```

六、在libhello目录下执行autoscan

生成autoscan.log 与 configure.scan

1. 删除autoscan.log

2.将configure.scan 改名为 configure.ac并修改里面的内容.

```
#                                     -*- Autoconf -*-

# Process this file with autoconf to produce a configure script.

AC_PREREQ([2.69])
AC_INIT([libhello], [1.0], [liuhw@itep.com.cn])
# AC_CONFIG_SRCDIR([lib/print_hello.h])
AC_CONFIG_HEADERS([config.h])
AC_CONFIG_AUX_DIR([build-aux]) //将生成的过程文件放入build-aux目录中
AC_CONFIG_MACRO_DIR([m4])      //将宏定义文件放入m4目录中
AM_INIT_AUTOMAKE([foreign -Wall -Werror]) //编译时显示警告与错误 foreign为不检测NEWS等文件.

# Checks for programs.
AM_PROG_AR
LT_INIT
AC_PROG_CC

# Checks for libraries.

# Checks for header files.

# Checks for typedefs, structures, and compiler characteristics.
```

```
# Checks for library functions.
```

```
AC_CONFIG_FILES([Makefile  
lib/Makefile])
```

```
AC_OUTPUT
```

七、在libhello目录下执行autoreconf -i -f -v

八、在libhello目录下执行.configure

九、在libhello目录下执行make

在libhello目录下的lib目录下生成libs隐藏目录,生成的libhello.so.1.0.0动态库就在libs目录下.