<https://blog.csdn.net/u010312436/article/details/52486811>

解决多个.a静态库，链接成一个.so动态库出现：

﻿﻿

relocation R\_X86\_64\_32 against `.rodata' can not be used when making a shared object; 的错误。

我的总目标是：写一个mainApi.cpp的文件，这个文件提供API给外部用户使用。mainApi.cpp需要依赖其他模块生成的静态库（在lib文件下）。将mainApi.cpp编译

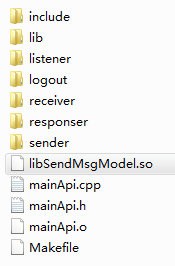
链接成一个.so动态库共外部使用。在编译的时候出现**relocation R\_X86\_64\_32 against `.rodata' can not be used when making a shared object**; 的错误。

原因是因为生成静态库时没有加 -fPIC 选项。

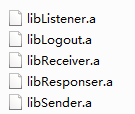
问题解决是如下（红色字体部分）：

ifeq ($(MK\_STATIC\_LIB),no)  
    TARGET = libSendMsgModel.so  
    CFLAGS += -fPIC             # 静态库链接成.so动态库，编译静态库的时候需要加 -fPIC这个参数。  
else  
    TARGET = libSendMsgModel.a  
endif

看下最后的实现目标（在主目录下生成libSendMsgModel.so文件）：



看看lib下的文件：



主目录下的Makefile:

XX = g++

AR = ar

ARFLAG = -rcs

CFLAGS = -g

CLIBS = -L./lib/ -lSender -lReceiver -lResponser -lListener -lLogout -lpthread -lrt

SUBDIRS = ./logout ./receiver ./responser ./sender ./listener

# INCLUDE = a.h b.h ... can't be defined like "INCLUDE = ./\*.h"

INCLUDES = $(wildcard ./include/\*.h ./sender/\*.h ./receiver/\*.h ./responser/\*.h)

#指定头文件目录，代码中就不需要把头文件的完整路径写出来了

INCLUDE\_DIRS = -I./include -I./sender/ -I./receiver/ -I./responser/ -I./listener -I./logout

OBJECTS = mainApi.o

export XX CFLAGS AR ARFLAG

ifeq ($(MK\_STATIC\_LIB),no)

TARGET = libSendMsgModel.so

CFLAGS += -fPIC # 静态库链接成.so动态库，编译静态库的时候需要加 -fPIC这个参数。

else

TARGET = libSendMsgModel.a

endif

ifeq ($(MK\_STATIC\_LIB),no)

$(TARGET) : $(OBJECTS) $(SUBDIRS)

$(XX) $(CFLAGS) -shared -fPIC $< -o $@ $(CLIBS) # $< 表示依赖列表的第一个 也就是 $(OBJECTS)

$(OBJECTS) : %.o : %.cpp

$(XX) -c $(CFLAGS) $< -o $@ $(INCLUDE\_DIRS)

else

$(TARGET) : $(OBJECTS) $(SUBDIRS)

$(AR) $(ARFLAG) $@ $< # $< 表示依赖列表的第一个 也就是 $(OBJECTS)

$(OBJECTS) : %.o : %.cpp

$(XX) -c $(CFLAGS) $< -o $@ $(INCLUDE\_DIRS)

endif

$(SUBDIRS):ECHO

+$(MAKE) -C $@

ECHO:

@echo $(SUBDIRS)

@echo begin compile

.PHONY : clean

clean:

for dir in $(SUBDIRS);\

do $(MAKE) -C $$dir clean||exit 1;\

done

rm -rf $(TARGET) $(OBJECTS) ./lib/\*.a

版权声明：本文为博主原创文章，未经博主允许不得转载。 https://blog.csdn.net/u010312436/article/details/52486811