

海思 arm-himix200-linux-g++交叉编译 opencv , 移植板端问题。

一 . 成功案例 :

1. 运行如下代码 (文件名为 opencv_image_crop.cpp), 代码功能为裁剪 (crop) 图像。

```
#include"opencv2/imgproc/imgproc.hpp"
#include"opencv2/highgui/highgui.hpp"

using namespace cv;//使用 cv 命名空间

int main() {

    Mat srcImg=imread("./bk.jpg");//读取图片

    Mat roi_img;
    roi_img = srcImg(Range(0,112),Range(50,162));

    imwrite("./crop2.jpg",roi_img);//保存结果图片

    return 0;
}
```

2. 首先将交叉编译后的 Opencv 动态库文件放入海思 arm-himix200-linux 安装目录的 target/lib 下 , 然后在服务器终端用如下指令编译 :

```
arm-himix200-linux-g++ opencv_image_crop.cpp -o crop -lpthread -lm -L../lib
-lopenv_imgproc -lopenv_highgui -lopenv_core -lgomp -lopenv_imgcodecs
```

-L 用来链接运行该程序所需要的动态库。

3. 将动态链接库移入 NFS 服务器的/cross_compiled_opencv3.4.3/lib 并在 minicom 板子终端输入如下指令 (在此之前要把这个 lib 文件夹拷贝到可执行文件的上两层目录):

```
export LD_LIBRARY_PATH=/mnt/cross_compiled_opencv3.4.3/lib  
  
./crop
```

以上，运行成功。

二：失败案例一

运行如下代码(代码功能为缩放)：

```
#include "stdio.h"  
#include "opencv2/imgproc/imgproc.hpp"  
#include "opencv2/highgui/highgui.hpp"  
using namespace cv;  
int main()  
{  
    Mat srcImage=imread("two_persons_face.jpg");  
    Mat temImage,dstImage1,dstImage2;  
    temImage=srcImage;  
  
    resize(temImage,dstImage1,Size(temImage.cols/2,temImage.ro  
ws/2),0,0,INTER_LINEAR);  
    resize(temImage,dstImage2,Size(temImage.cols*2,temImage.ro  
ws*2),0,0,INTER_LINEAR);  
    imwrite("crop3.jpg",dstImage1);  
    imwrite("crop4.jpg",dstImage2);  
    return 0;  
}
```

报如下错误：

```
libopencv_core.so: symbol _ZNSt15__exception_ptr13exception_ptrC1Ev, version  
CXXABI_1.3.11 not defined in file libstdc++.so.6 with link time reference
```

原因：

交叉编译链里面的 stdc++ 库版本过低。

解决办法：

交叉编译低版本 opencv，比如 opencv2.4.10 此问题便会解决。

三．失败案例二

把 opencv 代码移入到 mpp 的代码中，必须要用 g++ 编译。Main 函数会自动去找自己要的.o 文件，而编译器会自动把他需要的.c 文件自动编译成.o 文件（本来可以定义隐式规则，但似乎海思没有写或者我没有找到）。所以我目前改了一些自动编译时需要的参数，把需要链接的头文件和库文件加入了 CFLAGS 变量中。

Makefile 编译目标文件使用 \$^ 能够自动链接 \$(OBJ) 下的.o 文件，修改 makefile 如下：

```
File Edit View Search Terminal Help
# target source
OBJS := $(SMP_SRCS:%.c=%.o)
#CC:=arm-himix200-linux-gcc
CC:=arm-himix200-linux-g++
CFLAGS += -fPIC
CFLAGS += -fpermissive
CFLAGS += $(COMM_INC)
CFLAGS += $(COMM_INC)
CFLAGS += -I$(PWD)/include
CFLAGS += -I$(PWD)/include/opencv
CFLAGS += -I$(PWD)/include/opencv2
CFLAGS += -L$(PWD)/lib
CFLAGS += -lopencv_imgproc -lopencv_highgui -lopencv_core -lgomp

MPI_LIBS += $(REL_LIB)/libhdm1.a

.PHONY : clean all

all: $(TARGET)

$(TARGET):$(COMM_OBJ) $(OBJS)
    $(CC) $(CFLAGS) -lpthread -lm -o $(TARGET_PATH)/$@ $^ -Wl,--start-group $(MPI_LIBS) $(SENSOR_LIBS) $(AUDIO_LIBS) $(REL_LIB)/libsecurec.a -Wl,--end-group

clean:
    @rm -f $(TARGET_PATH)/$(TARGET)
    @rm -f $(OBJS)
    @rm -f $(COMM_OBJ)

cleanstream:
    @rm -f *.h264
```

1,0-1 顶端

目前已经连接上库了，但是还是会报错，错误如下：

```
/home/danale/code/mpp/sample/svp/nnie/../../lib/libopencv_core.so: undefined reference
to `std::__exception_ptr::exception_ptr::~~exception_ptr()@CXXABI_1.3.11'

/home/danale/code/mpp/sample/svp/nnie/../../lib/libopencv_core.so: undefined reference
to `std::__exception_ptr::exception_ptr::swap(std::__exception_ptr::exception_ptr&)@CX
XABI_1.3.11'

/home/danale/code/mpp/sample/svp/nnie/../../lib/libopencv_core.so: undefined reference
to `std::__exception_ptr::exception_ptr::exception_ptr()@CXXABI_1.3.11'

/home/danale/code/mpp/sample/svp/nnie/../../lib/libopencv_core.so: undefined reference
to `std::current_exception()@CXXABI_1.3.11'

/home/danale/code/mpp/sample/svp/nnie/../../lib/libopencv_core.so: undefined reference
to `std::rethrow_exception(std::__exception_ptr::exception_ptr)@CXXABI_1.3.11'

/home/danale/code/mpp/sample/svp/nnie/../../lib/libopencv_core.so: undefined reference
to `std::__exception_ptr::exception_ptr::exception_ptr(std::__exception_ptr::exception_ptr
const&)@CXXABI_1.3.11'
```

目前解决办法：

或许 MPP 中的有些.C 文件只能用 gcc 编译，而我只有一个.C 文件需要用到 g++ 编译（因为 opencv 只能用 g++ 编译）。所以先把其他文件用 gcc 编译成.o 文件，然后再用 g++ 编译那个特殊的.C 文件。最后用 g++ 生成可执行文件。问题解决。