2022-07-12

1.linux umask

查看umask:umask

设置umask:umask 0077

文件权限：777-umask

Example:将文件权限设置为对owner读写

umask 0077

touch aa

ls -l aa

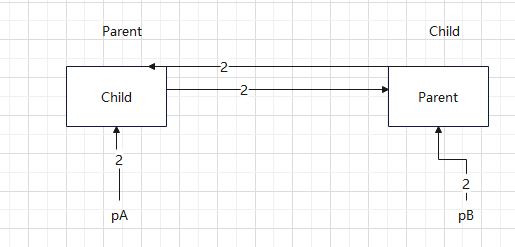
则可看出文件权限



2.shared\_ptr

new一个对象，shared\_ptr引用计数器会加1，将一个shared\_ptr拷贝到另一个shared\_ptr对象，引用计数器会加1，但使用引用传递，不会导致引用计数器加1。

当存在循环依赖关系，



造成内存泄漏。

3.weak\_ptr

只是指向内存，通过Lock定位到相应内存，获得shared\_ptr，shared\_ptr的reset是释放指针，引用计数器减一。

2022-07-13

1.android timePicker

时间控件，可以获取时间：TimePicker.getCurrentHour()、TimePicker.getCurrentMinute()

2.boost::base\_from\_member

当一个类继承boost::base\_from\_member，便可通过::member访问其对象。如下所示：

#include <iostream>

#include <sstream>

#include "boost/noncopyable.hpp"

#include "boost/utility/base\_from\_member.hpp"

class task\_processor : boost::noncopyable {

    std::ostream& log\_;

protected:

    virtual void do\_process() = 0;

public:

    explicit task\_processor(std::ostream& log) : log\_(log) {}

    void process() {

        log\_ << "starting process..." << std::endl;

        do\_process();

    }

};

class task\_processor\_fixed : boost::base\_from\_member<std::ostringstream>

                           , public task\_processor {

    typedef boost::base\_from\_member<std::ostringstream> logger\_t;

    virtual void do\_process() {

        logger\_t::member << "fix processor processing..." << std::endl;

    }

public:

    task\_processor\_fixed()

            : logger\_t()

            , task\_processor(logger\_t::member)

    {}

};

int main() {

    task\_processor\_fixed tf;

    tf.process();

    return 0;

}

3.情商

偏见是个体早期获得的一种情绪经验，偏见的情绪在童年时形成，但是支持偏见的信念是后来才出现的，偏见反应很难完全消除，可以改变的是人们对待偏见的态度！

一个工作团队内的成员，其优势和视角各有千秋，如果他们能够协调一致，相对于同一批人各自独立工作，可能会实现更出色，更有创造性，更有效的解决方案！

4.Java

abstract既可以定义类，也可以定义方法，定义类时代表该类是抽象类，没有具体实现，需要在派生类实现abstract方法。

JAVA类默认访问权限是public，具体demo：

**abstract** **class** Example0507 {

**int** x;

**abstract** **int** m1();

**abstract** **int** m2();

}

**abstract** **class** Example0507\_1 **extends** Example0507 {

**int** y;

**int** m1() {

**return** x + y;

}

}

**class** Example0507\_2 **extends** Example0507\_1 {

**int** z;

**int** m2() {

**return** x + y + z;

}

}

**public** **class** Hello {

**public** **static** **void** main(String[] args) {

Example0507\_2 mm = **new** Example0507\_2();

mm.x = 10;

mm.y = 20;

mm.z = 30;

System.***out***.print(mm.m2());

}

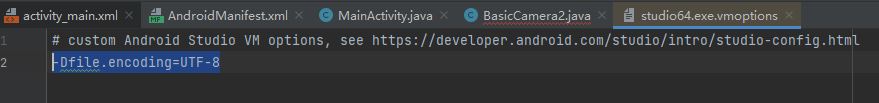
}

2022-07-14

1.Android Studio编译出现乱码

点击Help菜单，点击Edit Custom VM options，根据提示创建文件，写入一行：

-Dfile.encoding=UTF-8



2.YUV420\_888图像格式

YUV420是1个Y 对应0.25个U和0.25个V，例如YYYYYYYYUUVV，或者YYYYYYYYUVUV，YYYYYYYYVUVU，即NV12和NV21都是YUV420\_888具体实现。

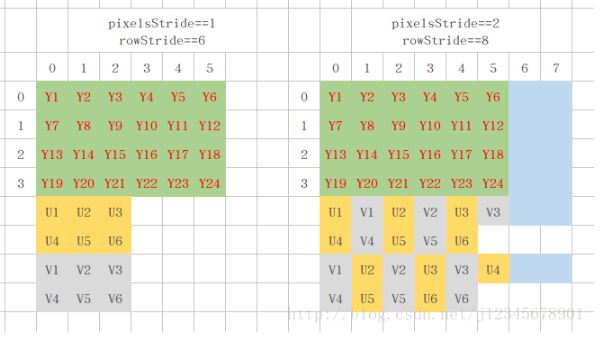
2022-7-15

1.android解决sdcard写权限问题

修改android studio的build.gradle，将targetSdk降级为26，如下所示：



2.从android YUV420\_888保存字节数组



假设图像尺寸是width \* height

原索引srcIndex：初始为0

目的索引destIndex：初始为0，指向目的字节数组

（1）将Y通道保存到字节数组

for (loop : height) { // 循环height

拷贝width字节数据

偏移行距：srcIndex += rowStride

偏移宽度：destIndex += width

}

（2）将U通道保存到字节数组

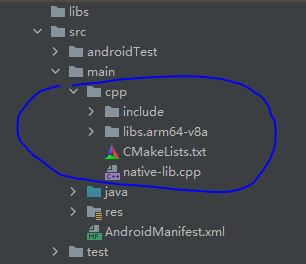
外层循环height/2

内层循环width/2

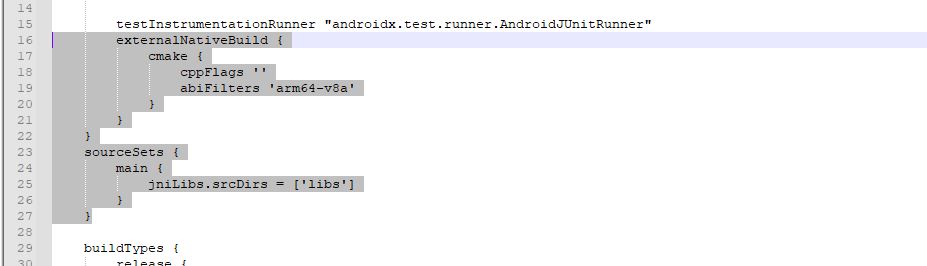
2022-07-18

1.android 集成JNI

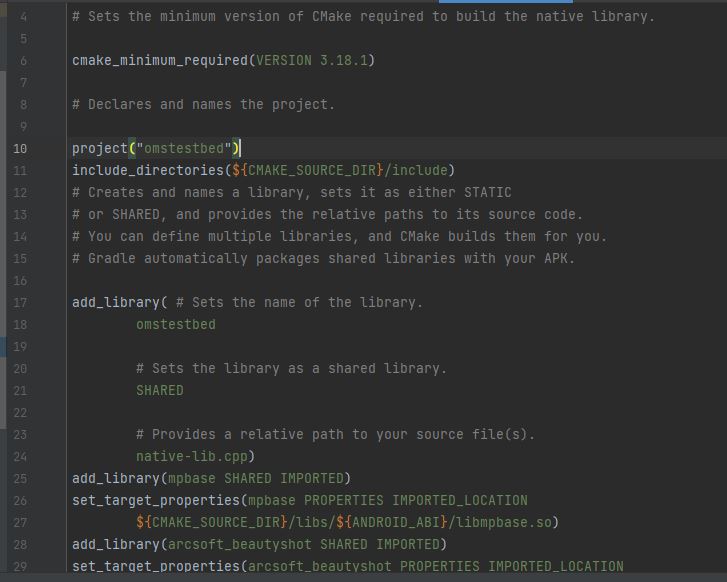
（1）目录结构



（2）修改android编译脚本build.gradle

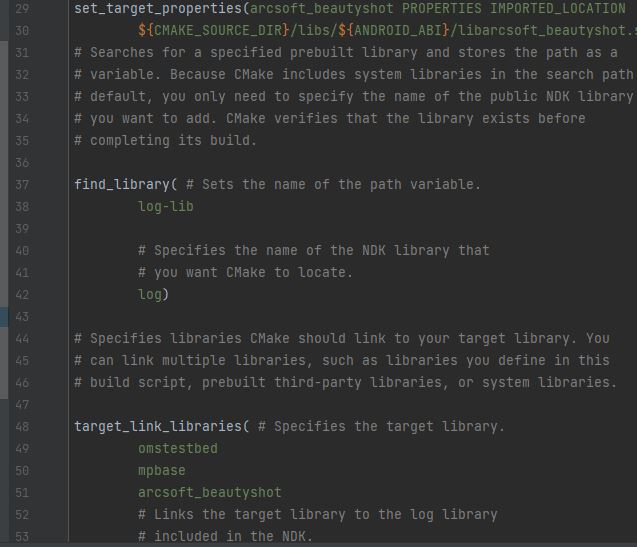


（3）修改CMakeLists.txt



2.情商

工作的单位是团队而不是个人，团队完成任务质量高低取决于群体智商的高低，群体智力不一定大于所有群体成员才能的总和，如果群体成员无法共享才智，群体智力有可能小于成员才能的总和！



2022-07-25

1.boost::conversion::try\_lexical\_convert

不需要加异常，直接将字符串转为一定格式

#include <iostream>

#include "boost/lexical\_cast.hpp"

int main() {

    int i = 0;

    const bool ok = boost::conversion::try\_lexical\_convert("bad stuff!", i);

    if (!ok) {

        std::cerr << "convert failed...!" << std::endl;

        if (boost::conversion::try\_lexical\_convert("123", i)) {

            std::cout << "i:" << i << std::endl;

        }

    }

    return 0;

}

2022-07-27

1. Camera.setPreviewCallbackWithBuffer：在摄像头开启时增加一个回调函数,在每一帧出现时调用.通过addCallbackBuffer(byte[])使用一个缓存容器来显示这些数据.

（1）Android每接受一个预览帧，都会调用该回调；

（2）该回调函数重复使用buffer；

（3）使用方法：

addCallbackBuffer=》setPreviewCallback=》addCallbackBuffer

demo:

openCamera:

camera.addCallbackBuffer(new byte[1280 \* 720 \* 3 / 2]);

camera.setPreviewCallbackWithBuffer(this.mPreviewCallback);

预览回调：

private Camera.PreviewCallback mPreviewCallback = new Camera.PreviewCallback() {  
 @Override  
 public void onPreviewFrame(byte[] bytes, Camera camera) {  
 Log.*d*("LAILE", "DDDD:" + bytes.length);  
 camera.addCallbackBuffer(bytes);

2.JAVA 接口

**interface** A {

**public** **abstract** **void** method1(**int** i);

}

**class** B **implements** A {

@Override

**public** **void** method1(**int** i) {

// **TODO** Auto-generated method stub

}

}

3.linux shell一次touch多个文件

touch {a,b}.txt

此时会创建a.txt和b.txt

4.情商：无论是预防疾病还是治疗疾病，在治疗病人的同时调节病人的情绪状态，可以获得额外的治疗效果，医学的使命被定位为治疗疾患，而忽视了病态，即病人的情绪反应！

2022-07-28

1.boost pool

#include "boost/pool/pool.hpp"

typedef struct student\_st {

    char name[10];

    int age;

} CStudent;

int main() {

    boost::pool<>student\_pool(sizeof(CStudent));

    CStudent \*const obj = (CStudent \*)student\_pool.malloc();

    obj->age = 18;

    student\_pool.free(obj);

    return 0;

}

2.boost与C++、C整数转字符串耗时demo

#include <string>

#include <cassert>

#include <sstream>

#include <cstdlib>

#include "boost/lexical\_cast.hpp"

const int LOOP = 10000000;

void lexical\_cast\_example() {

    for (int i = 0;i < LOOP;i++) {

        const std::string s = boost::lexical\_cast<std::string>(100);

        assert(s == "100");

    }

}

void cpp\_convert\_example() {

    for (int i = 0;i < LOOP;i++) {

        std::stringstream ss;

        ss << 100;

        std::string s;

        ss >> s;

        assert(s == "100");

    }

}

void c\_convert\_example() {

    for (int i = 0;i < LOOP;i++) {

        char buf[64] = "";

        std::snprintf(buf, sizeof(buf), "%i", 100);

        std::string s = buf;

        assert(s == "100");

    }

}

int main() {

    lexical\_cast\_example();         // 0.94s

    //cpp\_convert\_example();        // 3.323s

    //c\_convert\_example();          // 1.022

    return 0;

}