CrashHandler

1、如何去收集用户的crash信息?

> 1. Android 的 Thread 类中的一个方法 setDefaultUncaughtExceptionHandler() 可以设置系统默认的异常处理器。
> 2. crash 发生时，系统会回调 UncaughtExceptionHandler 的 uncaughtException() 方法---内部可以将crash信息保存到 SD卡 或者 上传服务器 。
> 3. crash 发生时我们可以在 uncaughtException() 中弹出对话框告知用户

```
public static void setDefaultUncaughtExceptionHandler(UncaughtExceptionHandler eh) {
     defaultUncaughtExceptionHandler = eh;
 }
```

2、自定义异常处理器的实现:

```java
public class CrashHandler implements Thread.UncaughtExceptionHandler{
    private static final String TAG = CrashHandler.class.getName();
    private static final boolean DEBUG = true;
    //1-单例
    private static CrashHandler mInstance;
    private CrashHandler(){
    }
    public static CrashHandler getInstance(){
        if(mInstance == null){
            synchronized (CrashHandler.class){
                if(mInstance == null){
                    mInstance = new CrashHandler();
                }
            }
        }
        return mInstance;
    }
    //2-初始化CacheHandler
    private Thread.UncaughtExceptionHandler mDefaultCrashHandler;
    private Context mContext;
    public void init(Context context){
        mDefaultCrashHandler = Thread.getDefaultUncaughtExceptionHandler();
        Thread.setDefaultUncaughtExceptionHandler(this);
        mContext = context.getApplicationContext();
    }
    //3-进行crash信息的处理，并最终由系统的异常处理器进行收尾工作。
    @Override
    public void uncaughtException(Thread t, Throwable e) {
        //1. 存储异常信息到本地
        dumpExceptionToSDCard(e);
        //2. 存储异常信息到服务器
        uploadExceptionToServer();
        e.printStackTrace();
        //3. 存在系统默认处理器，就后续交给其执行。
        if(mDefaultCrashHandler != null){
            mDefaultCrashHandler.uncaughtException(t, e);
        }
    }
    //4-将异常信息保存到SD卡中
    private static final String PATH = Environment.getExternalStorageDirectory().getPath() + "/
    private static final String FILE_NAME = "crash";
    private static final String FILE_NAME_SUFFIX = ".trace";

    private void dumpExceptionToSDCard(Throwable ex){
        //1. 获取缓存地址
        if(!Environment.getExternalStorageState().equals(Environment.MEDIA_MOUNTED)){
            if (DEBUG){
                Log.d(TAG, "sdcard unmounted, skip dump exception");
                return;
            }
        }
        //2. 创建文件夹
        File dir = new File(PATH);
        if(!dir.exists()){
            dir.mkdirs();
        }
        //3. 获取当前时间
        long current = System.currentTimeMillis();
        String time = new SimpleDateFormat("yyyy-MM-dd HH:mm:ss").format(new Date(current));

        //3. 存储异常信息
        File file = new File(PATH + FILE_NAME + time + FILE_NAME_SUFFIX);
        try {
            PrintWriter printWriter = new PrintWriter(new BufferedWriter(new FileWriter(file))
            printWriter.println(time); //时间
            dumpPhoneInfo(printWriter);//手机信息
            printWriter.println();
            ex.printStackTrace(printWriter);//异常信息
            printWriter.close();
        } catch (Exception e) {
            Log.e(TAG, "dump crash info failed");
        }
    }
    //5-保存APP和系统的版本信息
    private void dumpPhoneInfo(PrintWriter printWriter) throws PackageManager.NameNotFoundExcep
        PackageManager packageManager = mContext.getPackageManager();
        PackageInfo packageInfo = packageManager.getPackageInfo(mContext.getPackageName(), Pack
        //1. APP版本信息
        printWriter.println("App Version:" + packageInfo.versionName + "_" + packageInfo.versio
        //2. Android版本号
        printWriter.println("OS Version:" + Build.VERSION.RELEASE + "_" + Build.VERSION.SDK_IN
        //3. 手机制造商
        printWriter.println("Vendor:" + Build.MANUFACTURER);
        //4. 手机型号
        printWriter.println("Model:" + Build.MODEL);
        //5. CPU架构
        printWriter.println("CPU BAI:" + Build.CPU_ABI);
    }

    //6-上传异常信息到服务器
    private void uploadExceptionToServer(){
        //TODO: upload Exeception Msg to Server
    }
}
```

3、更改系统默认的Crash处理器

```
//1. 在Application的onCreate中设置
public class BaseApplication extends Application{
    @Override
    public void onCreate() {
        super.onCreate();
        //1. 为APP设置异常处理，此后程序才能获取未处理的异常
        CrashHandler crashHandler = CrashHandler.getInstance();
        crashHandler.init(this);
    }
}
```

```
//需要增加系统权限
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>
<uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE"/>
<uses-permission android:name="android.permission.MOUNT_UNMOUNT_FILESYSTEMS"/>
```