在读程序的时候，读到这样一段代码，其实刚开始我是不明白这个是干嘛用的，知道前几天读到一些博客，才明白这段代码的作用：当程序异常崩掉的时候，可以用这个进行相应的处理。

```
// 注册App异常崩溃处理器
Thread.setDefaultUncaughtExceptionHandler(AppException
        .getAppExceptionHandler(this));
```

在写程序时，肯定会碰到各种问题，在解决这些问题肯定要去看控制台打印的异常信息，根据控制台打印的异常信息来进行针对性的解决。

那么要解决程序运行在用户手机上崩溃的问题，必须得找到问题的原因。因此就要收集崩溃信息，也就是log日志。

Android程序Crash时我们可以做的操作：

1、将Crash信息存到本地，然后上传到服务器，根据上传的异常信息进行针对性的处理；

2、系统自带的Crash界面是很不友好的，我们可以自定义程序Crash后的界面，做的友好点；

关于以上2中操作方式，自己的见解：

1、应用中集成的友盟统计SDK，已经做了错误统计的功能，可以在友盟控制台直白的看到错误信息，当然也可以自己处理。

2、其实到这儿关注的重点已经是，程序Crash后，如何让它以一种更友好的方式消失

3、或者是，重启应用！

Git-osc的处理方式是：

当程序Crash掉的时候，弹出一个对话框，点击“发送崩溃报告”，将以邮件的形式把报告发送给开发者。

代码如下：

```
/**
 * 获取APP异常崩溃处理对象
 * @param context
 * @return
 */
public static AppException getAppExceptionHandler(Context context){
        return new AppException(context.getApplicationContext());
}
```

```
    @Override
    public void uncaughtException(Thread thread, Throwable ex) {
        if(!handleException(ex) && mDefaultHandler != null) {
            mDefaultHandler.uncaughtException(thread, ex);
        }

    }

/**
     * 自定义异常处理:收集错误信息&发送错误报告
     * @param ex
     * @return true:处理了该异常信息;否则返回false
     */
    private boolean handleException(final Throwable ex) {
        if(ex == null) {
            return false;
        }

        if(mContext == null) {
            return false;
        }
        final Context context = AppManager.getAppManager().currentActivity();

        final String crashReport = getCrashReport(context, ex);
        //显示异常信息&发送报告
        new Thread() {
            public void run() {
                Looper.prepare();
                // 上传错误信息到友盟的后台
                MobclickAgent.reportError(mContext, ex);
                UIHelper.sendAppCrashReport(context, crashReport);
                Looper.loop();
            }

        }.start();
        return true;
    }
```

```
/**
         * 获取APP崩溃异常报告
         * @param ex
         * @return
         */
        private String getCrashReport(Context context, Throwable ex) {
                PackageInfo pinfo =
((AppContext)context.getApplicationContext()).getPackageInfo();
                StringBuffer exceptionStr = new StringBuffer();
                exceptionStr.append("Version: "+pinfo.versionName+" ("+pinfo.versionCode+")\n");
                exceptionStr.append("API Level: "+android.os.Build.VERSION.SDK_INT + "\n");
                exceptionStr.append("Android: "+android.os.Build.VERSION.RELEASE
+" ("+android.os.Build.MODEL+")\n\n\n");
                exceptionStr.append("异常信息: \n");
                exceptionStr.append("Exception: "+ex.getMessage()+"\n\n\n");
                exceptionStr.append("堆栈信息: \n");
                StackTraceElement[] elements = ex.getStackTrace();
                for (int i = 0; i < elements.length; i++) {
                        exceptionStr.append(elements[i].toString()+"\n");
                }
                return exceptionStr.toString();
        }
```