**线程**

1.在一个应用程序中，主线程(Main Thread)通常用于接收用户的输入，以及将运算的结果反馈给用户，所以对于一些会产生阻塞的操作必须放置在worker thread中。但是work thread大多数情况下不能直接访问UI（progressbar除外）

2.安卓的消息传递机制

Handler的作用是将消息加入到消息队列（FIFO）,Looper的作用是将消息从队列中取出（队列中有消息对象，取出；没有，等待），然后调用Handler中的handlemessage方法来处理消息。

**从worker thread向main thread发送消息**

Handler handler=new xxHandler ();

Thread t=new xxThread();

t.start();

class xxHandler extends Handler{

void handlemessage(Message msg){

通过msg.what获取数据；

操作UI；

}

}

class xxThread extends Thread{ //worker thread

Message msg=handler.obtainMessage();

msg.what=xxx; //整型数据

message.obj=xxx;

handler.sendMessage(msg); //send the message to message queue, then looper uses handlemessage which belongs to the handler which sent the message.

}

**从Main thread向worker thread发送消息**

在worker thread中准备looper.prepare(),然后生成handler对象，然后调用loop方法循环取出消息，最后在main thread中调用handler.sendmessage发送消息。

Main thread:

Thread t=new xxThread();

t.start();

Message msg=handler.obtainMessage();

msg.what=xxx; //整型数据

handler.sendMessage(msg);

worker thread:

class xxThread extends Thread{ //worker thread

public void run(){

Looper.prepare();

Handler handler=new xxHandler (){

public void handlemessage(Message msg){

}

};

Looper.loop();

}

}

3. ThreadLocal

执行ThreadLocal对象的set（object obj）方法时将会向ThreadLocal中存入一个键值对（键是当前的线程，值是其中的数据），执行get()方法会取出当前线程值。

4.异步任务处理(以网络为例)

**ReadUrl( String url)**{

new AsyncTask<初始化传入数据的数据类型String，Void处理过程中返回值的类型，String返回结果的数据类型>{

**doInBackground**(初始化传入数据的数据类型String… arg0){ //执行后台的耗时操作，… 代表数组

URL url=new URL(arg0[0]); //arg0数组中第一个元素就是url‘

URLConnection connection= url.openConnection(); //获取当前的网络连接

InputStream in=connection.getInputStream();

InputStreamReader reader=new InputStreamReader(in);

BufferedReader bufferReader= new BufferedReader(reader);

String line;

StringBuilder stbuilder=new StringBuilder();

while((bufferReder.readLine())!=null){ //先赋值给line，在判断当前行是否有数据

stbuilder.append(line);

}

bufferReader.close();

reader.close();

in.close|();

return stbuilder.toString();

}.execute(url);

}

**回调方法**

**e.g.**

**onPostExecute**(String result){ //result就是doInBackground的返回值

}

}

注：涉及UI的操作只能在回调方法中执行，不能在doinback中。