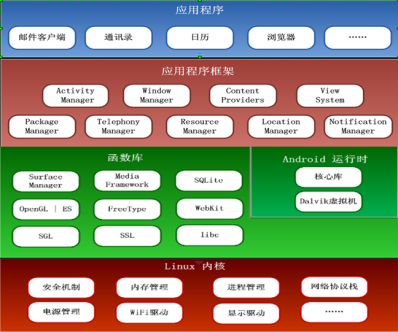
1、常见的智能手机操作系统：iPhoneOS、Android、Symbian、Windows Phone、BlackBerry OS等。  
2、2007年11月5日谷歌成立开放手机联盟OHA，共同开发 Android 移动系统

3、Android体系结构：



1、**简述Android体系结构，并说明各层次的作用**。

答：Android是基于Linux内核的软件平台和操作系统，共分为四层。第一层是Linux内核及Linux文件系统。Linux内核是硬件和其他软件堆层之间的一个抽象隔离层，提供由操作系统内核管理的底层基础功能，主要有安全机制、内存管理、进程管理、网络协议校和驱动程序等。第二层是Android函数库及运行时。第三层是应用程序框架层。提供了Android平台的管理功能和组件重用机制，包括Activity管理、资源管理、位置管理、通知消息管理、View系统和内容提供者等。第四层是应用程序层。提供了一系列核心应用程序，如打电话、浏览器、联系人、相册、地图和电子市场等。

1、Android开发环境所需软件：

JDK（Java开发包）、Eclipse（集中开发软件）、Android SDK（Android开发包）、ADT（Android开发工具）

2、Android SDK目录结构

dx.bat：将.class字节码文件转换成Android字节码.dex文件

android.jar：Android API核心包

adb.exe：Android调试桥，用于将Android手机连接到PC端

sqlite3.exe：数据库

AVD Manager.exe：模拟器管理程序

ddms.bat：Android系统的调试工具

1、Android工程的文件系统结构

源程序文件夹src，资源文件夹res、assets、gen，编译文件夹bin，包文件夹libs，工程配置清单文件AndroidManifest.xml，布局文件夹res/layout

2、Android四大组件

Activity、Service、BroadcastReceiver、ContentProvider

3、Android虚拟机Dalvik

1、在Android工程的文件系统结构中，请简述文件夹src、res/Layout以及文件AndroidManifest.xml的用途。  
答：源程序文件夹src，用于存放Java源程序文件。res/Layout用于存放扩展名为.xml的布局文件。工程配置清单文件AndroidManifest.xml用于配置Android系统运行前必须掌握的相关信息，例如：应用程序名称、组件注册、授权等。

2、简述Android的四大组件名称，并说明它们的功能。  
答：Android应用程序划分为四类核心组件，即Activity、Service、Broadcast Receiver和ContentProvider。其中，Activity是Android最重要的组件，负责用户界面的设计。Service是Android提供的无用户界面、长时间在后台运行的组件。BroadcastReceiver，即广播接收者，用来接收来自系统或其他应用程序的广播，并做出回应。ContentProvider，对共享数据进行查询、添加、删除和更新等操作。

3、试比较Activity和Service的异同点。  
答：相同点：都是Android四大核心组件之一。不同点：（1）功能。Activity是Android最重要的组件，负责用户界面的设计，在清单文件中使用标签<activity＞注册。而Service是Android提供的无用户界面、长时间在后台运行的组件，在清单文件中使用标签<service>注册。（2）生命周期。Activity生命周期的回调函数比Service多。（3）启动方式。Service能以绑定的方式启动。

4、试比较res文件夹和assets文件夹的异同点。

答：相同点：都是用于存放资源文件的文件夹。不同点：res文件夹内的资源文件最终被打包到编译后的Java文件中，res文件夹内的资源文件可以通过R资源类访问，res文件夹不支持深度的子目录；而assets中的资源文件不会打包到应用中，assets中的资源文件不能通过R资源类访问，assets支持任意深度的子目录。

5、试比较Dalvik和JVM的异同点。  
答：不同点：（1）结构。JVM是基于栈的，而 Dalvik 是基于寄存器实现的。因此 Dalvik 比JVM有更好的性能，只是硬件通用性差些。（2）运行。JVM运行的Java字节码，它从.class文件或Jar包中加载字节码然后执行； Dalvik 运行的是 dex 文件(Dalvik Executable)，生成APK时，Dx工具把所有.class文件编译成一个 dex 文件。（3）资源。相对于JVM来说， Dalvik 占用的CPU资源和内存空间都比较少。

4、阅读程序，回答问题。

<uses-permission android:name="android.permission.INTERNET”  
<application android:icon="@drawable/icon"

android:label="@string/app\_name">  
 < activity android:name=".IntentDemo"  
 android:label="@string/app\_name">

< intent-filter >

< action android:name="android.intent.action.MAIN" />

< category android:name="android.intent.category.LAUNCHER" />

</ intent-filter >

< activity >

< activity android:name=".NewActivity"

android:label="@string/app\_name" >

</activity>

</application>

1、常用控件

文本框控件TextView和EditText、显示图像控件ImageView、命令按钮控件Button和ImageButton、单选控件RadioButton、复选控件CheckBox、对话框控件AlertDialog、进度控件ProgressDialog、列表控件ListView、菜单Menu，等等。

2、常用布局

线性布局LinearLayout、相对布局RelativeLayout、帧布局，等。

3、用户界面事件

点击事件、 按键事件、触摸事件。

4、文件存储

内部存储、外部存储 、SharedPreferences存储

5、意图类Intent

用途：启动组件、组件之间传递数据。  
 分类：显式启动、隐式启动。

1、在用户界面设计中，常用的控件有哪些? 请举例5个并简述他们的用途。  
答：文本框控件TextView和EditText，用来输入和编辑字符的控件。显示图像控件ImageView，用来显示图像。命令按钮控件Button，用来定义命令按钮，当用户单击按钮时会有相应的动作。RadioButton是单选按钮控件。复选控件CheckBox是复选按钮控件。

1、在用户界面设计中，常用的控件有哪些? 请举例5个并简述他们的用途。  
答：文本框控件TextView和EditText，用来输入和编辑字符的控件。显示图像控件ImageView，用来显示图像。命令按钮控件Button，用来定义命令按钮，当用户单击按钮时会有相应的动作。RadioButton是单选按钮控件。复选控件CheckBox是复选按钮控件。

2、布局文件：

< LinearLayout

android:layout\_width="fill\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="vertical" >  
 < TextView android:id="@+id/label"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="用户名:" />  
 < EditText android:id="@+id/entry"

android:layout\_height="wrap\_content"

android:layout\_width="fill\_parent" />

< Button android:id="@+id/ok"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="确认" />

</ LinearLayout >

3、按钮点击事件：

Button button = (Button)findViewById(R.id.btn);  
 button.setOnClickListener(new OnClickListener(){

public void onClick(View view){

Intent intent = new Intent(IntentDemoActivity.this, NewActivity.class );

startActivity( intent );

}

}

4、SharedPreferences文件存储。

public static final String PREFERENCE\_NAME = "SaveSetting";

public static int MODE = Context.MODE\_WORLD\_READABLE +

Context.MODE\_WORLD\_WRITEABLE;

private void saveSharedPreferences(){

SharedPreferences sharedPreferences =

getSharedPreferences(PREFERENCE\_NAME, MODE);

SharedPreferences.Editor editor = sharedPreferences.edit();

editor.putString("Name", nameText.getText().toString());

editor.putInt("Age", Integer.parseInt(ageText.getText().toString()));

editor.putFloat("Height", Float.parseFloat(heightText.getText().toString()));

editor.commit();

}

1、手机基本功能经有：打电话、发短信、媒体播放、录音、拍照、二维码扫描

2、权限注册：

<uses-permission android:name="android.permission.CALL\_ PHONE" />

1、播放音频文件：

public class MainActivity extends Activity {

private MediaPlayer mp;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

mp=MediaPlayer.create(this,R.raw.black);

mp.start();

}

@Override

public void finish(){

mp.stop();

super.finish();

}

}

1、服务Service：

服务的显式启动和隐式启动。

绑定服务。

本地服务与远程服务。接口定义语言AIDL

2、广播BroadCast和广播接受者BroadCastReceiver

1、绑定服务：

public class MyAudioService extends Service {

public class PlayBinder extends Binder{

public void MyMethod(){

String path="sdcard/Music/white.mp3";

MediaPlayer mediaplayer=new MediaPlayer();

try {

mediaplayer.setDataSource(path);

mediaplayer.prepare();

mediaplayer.start(); }

catch (IOException e) { e.printStackTrace(); }

}

}

@Override

public IBinder onBind(Intent arg0) {

return new PlayBinder();

}

}

1、绑定服务：

public class MainActivity extends Activity {

private ServiceConnection conn = new ServiceConnection() {

@Override

public void onServiceConnected(ComponentName name, IBinder service) {

PlayBinder myAudioBinder = (MyAudioService.PlayBinder) service;

myAudioBinder. MyMethod( );

}

};

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

Intent intent = new Intent(this, MyAudioService.class);

bindService(intent, conn, BIND\_AUTO\_CREATE);

}

}

1、绑定服务：

public class MainActivity extends Activity {

private ServiceConnection conn = new ServiceConnection() {

@Override

public void onServiceConnected(ComponentName name, IBinder service) {

PlayBinder myAudioBinder = (MyAudioService.PlayBinder) service;

myAudioBinder. MyMethod( );

}

};

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

Intent intent = new Intent(this, MyAudioService.class);

bindService(intent, conn, BIND\_AUTO\_CREATE);

}

}

2、广播接收者：

public class MyReceiver extends BroadcastReceiver {

@Override

public void onReceiver(Context context, Intent intent) {

MediaPlayer.create(context, R.raw.tick).start();

}

}

<receiver

android:name="com.example.audio\_service.MyReceiver"

android:enabled="true"

android:exported="true" >

<intent-filter>

<action android:name="android.provider.Telephony.SMS\_RECEIVED" />

</intent-filter>

</receiver>

3、发送广播：

Intent intent=new Intent("com.example.audio\_service.MyReceiver");

Bundle bundle=new Bundle();

bundle.putString("msg","测试信息");

intent.putExtras(bundle);

sendBroadcast(intent);

4、BroadcastReceiver注册：静态注册、动态注册

IntentFilter filter = new IntentFilter();

filter.addAction("wustzz.helloandroid.mybroadcast");

BroadcastReceive r = new HelloReceiver();

registerReceiver( r , filter);

3、发送广播：

Intent intent=new Intent("com.example.audio\_service.MyReceiver");

Bundle bundle=new Bundle();

bundle.putString("msg","测试信息");

intent.putExtras(bundle);

sendBroadcast(intent);

4、BroadcastReceiver注册：静态注册、动态注册

IntentFilter filter = new IntentFilter();

filter.addAction("wustzz.helloandroid.mybroadcast");

BroadcastReceive r = new HelloReceiver();

registerReceiver( r , filter);

3、发送广播：

Intent intent=new Intent("com.example.audio\_service.MyReceiver");

Bundle bundle=new Bundle();

bundle.putString("msg","测试信息");

intent.putExtras(bundle);

sendBroadcast(intent);

4、BroadcastReceiver注册：静态注册、动态注册

IntentFilter filter = new IntentFilter();

filter.addAction("wustzz.helloandroid.mybroadcast");

BroadcastReceive r = new HelloReceiver();

registerReceiver( r , filter);

1、SQLite数据库，存储在：/data/data/应用程序包名/databases/

2、使用抽象类SQLiteOpenHelper创建、打开或更新数据库。

3、数据库的四种操作：

db.execSQL(); 插入insert、更新update、删除delete

db.rawQuery(); 查询select

1、创建和打开数据库

class MyDbOpenHelper extends SQLiteOpenHelper{

public MyDbOpenHelper(Context context) {

super(context, "test.db", null,1);

}

@Override

public void onCreate(SQLiteDatabase db) {

db.execSQL("CREATE TABLE person(id INTEGER PRIMARY KEY

+AUTOINCREMENT, name VARCHAR(20))");

db.execSQL("insert into person values(null,‘Chen’)");

}

@Override

public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {

db.execSQL("ALTER TABLE person ADD tel INTEGER");

db.execSQL("update person set tel=15527643858 where name=‘Chen’");

}

}

MyDbOpenHelper helper=new MyDbOpenHelper(this);

helper.getWritableDatabase();

2、查询数据

DBOpenHelper helper = new DBOpenHelper(getApplicationContext(), "test.db", null,DB\_VERSION);

SQLiteDatabase db=helper.getWritableDatabase();

Cursor cursor = db.rawQuery( "SELECT \* FROM person where age>?", new String[]{"10"} );

while (cursor.moveToNext()) {

int id = cursor.getInt(cursor.getColumnIndex("id") );

String name = cursor.getString(cursor.getColumnIndex("name") );

int age = cursor.getInt(cursor.getColumnIndex("age") );

}

cursor.close();

db.close();

1、ContentProvider组件：内容提供者。

2、ContentResolver类：内容解析器。

3、URI: 统一资源标识符

URL是标识资源的物理位置；URI则是标识资源的逻辑位置

基本格式： content://<authority>/<data\_path>/<id>

手机联系人：ContactsContract.Contacts.CONTENT\_URI

2、读取手机联系人

Uri nameuri = ContactsContract.Contacts.CONTENT\_URI;

Uri phoneuri = ContactsContract.CommonDataKinds.Phone.CONTENT\_URI;

ContentResolver resolver = getContentResolver();

Cursor cursor = resolver.query(nameuri, null, null, null, null);

while(cursor.moveToNext()){

String contactId = cursor.getString(cursor.getColumnIndex("\_id"));

String name = cursor.getString(cursor.getColumnIndex("display\_name")nameIndex);

Cursor phone = resolver.query(phoneuri, null, "contact\_id = ?", new String[]{ contactId }, null);

while( phone.moveToNext() ){

String strPhoneNumber = phone.getString(phone.getColumnIndex("data1"));

}

phone.close();

}

cursor.close();

Android通信及其应用

1、WiFi通信

2、蓝牙通信Bluetooth

3、近场通信NFCI

位置服务与地图应用开发

1、位置服务

GPS定位、WIFI定位、基站定位

2、地图应用开发

1、使用GPS定位功能，获取当前位置的经纬度

LocationManager lm =(LocationManager) this.getSystemService(Context.LOCATION\_SERVICE);

if(!lm.isProviderEnabled(Lm.GPS\_PROVIDER)){

Intent myintent = new Intent(Settings.ACTION\_LOCATION\_SOURCE\_SETTINGS);

startActivity(myintent); }

lm.requestLocationUpdates(LocationManager.GPS\_PROVIDER, 0, 0, new MyLocationListener());

class MyLocationListener implements LocationListener{

@Override

public void onLocationChanged(Location location) {

tv\_01.setText("您当前位置的经度为："+location.getLongitude());

tv\_02.setText("您当前位置的纬度为："+location.getLatitude());

}

Android网络编程1、HTTP网络编程

Java网络编程：HttpURLConnection编程

Apache网络编程：HttpClient编程

Android模拟器：127.0.0.1，本机的Web服务器：http://10.0.2.2

2、Socket网络编程

1、HttpClient 开发手机客户端

HttpClient client = new DefaultHttpClient();

HttpPost post = new HttpPost( "http://webservice.webxml.com.cn");

List<NameValuePair> params = new ArrayList<NameValuePair>();

params.add(new BasicNameValuePair("mobileCode", phoneSec));

params.add(new BasicNameValuePair("userId", ""));

try {

post.setEntity(new UrlEncodedFormEntity(params, HTTP.UTF\_8));

HttpResponse response = client.execute(post);

String result="";

if (response.getStatusLine().getStatusCode() == HttpStatus.SC\_OK) {

result = EntityUtils.toString( response.getEntity(),"utf-8");

}

resultView.setText(filterHtml(result));

} catch (Exception e) {

e.printStackTrace();

}