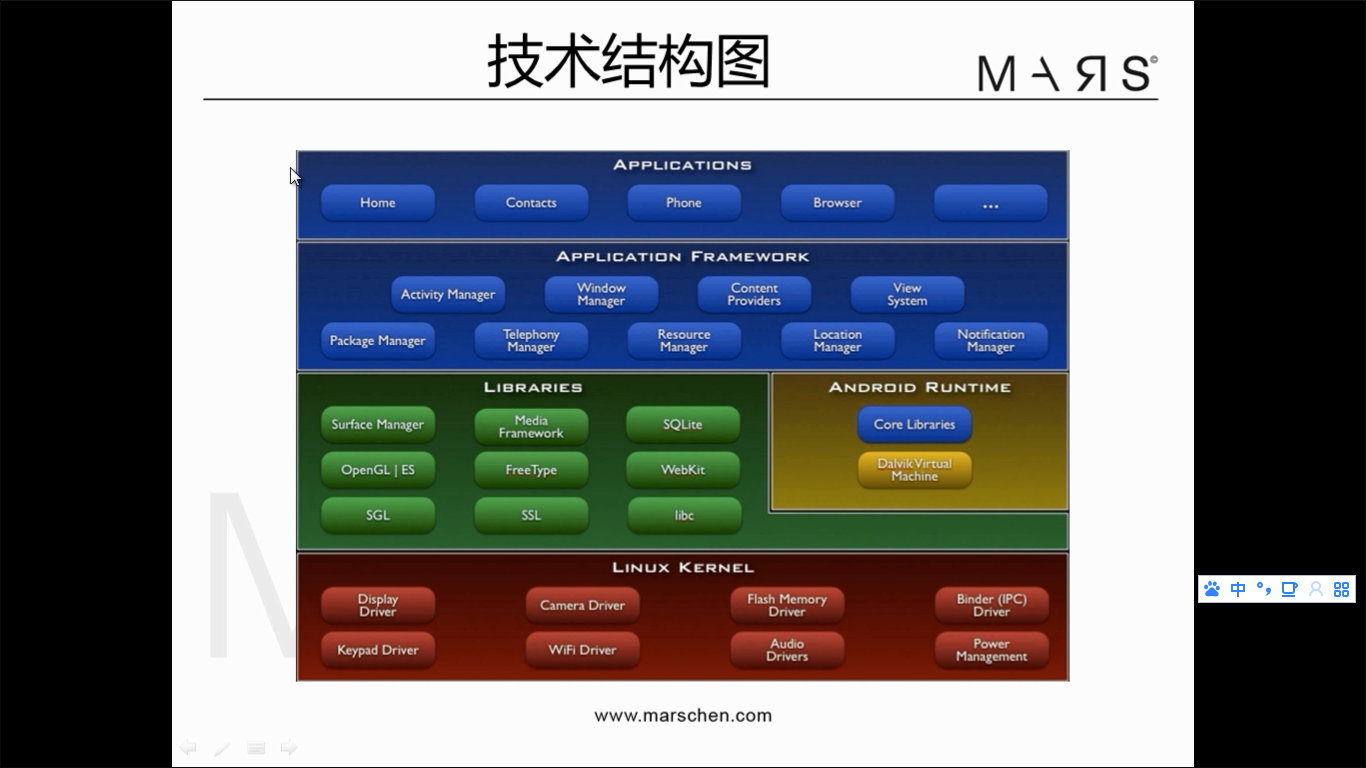
**安卓技术结构图**



Linux 核心、库（Core Libraries 相当于JDK， DVM 是虚拟机） 、应用程序框架、应用程序

**基于组件的开发**

1. activity UI交互

2. service 后台完成耗时较长的任务，例如网络连接

3. content provider 负责对外提供数据

4. broad cast receiver 监听系统行为

**Activity启动的基本流程**

AndroidManifest.xml是安卓的主配置文件，决定启动默认的Activity.

操作系统首先生成MainActivity，调用onCreate ()，其中读取布局文件。

onCreate ()中的setContentView()接受布局文件的id，让activity显示该布局文件下的内容。

TextView textView= (TextView) findViewById (R.id.textView)可以构建代表控件的对象,从而实现通过java代码修改控件的属性。

**View**

View代表控件，是所有控件类（文本，按钮，多选等等）的父类。这也是为什么(TextView) findViewById (R.id.textView)需要向下转型，因为它返回的是view类型。

**监听器**

监听器是一种对象，监控着控件对象状态的变化。

每一个控件都可以绑定不同的监听器，以此响应不同的事件。绑定监听器的流程（以按钮为例）：

1.布局文件中定义控件

2.获取button对象

3.绑定监听器

button. setOnClickListener(new OnClickListener() {重写public void onClick(){需要执行的内容} } );

**距离单位**

px 像素

dpi （dots per inch）=根号下（长像素平方+宽像素平方）/对角线英寸

dp = dip device independent pixels

dp\*（dpi）/160=px 在dpi为160的屏幕上 1dp=1px

sp 用来指定字体大小，随着手机字体大小的修改而改变

内边距padding 内容和控件边距

外边距layout\_margin 控件和控件的距离



**多选按钮控件**

方法一：new onClickListener(this);

onClick ( View view){

CheckBox box= (CheckBox)view;

Switch (box.getId ){

case

case

case

}

box.isChecked()== true; //选中

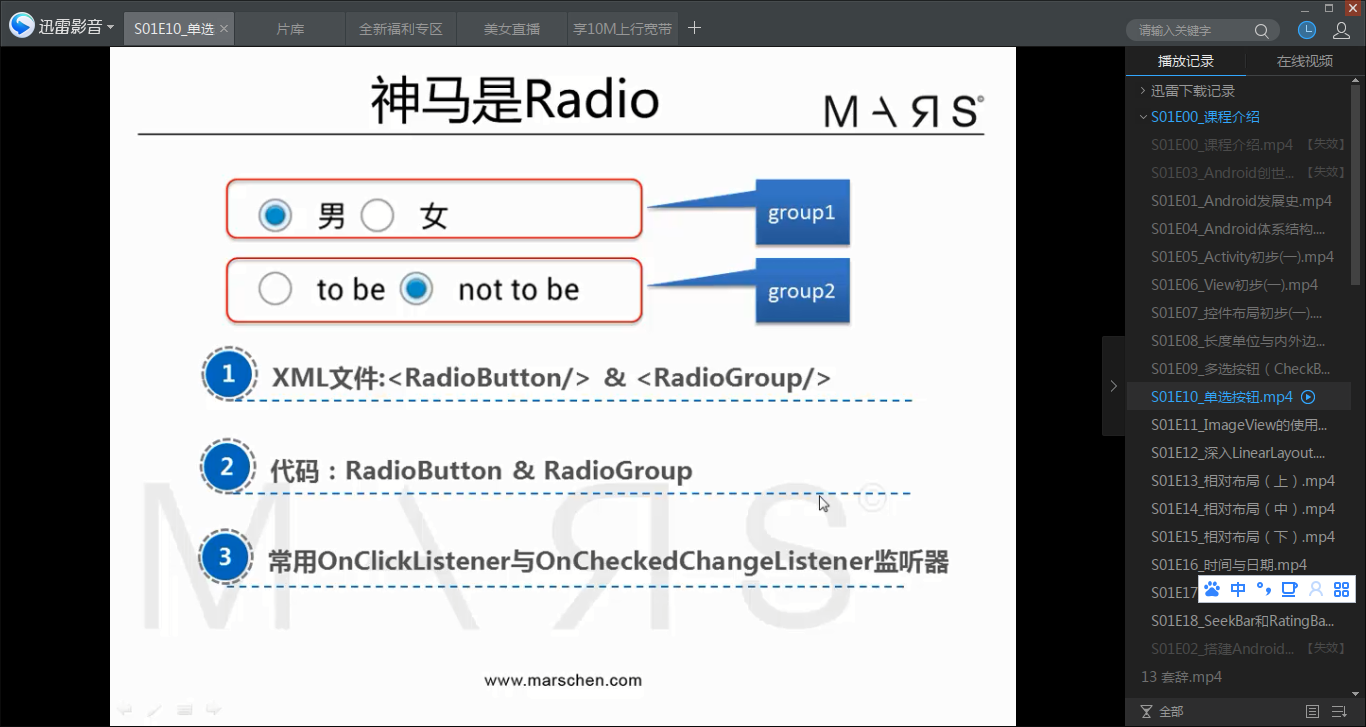
}

方法二：new OnCheckedChangeListener(this);

onCheckeChange (CompoundButton buttonView, Boolean isChecked){

}

**单选按钮控件**



注：<RadioButton/>是<RadioGroup/>的子标签

**ImageView**

<ImageView

android: src=”drawable/xxx”/>

ScaleType用来定义图片拉伸的类型

android: scaleType=” ” 或者.setScaleType (ScaleType. );

fitCenter: 等比例变化后放在ImageView 的中间（大的缩小的放大）；

fitStart: 等比例变化后放在ImageView 靠上显示；

fitEnd: 等比例变化后放在ImageView 靠下显示；

center: 截取大图的中间部分；

centerInside: 等比例缩小大图置于ImageView的中央，小图不变；

centerCrop: 缩小或放大图片，使其填满ImageView。

**LinearLayout**

layout\_weight平分的是父控件的**剩余空间**，控件的比例还与控件本身占据的空间有关，如果想真正意义上的实现通过layout\_weight实现一定比例的瓜分父控件，可以把子控件的宽（或者高，视情况）设置成0dp

**RelativeLayout**

控件默认在左上角

属性android:xxxxx=”id/xxxxx”

子控件和父控件之间：true or false

layout\_alignBaseline=.... 控件基准线对齐

layout\_alignParentLeft/Right/Top/Bottom=”true /false”

layout\_centerInParent/Horizontal/Vertical

layout\_align(Parent)Start/End

**时间与日期控件**

**TimePicker**

new onTimeChangedListener (this);

onTimeChanged( TimePicker view, int hourofDate, int minute){}

timepicker对象.setIs24HourView(true/false);

timepicker对象.getCurrentHour/Minute();

**DatePicker**

Datepicker对象.getYear/Month/DayofMonth(); **注：月份从0开始**

Datepicker对象.update（int, int, int）;设置默认显示时间

**AnologClock**

**进度条ProgressBar**

风格style="?android:attr/progressBarStyleHorizontal"水平进度条

默认风格有属性：Small/Large/Inverse(对比度高)/Small.Innverse/Large.Inverse

进度条的主要属性：android: max（最大值）/progress（当前进度）/SecondaryProgress(类似看视频时的缓冲进度条) java代码中用set更改。

boolean flag=progressBar.isIndeterminate();判断进度条是不是不确定的状态（圆圈型的）

progressBar.increment（Secondary）ProgressBy(int );进度加。。

**SeekBar进度可以拖动**

监听器OnSeekBarChangeListener类中有方法onProgressChnaged(...)最后一个参数是布尔型，用来测试进度是不是有用户直接拖拽引起的

onStart(Stop)TrackingTouch(..)开始或停止拖拽

**RatingBar**

属性

星星数 numStars

当前等级progress

每次的进度stepSize

setRating

OnRatingBarChangedListener监听器

**Activity的生命周期**

1.定义多个activity

定义类继承，复写oncreate，Manifest注册。

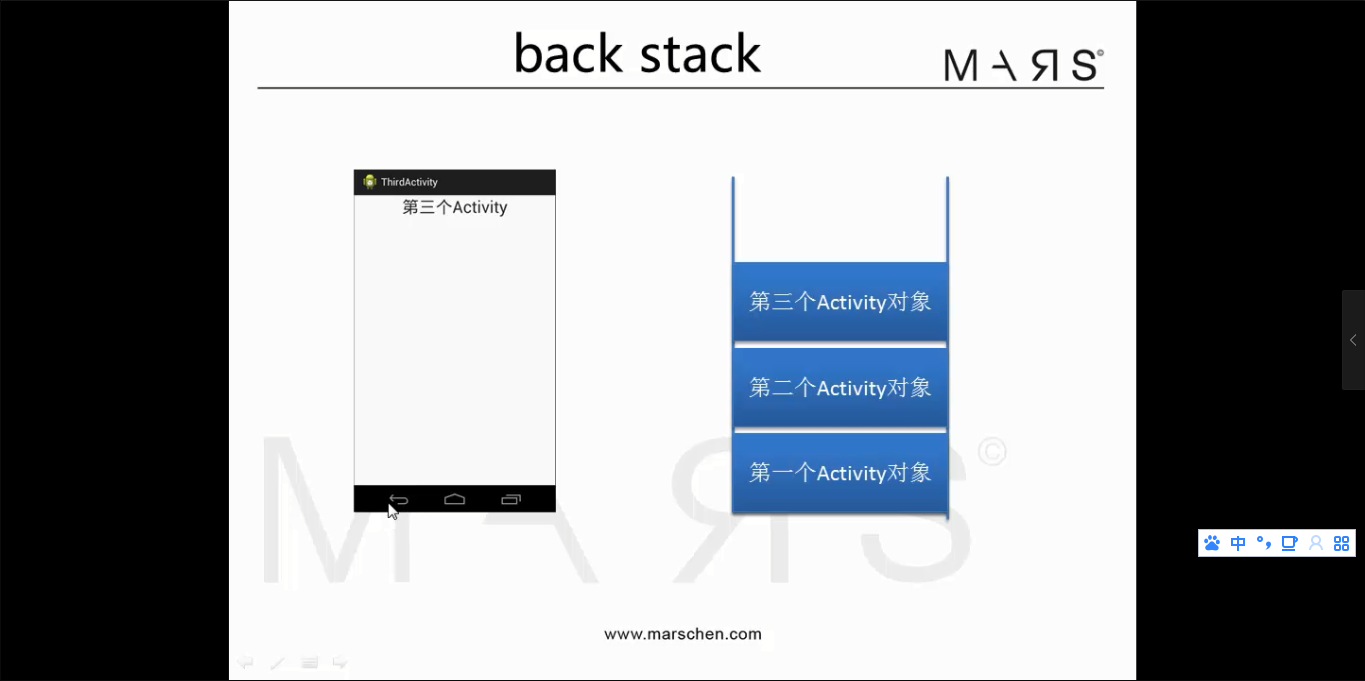
2. Activity跳转

Intent intent=new Intent();

Intent.setClass(原来的activity.this,下一个activity.class);

**startActivity(intent);**

3. back stack

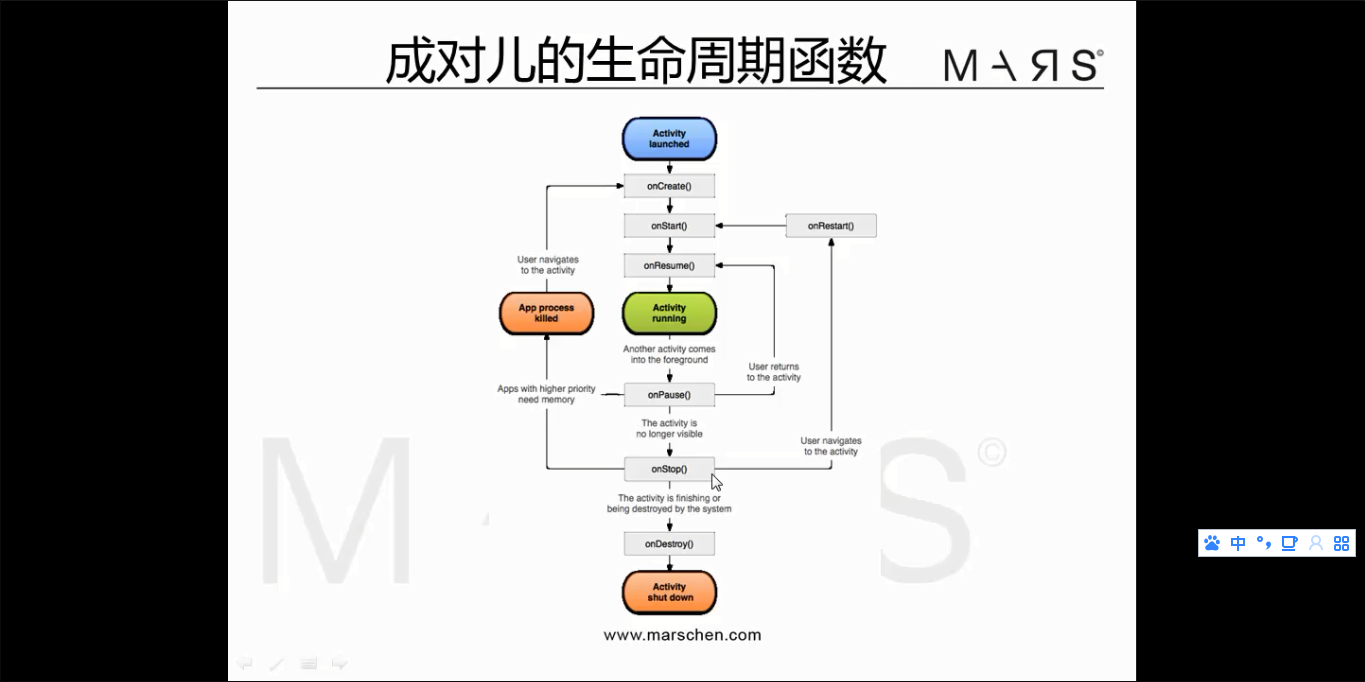
系统只显示栈顶的Activity

4.生命周期函数（**由操作系统调用而非程序员**）



5.





**Intent组件**

Activity 1:

Intent intent=new Intent();

Intent.setClass(原来的activity.this,下一个activity.class);

Intent.putExtra(“packagename.xxx”, value);

**startActivity(intent);**

Activity 2:

Intent intent=getIntent(); //获取启动Activity 2的Intent

DateType xxx=intent.get(DataType)Extra(“packagename.xxx”, defaultvalue);

**Context**访问当前包的资源（getResources、getAssets）和启动其他组件（Activity、Service、Broadcast）以及得到各种服务（getSystemService）。

**Application** 对于同一个应用程序是唯一的，在Manifest文件中进行配置，可以实现数据共享。Application的onCreate()函数会比Activity的OnCreate()先执行。生命周期最长。

**Service**对于同一个应用程序是唯一的