

## Activity

onCreate

界面的初始化 setContentView findViewById

数据的初始化

文件/数据库/网络/其它的组件/sp

开启其它的组件(开启服务,注册广播接收者) 动态注册广播接收者(onCreate注册 onDestroy注销  
onstart注册的onStop注销 onResume注册 onPause注销)

onstart

onResume 刷新界面

onPause 停止刷新界面

onStop

onDestroy 释放资源 数据库的连接 cursor sqlitedatabase bind的服务 广播接收者注销 退出线程

onRestart

startActivityForResult(intent,requestCode);

onActivityResult(int requestCode, int resultCode, Intent data);

返回数据的activity

setResult(resultCode, intent data);

finish();

launch mode

standard

singleTop

singleTask

singleInstance

Service startService 在某些情况下可以提高应用的进程优先级

startService(intent ) onCreate onStartCommand(Intent intent)

bindService onCreate onBind->Binder ondestory()

LRU 刚用过的最后回收 最早用过的最先回收

BroadcastRecevier

onReceiver

sendBroadcast 无序广播 不可以中断 不可以修改广播的内容

sendOrderBroadcast 可以中断 可以修改广播的内容

contentprovider/ contentresolver

authorities

exported

UriMatcher addURI()

URI content://authorities/子路径

文件的存储

访问网络 获取&提交数据

访问网络必须在子线程

修改UI必须在主线程 子线程获取数据想修改界面 必须使用handler runOnUiThread

UI相关

listview

TextView

EditText

Button

Imageview

ImageButton

RadioGroup

CheckBox

progressbar

seekbar

AlertDialog

progressDialog

AutocompleteTextView

Menu

Surfaceview

VideoView

多媒体

图片处理

音频视频的播放

动画

JNI

java native interface

native 本地 android 的本地语言 c/c++

native language

操作系统用哪种语言开发的 那么对于这个操作系统这种语言就是本地语言

jni的作用 实现java和本地语言的相互调用

android平台 jni的作用 java和c/c++的相互调用

①c可以操作硬件

使用jni可以使java代码操作硬件

②c直接运行在底层 c->机器码 java cocos2d 视频解码 人脸识别

通过jni把需要效率的逻辑放到c去实现

③c 1972 开源的项目 ffmpeg opencv

使用jni可以java代码 访问c的优秀开源项目

④ 处于安全的考虑 对安全要求比较的高的逻辑 通过jni放到C处理

c 能看懂 会调用

1 helloworld

```
1. #include<stdlib.h> 相当于java的import .h是c的头文件
2. #include<stdio.h> std standard library stdlib.h标准函数库 stdio 标准输入输出流
3. // stdlib.h stdio.h跟java.lang左右类似
4. main(){ //public static void main(String[] args)
5.     printf("hello world!\n"); //向控制台输出一句话 system.print() \n换行
6.     system("calc");
7.     system("pause"); //调用当前系统的命令 pause 让控制台停住 相当于在cmd控制台
   上敲 pause命令
8. }
```

2 c的基本数据类型

java的基本数据类型

boolean 0 非0表示 false true

byte 1

char 2 char 1个字节

short 2 short

int 4 int

long 8 long 4个字节

float 4 float

double 8 double

signed unsigned只能用来修饰整形变量 char short int long

signed 有符号 默认情况都是有符号的 可以表示负数 最大值相对小

unsigned 无符号 如果加上了unsigned修饰符 不可以表示负数 最大值相对大

### 3 输出函数

#### printf

```
1.  #include<stdlib.h>
2.  #include<stdio.h>
3.  /*
4.  %d - int
5.  %ld - long int
6.  %lld - long long
7.  %hd - 短整型
8.  %c - char
9.  %f - float
10. %lf - double
11. %u - 无符号数
12. %x - 十六进制输出 int 或者long int 或者short int
13. %o - 八进制输出
14. %s - 字符串
15.
16. 12345678 101111100 01100001 01001110
17. 24910      1100001 01001110
18. */
19.
20. main(){
21.     char c = 'c';
22.     short s = 1234;
23.     int i = 12345678;
24.     long l = 1234567890;
25.     float f = 3.14;
26.     double d = 3.1415926;
27.     printf("c = %c\n",c);
28.     printf("s = %hd\n",s);
29.     printf("i = %d\n",i);
30.     printf("l = %ld\n",l);
31.     printf("f = %.2f\n",f); //printf输出小数的时候默认保留6位有效数字 可以通过%.
    2f来指定有效数字的位数
32.     printf("d = %.7lf\n",d);
33.
34.     printf("i八进制%#o\n",i);
35.     printf("i16进制%#x\n",i);// %#x %#o 输出带前缀的八进制和十六进制数
36.     //c定义数组 []必须放在变量名后面
37.     char str[] = {'a','b','c','d','\0'}; // c的数组不检测越界 \0 就是字符串结束
    的标志
38.     char str1[] = "hello world! 你好世界";//
39.     printf("%s\n",str1);
40.     system("pause");
41. }
```

### 4.输入函数

```

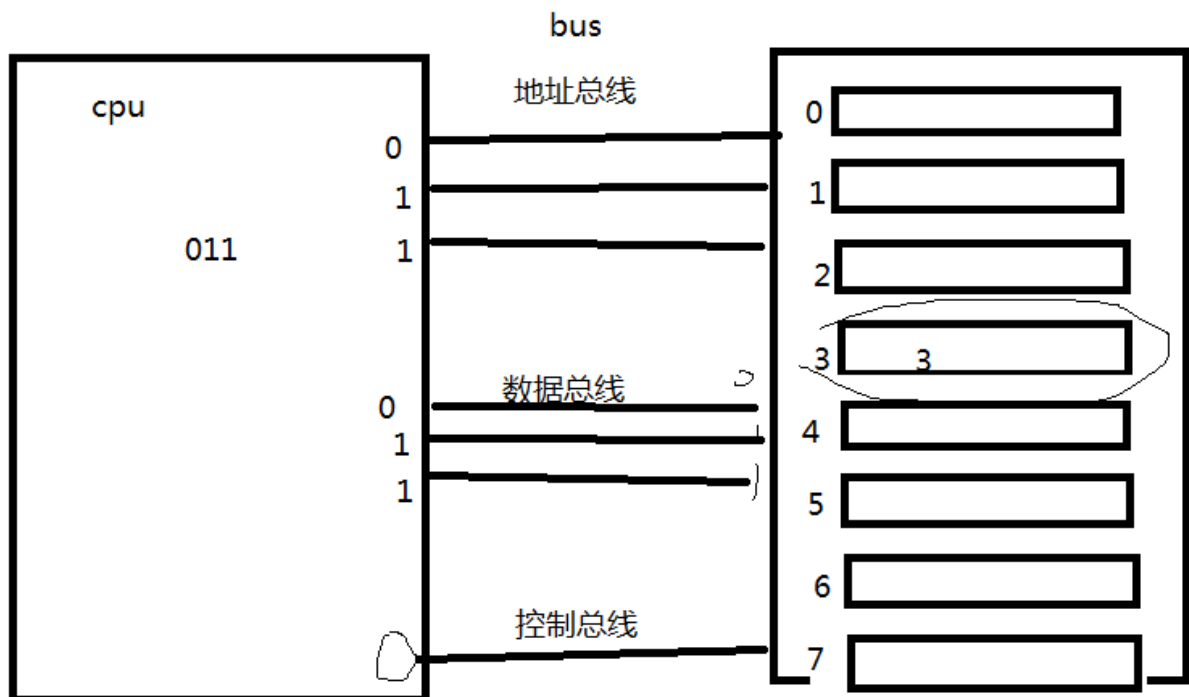
1. #include<stdlib.h>
2. #include<stdio.h>
3. //scanf
4. //c的数组不检测越界 需要程序员注意
5. main(){
6.     printf("请输入班级的人数\n");
7.     //定义一个变量 count来接收用户的输入
8.     int count;
9.     printf("count的地址%d\n",&count);
10.    scanf("%d",&count); //取地址符号
11.    printf("班级的人数为%d\n",count);
12.    printf("请输入班级的名字\n");
13.    //定义字符数组 用来接收班级的名字
14.    char name[18];
15.    scanf("%s",&name);
16.    printf("name的地址%d\n",&name);
17.    printf("班级的名字是:%s,班级的人数为%d\n",name,count);
18.    system("pause");
19. }

```

## 5. 内存地址

内存的编号 没有编号的内存是没有办法使用的

内存的最小单位是 byte 8位的二进制数



32位win7 最大支持多大内存 4G 32位的操作系统做多生成4g编号  
地址线32位  $2^{32}$

## 6. 指针 ☆☆☆☆☆☆

pointer 保存内存地址 保存内存的编号

7指针的常见错误

8 指针的练习

9指针和数组的关系

10多级指针

学生学号管理系统

11 结构体☆☆☆☆☆

结构体的指针

12 联合体(共用体)

13 枚举

14 自定义类型

学生学号管理系统

11 结构体☆☆☆☆☆

结构体的指针