

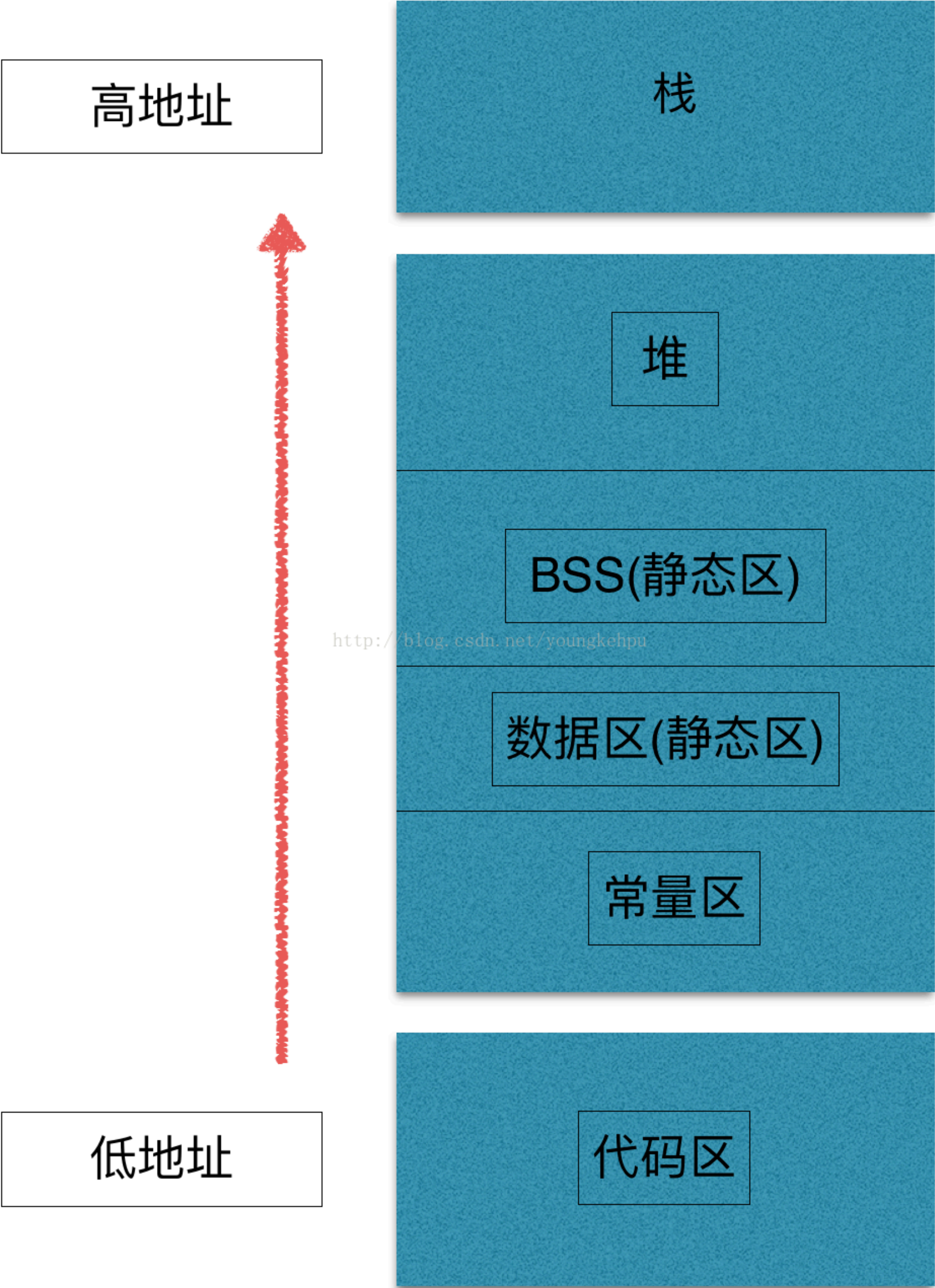
原

iOS 内存分配（栈和堆）

2017年02月21日 14:57:18

Janko_

阅读数 1735



代码示例：

import "JKViewController.h"

@interface JKViewController ()

@end

@implementation JKViewController

int num = 1;//数据区 （全局区/静态区）

NSString str;//BSS区 （全局区/静态区）

static NSString str2 = @"string";//静态区 （静态初始化区/全局区）

- (void)viewDidLoad {
[super viewDidLoad];
int age;//栈
NSString name = @"xiaoming";//栈
NSString number = @"123";//123在常量区， number在栈上
NSMutableArray *array = [NSMutableArray arrayWithCapacity:1];//分配而来的8字节的区域就在堆中(相当于alloc分配内存)， array在栈中， 指向堆区的地址

}

/

方法中的num1和num2都在栈中， 返回值num也暂存在栈中！

- (int)changenum1:(int)num1 num2:(int)num2{
int num = num1 + num2;
return num;
}
@end

1、代码区：用来存放函数的二进制代码，在运行时要防止被非法修改，只允许读取不允许操作

2、常量区：存储常量

3、静态区：（比如static声明的变量）

数据区：存放程序静态分配的变量和全局变量

BSS：包含了程序中未初始化全局变量

4、堆：由程序员分配和释放，存放进程运行中被动态分配的内存段，可大可小，根据存储的多少来调节大小。调用alloc等分配内存时，新分配的内存就被动态添加到堆上（堆被扩张）；当利用realse释放内存时，被释放的内存从堆中被踢除（堆被缩减），因为我们现在iOS基本都使用ARC来管理对象，所以不用我们程序员管理，但是我们要知道这个对象存储的位置

5、栈：栈是由编译器自动分配并释放，用来存放函数括弧“{}”中定义的变量。当函数被调用时，函数带有的参数也会被压入发起调用的进程栈中，待到调用结束，函数的返回值也回被存放回栈中。由于栈的先进先出特点，所以栈特别方便用来保存/恢复调用现场。可以把栈看成一个临时数据寄存、交换的内存区

以上中堆和静态区以及常量区都是连续的，栈和代码区都是独立存放的，栈是向低地址扩展的数据结构，是一块连续的内存的区域。堆是向高地址扩展的数据结构，是不连续的内存区域。堆和栈不会碰到一起，之间间隔很大，绝少有机会能碰到一起，况且堆是链表方式存储！



硕士毕业论文两天写完

硕士毕业论文