
修炼内功

必须要搞清楚嵌入式内存管理！

创客学院 正道老师

1 / 嵌入式内存管理需求在哪？

2 / 常见的内存管理方案

3 / freeRTOS内存分配原理

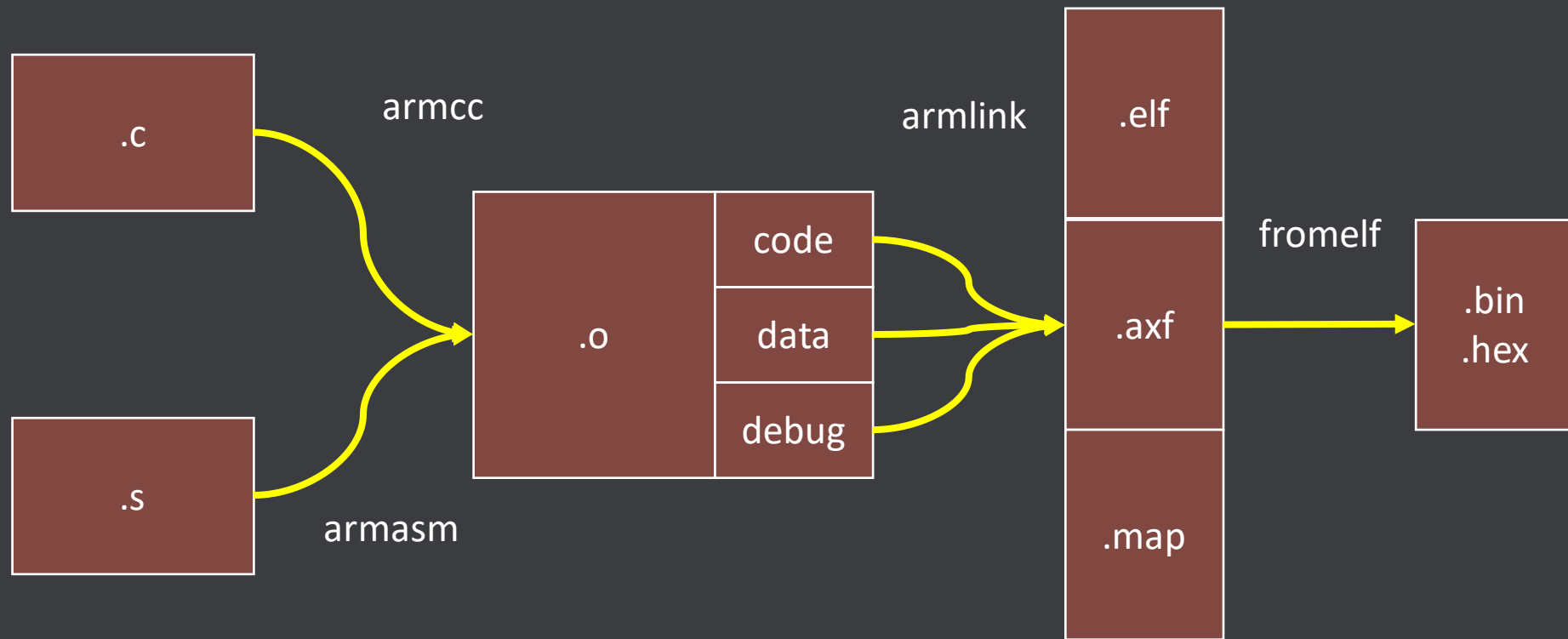
C程序编译过程

Source code

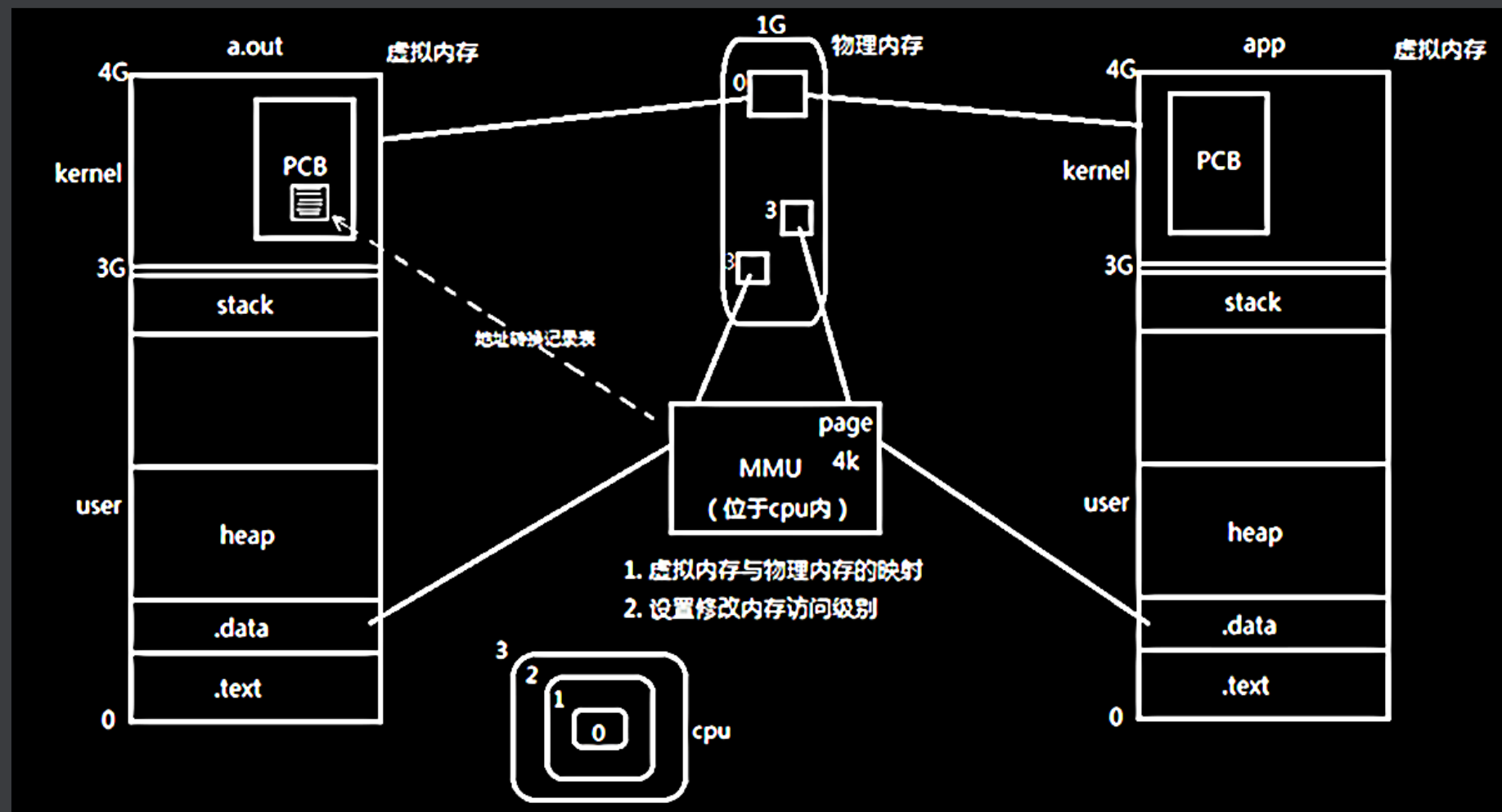
Object code

Image

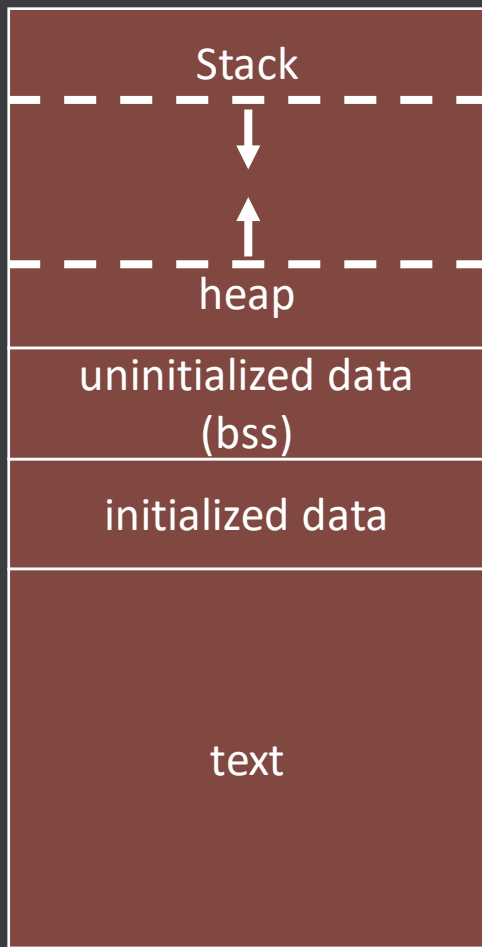
Flash format



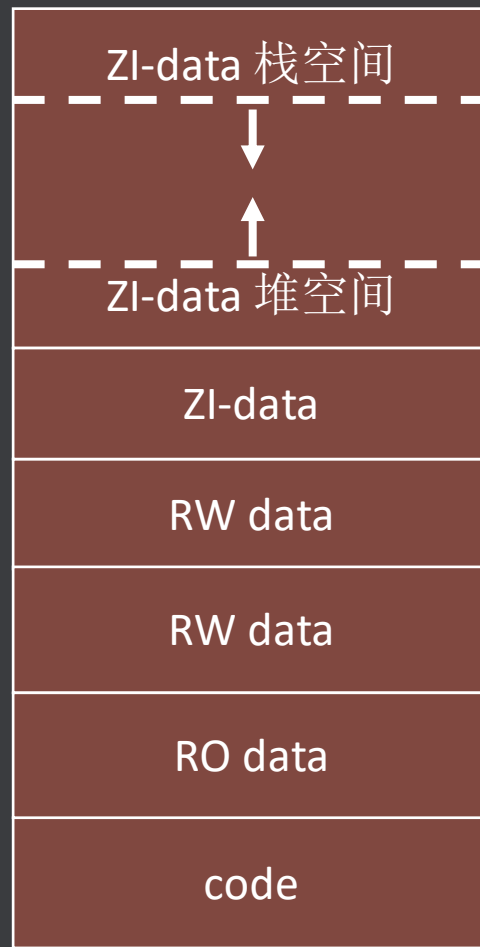
进程程序组成



裸机程序组成



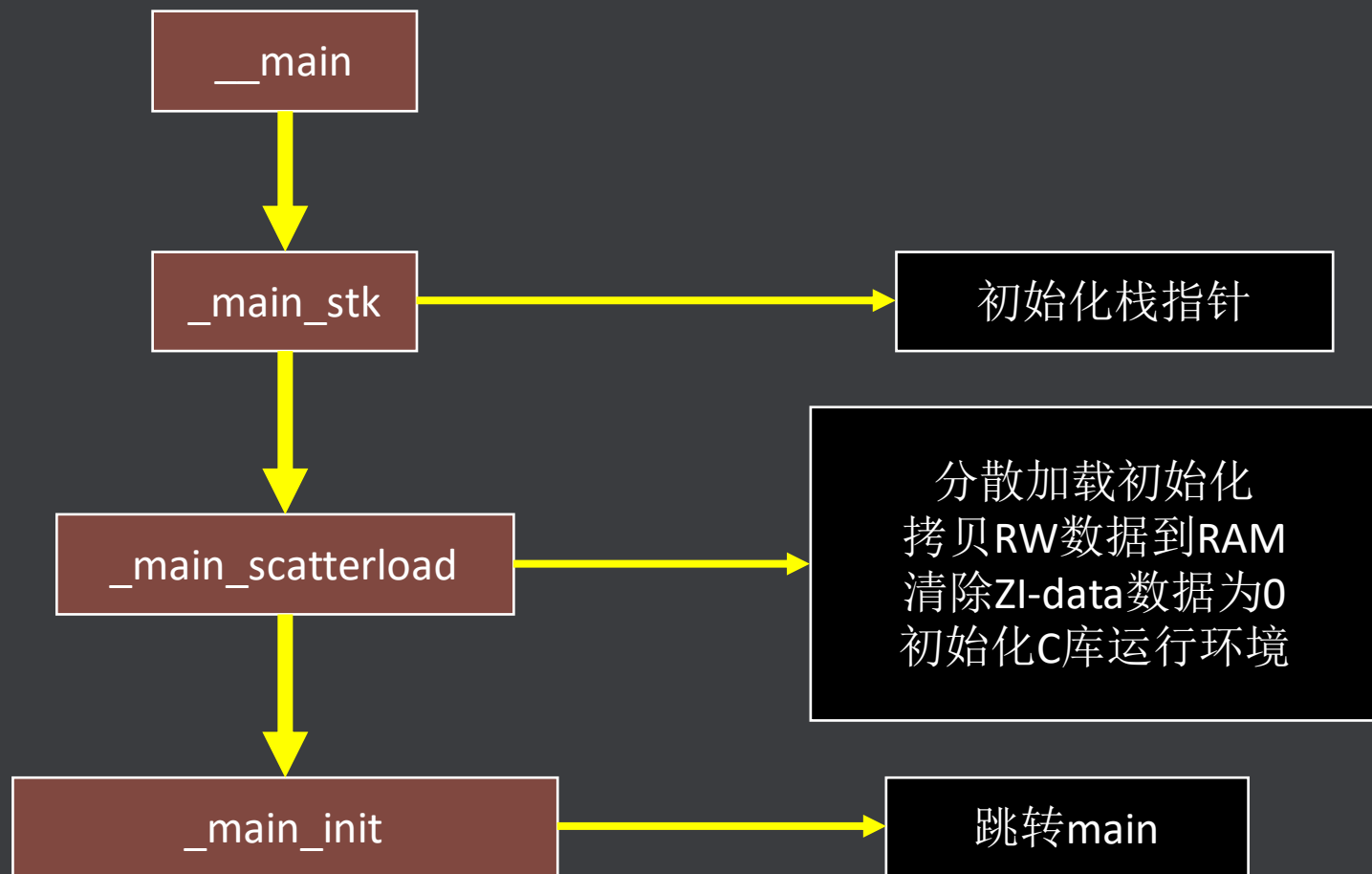
C程序组成



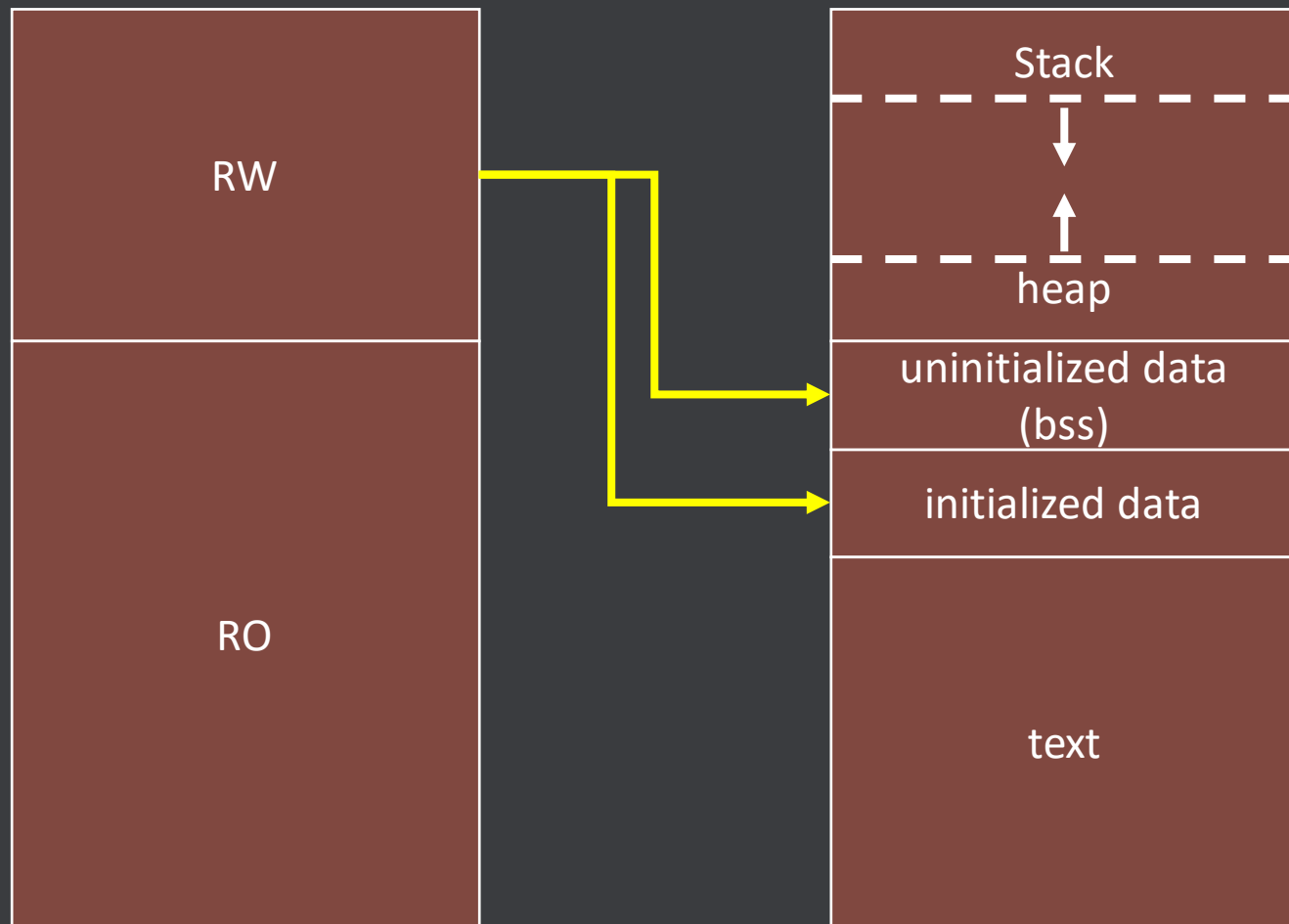
STM32程序组成

程序运行

生成反汇编指令 `fromelf --text -c --output LedTask\LedTask.text LedTask\LedTask.axf`



C运行环境初始化



静态内存

ZI-data栈空间
ZI-data
RW data

静态内存：

静态内存是指在程序开始运行时由编译器分配的内存，它的分配是在程序开始编译时完成的，不占用CPU资源。程序中的各种变量，在编译时系统已经为其分配了所需的内存空间，当该变量在作用域内使用完毕时，系统会自动释放所占用的内存空间。变量的分配与释放，都无须程序员自行考虑。

动态内存

ZI-data堆空间

动态内存：

动态内存分配是按输入信息的大小分配所需要的内存单元，他的特点是按需分配，内存分配在堆区。

用户无法确定空间大小，或者空间太大，栈上无法分配时，会采用动态内存分配。

区别

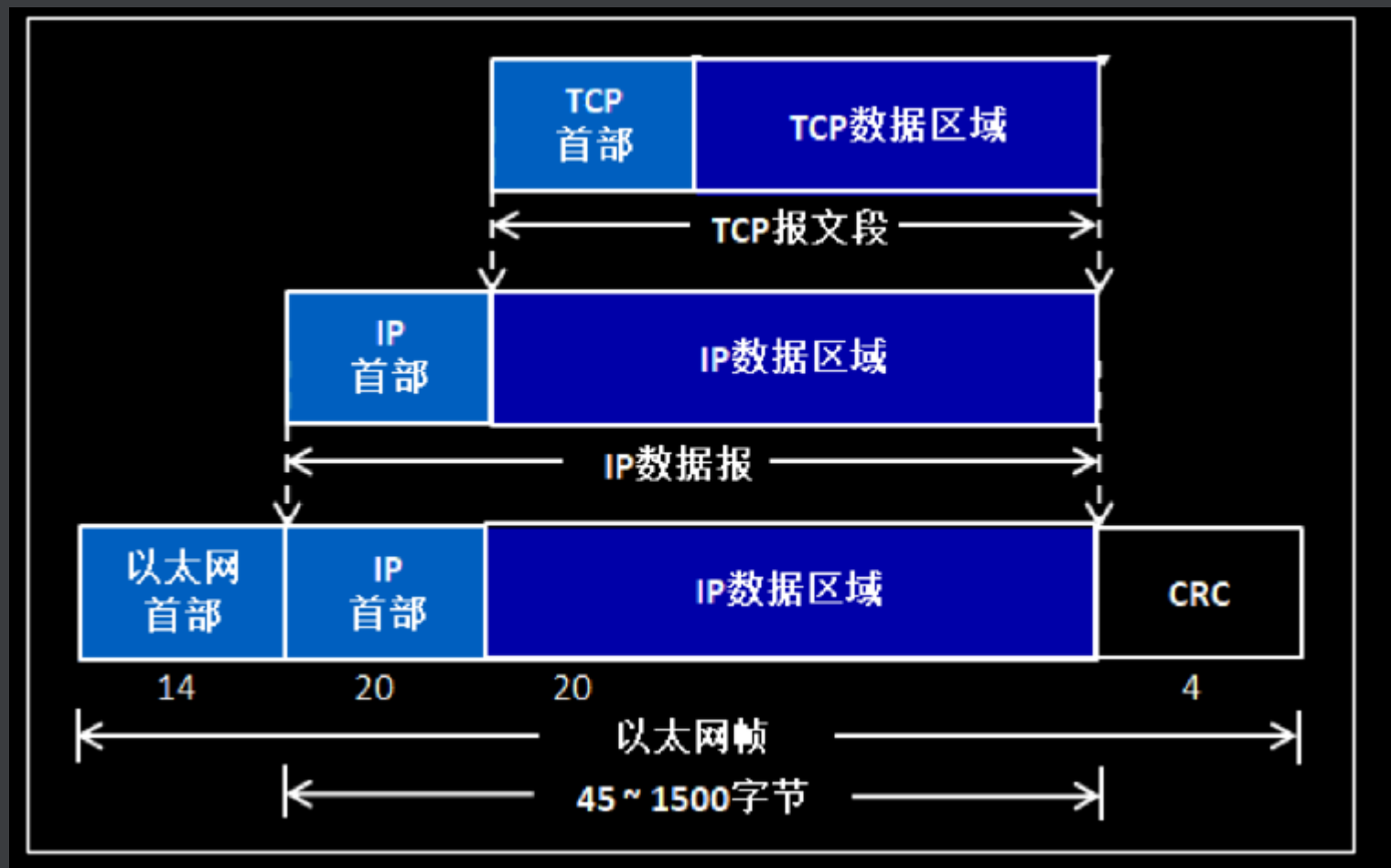
时间不同：

静态分配发生在程序编译和连接的时候。动态分配则发生在程序调入和执行的时候。

空间不同：

堆都是动态分配的，没有静态分配的堆。栈有2种分配方式：静态分配和动态分配。静态分配是编译器完成的，比如局部变量的分配。动态分配由函数malloc进行分配。不过栈的动态分配和堆不同，他的动态分配是由编译器进行释放，无需我们手工实现。

动态内存分配需求



动态内存分配需求



C标准库-动态内存分配接口

```
#include <stdlib.h>

void *malloc(size_t size);

void free(void *ptr);

void *calloc(size_t nmemb, size_t size);

void *realloc(void *ptr, size_t size);
```

C标准库-动态内存分配缺陷

- 这些函数在小型嵌入式系统中并不总是可用的，小型嵌入式设备中的 RAM 不足。
- 它们的实现可能非常的大，占据了相当大的一块代码空间。
- 他们几乎都不是安全的。
- 它们并不是确定的，每次调用这些函数执行的时间可能都不一样。
- 它们有可能产生碎片。
- 这两个函数会使得链接器配置得复杂。
- 如果允许堆空间的生长方向覆盖其他变量占据的内存，它们会成为 debug 的灾难。

动态内存分配缺陷

- 没有虚拟内存机制，需要用户亲自管理。
- 对内存的分配时间要求更为苛刻，分配内存的时间必须是确定的
- RAM比较小，需要比较好的内存分配算法解决内存碎片问题





立鑫老师

小米、百度研发总监

华为、爱立信研发总监

俊龙老师

小美老师


ARM全球认证讲师

BAT导师团 基础预科班



项目	内容
周期	30 天
导师团	百度、华为互联网大牛执教
课程内容	C 语言从入门到精通、Linux 及 C 高级、shell 脚本、数据结构及算法
	两次就业指导
	经典面试题讲解
	BAT 导师 1v1 模拟面试
	每日作业批改+两次测评（期中、期末）
授课形式	录播+每周至少 2 次的直播
教学服务	技术老师 7*13 小时答疑辅导+班主任督学



李宇卓(华清 创客学院) 



扫一扫上面的二维码图案, 加我微信

1 / 嵌入式内存管理需求在哪？

2 / 常见的内存管理方案

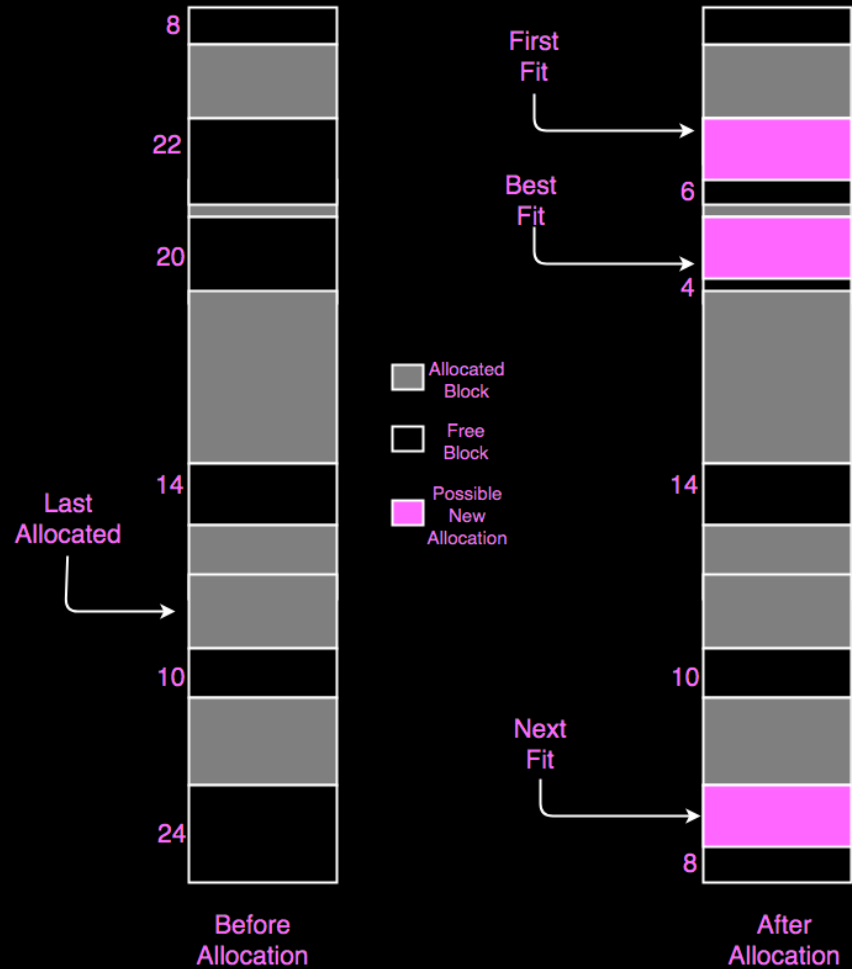
3 / freeRTOS内存分配原理

内存池管理方案



内存堆管理方案

Memory Allocation before and after allocation of 16 M of memory

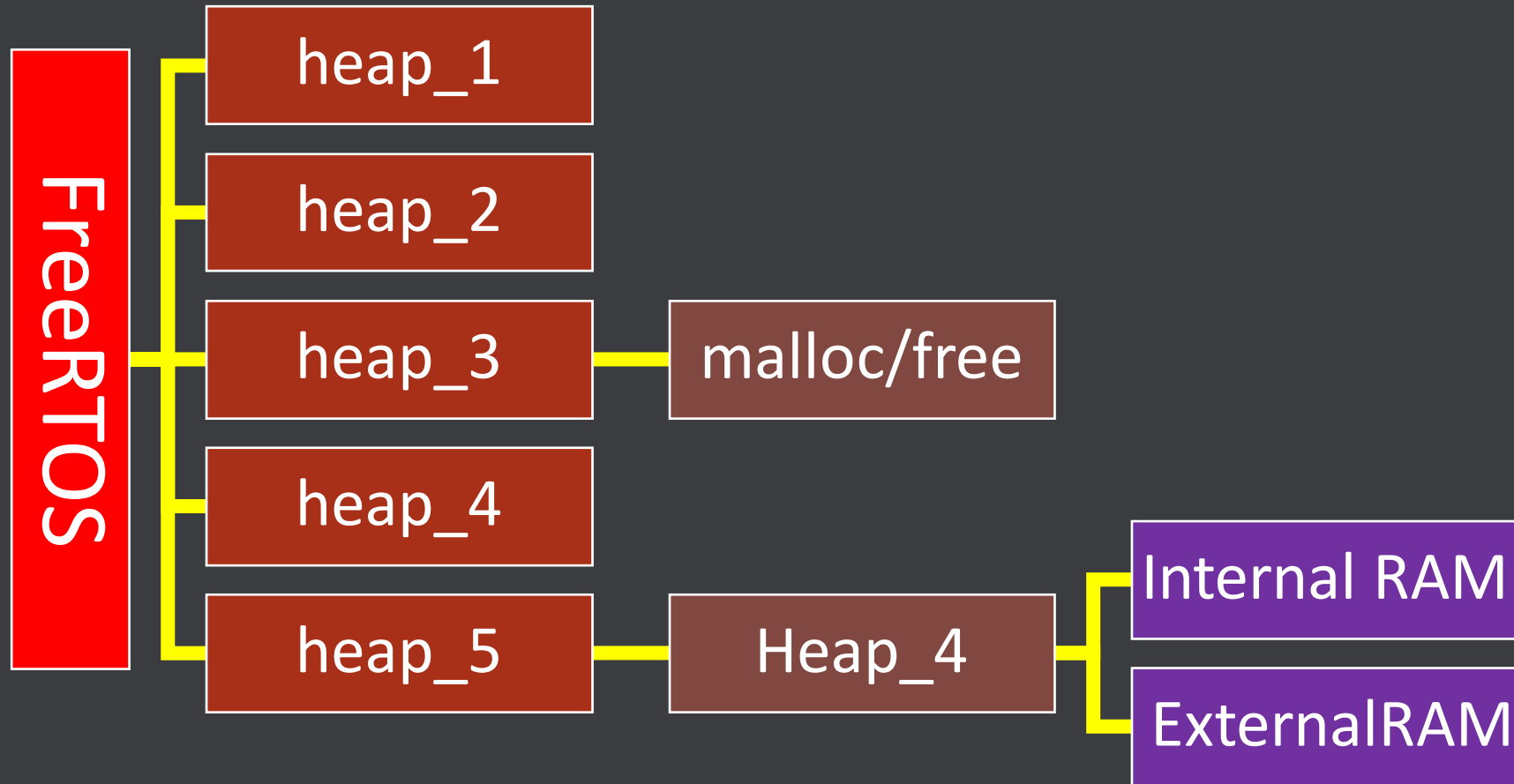


1 / 嵌入式内存管理需求在哪？

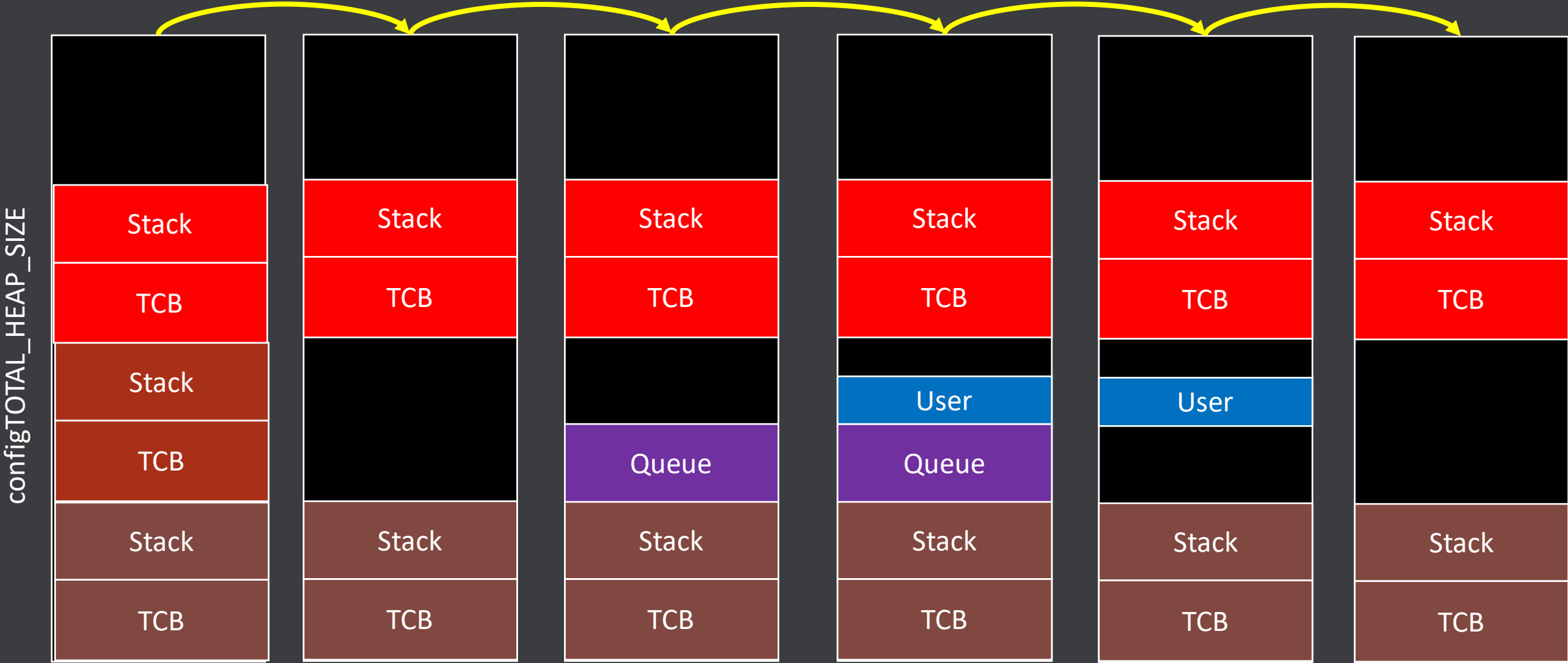
2 / 常见的内存管理方案

3 / freeRTOS内存分配原理

FreeRTOS动态内存分配



heap_4



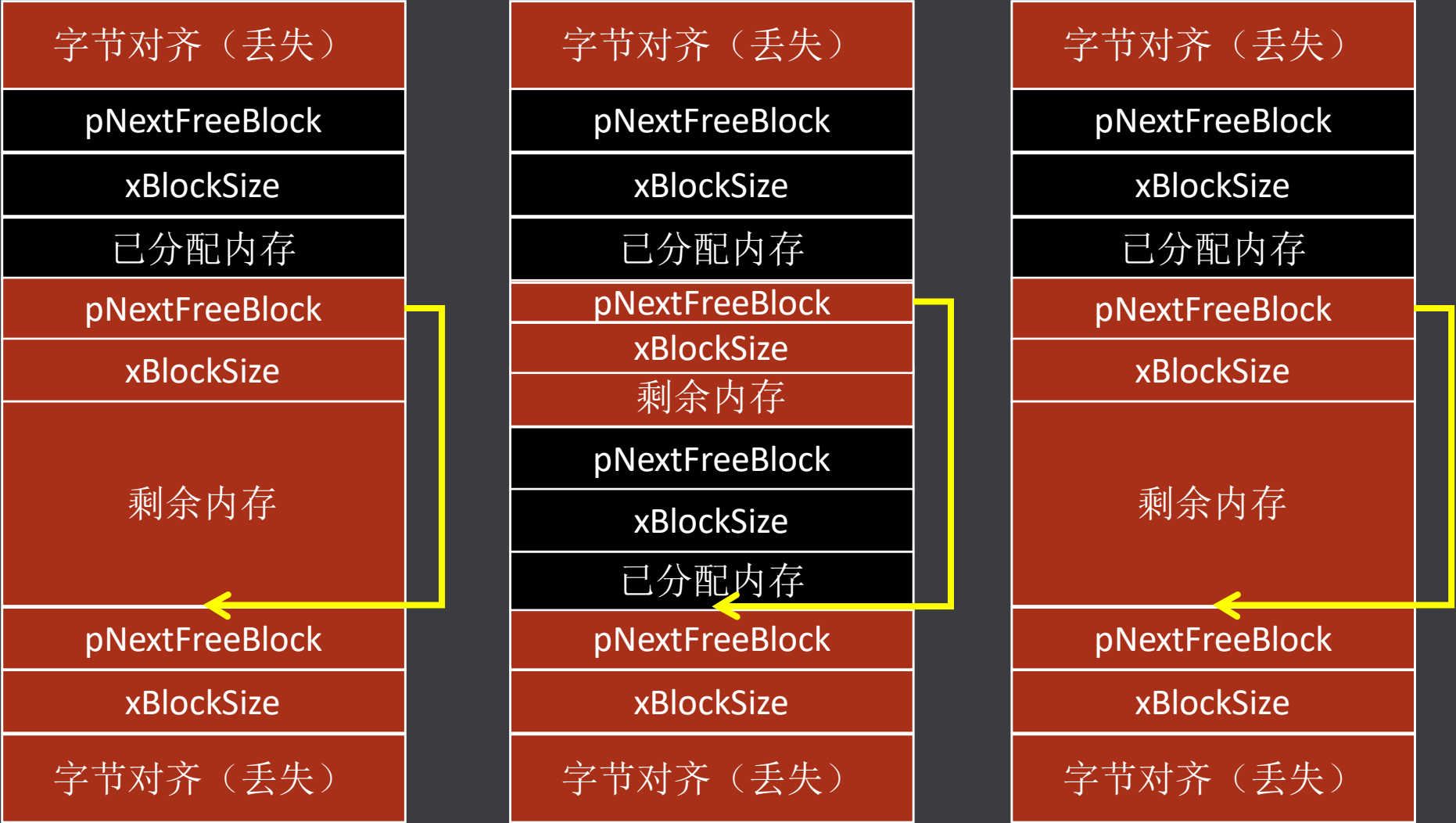
内存管理性能要求

- 按照调用者的要求分配合适的大小的动态内存区，返回该内存块的首地址
- 如果没有足够的内存，则返回空指针
- 用户不在使用该内存是可以调用Free函数释放该内存块，被释放的内存块在归还系统堆之后，可以被重新分配
- 由于动态内存分配算法的重要性，要求快速分配算法并尽量减少内存碎片

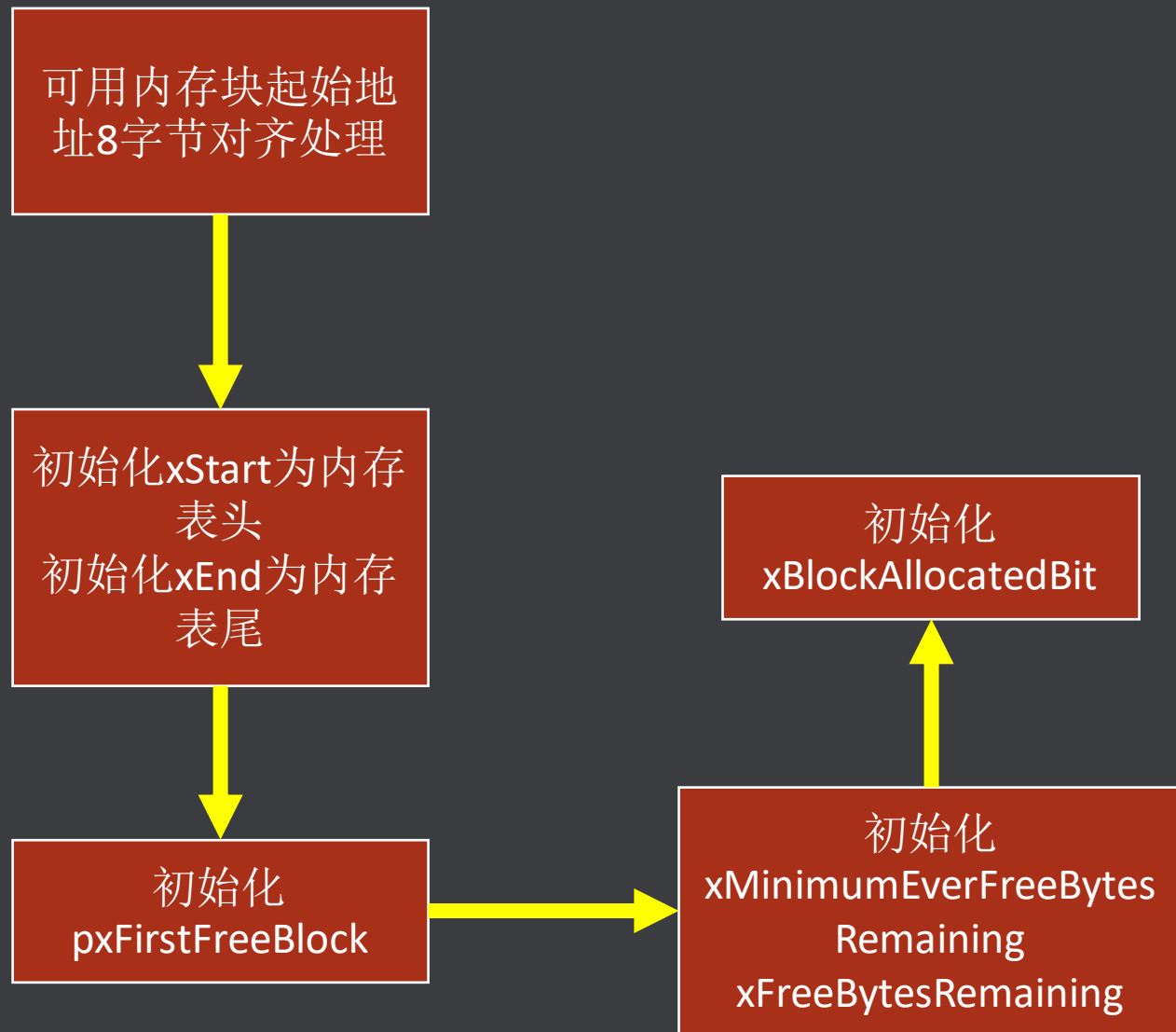
Heap_4内存管理数据结构

ucHeap[configTOTAL_HEAP_SIZE]	静态数组用于内存分配空间
typedef struct A_BLOCK_LINK { struct A_BLOCK_LINK *pNextFreeBlock; size_t xBlockSize; } BlockLink_t;	内存块链表
xHeapStructSize	内存块头长度
BlockLink_t xStart, *pxEnd	链表头结点, 链表尾节点
size_t xFreeBytesRemaining	空闲内存尺寸
size_t xMinimumEverFreeBytesRemaining	每次分配后最小空闲内存块尺寸
xBlockAllocatedBit	内存已使用标记

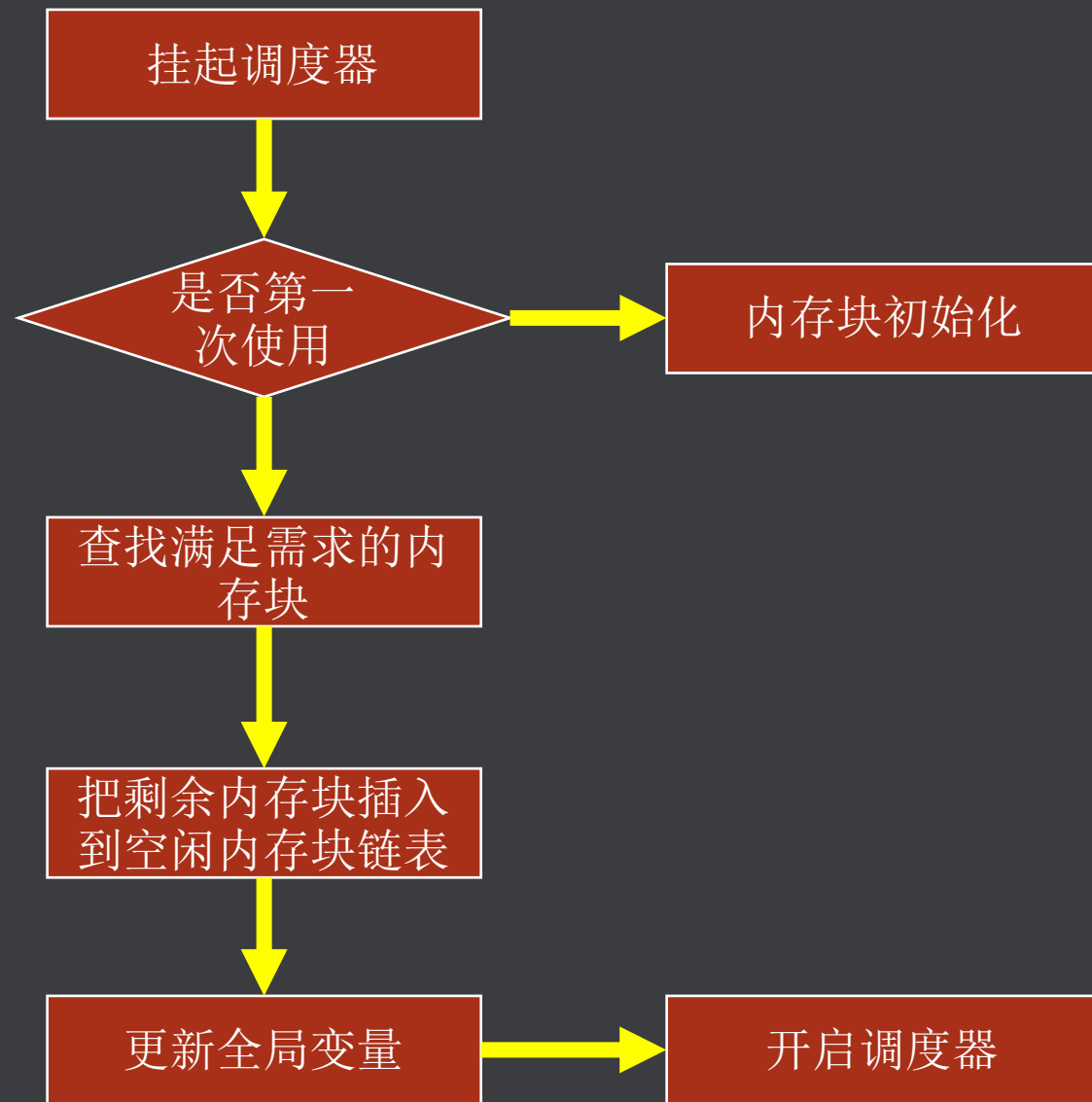
Heap_4内存管理方案



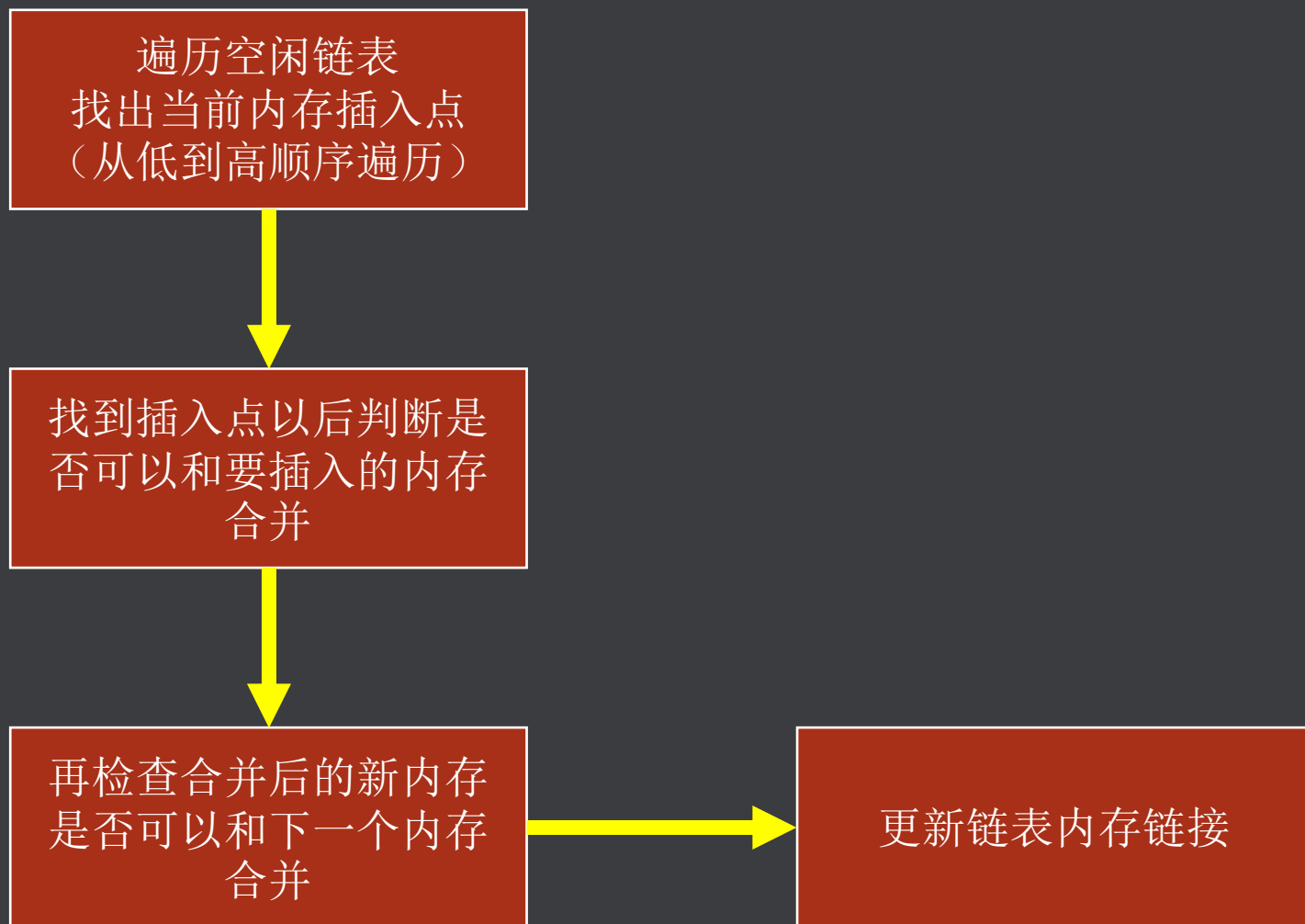
Heap_4内存块初始化



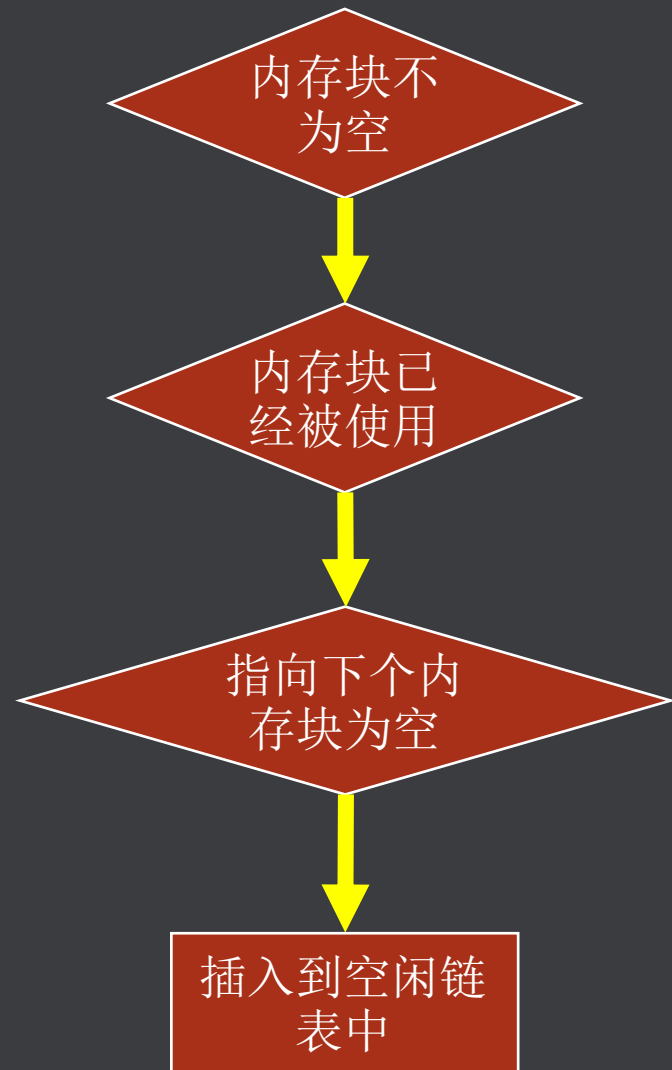
Heap_4内存块申请



Heap_4内存块插入



Heap_4内存块释放





立鑫老师

小米、百度研发总监

华为、爱立信研发总监

俊龙老师

小美老师


ARM全球认证讲师

BAT导师团 基础预科班



项目	内容
周期	30 天
导师团	百度、华为互联网大牛执教
课程内容	C 语言从入门到精通、Linux 及 C 高级、shell 脚本、数据结构及算法
	两次就业指导
	经典面试题讲解
	BAT 导师 1V1 模拟面试
	每日作业批改+两次测评（期中、期末）
授课形式	录播+每周至少 2 次的直播
教学服务	技术老师 7*13 小时答疑辅导+班主任督学



李宇卓(华清 创客学院) 



扫一扫上面的二维码图案, 加我微信

扫一扫，获取更多信息



THANK YOU