内存的大管家 内存管理,是指软件运行时对MCU内存资源的分配和使用的技术 内存管理目的 高效、快速的分配,适当时候释放和回收内存资源 1、内存管理简介(熟悉) malloc函数用于内存申请 内存申请(分配) 内存使用三部曲 内存使用 内存释放 free函数用于内存释放 分块式内存管理由 内存池 和 内存管理表 两部分组成。 内存管理 内存块 内存块 内存块 内存块 内存池 内存池被等分为n块,对应的内存管理表,大小也是n 第1项 第2项 第3项 内存管理表 内存管理表的每一个项对应内存池的一块内存 分配方向 0时,代表对应的内存池未被占用 malloc, free等函数 内存管理表的项值 非0时,代表该项对应的内存块已占用并代表连续占用的内存块数 内存管理初始化时,内存管理表全部清零 分配方向 从最末端开始找空内存 ① 先判断p要分配的内存块数(m) ② 从第n项开始,向下查找,直到找到m块连续的空内存块(即对应内存管理表项为0) 当指针p调用malloc申请内存时 ③ 将这m个内存管理表项的值都设置为m(标记被占用) 2、分块式内存管理介绍(掌握) ④ 把最后的这个空内存块的地址返回指针p, 完成一次分配 内存管理实验 注意:内存不够(没有那么多连续m块空闲内存),返回NULL,表示分配失败 分配原理 内存管理表 第1项 第2项 底←顶 ① 先判断p指向的内存地址所对应的内存块 调用free函数释放内存时 ② 找到对应的内存管理表项目,得到p所占用的内存块数目m ③ 将这m个内存管理表项目的值都清零,标记释放,完成一次内存释放 释放原理 指针p 内存管理表 第1项 第2项 第3项(0 内存池大小 内存块大小 分块内存管理需要的参数 内存块数/管理表项数目 管理表项大小 1、初始化内存 void my_mem_init(uint8_t memx) 2、申请内存 void *mymalloc(uint8 t memx, uint32 t size) 使用步骤 3、操作内存 sprintf((char *)p, "Memory Malloc Test %3d", i) 4、释放内存 void myfree(uint8_t memx, void *ptr) 3、内存管理使用(掌握) 内存管理控制器 struct _m_malloc_dev