第一节课内容总结

alloc方法的底层调用流程

alloc-> objc_alloc —> callAlloc —> objc_msgSend —> alloc —> _objc_rootAlloc —> callAlloc —> _objc_rootAllocWithZone —> _class_createInstanceFromZone

cls->instanceSize()

计算对象需要的内存的大小。

calloc()`

系统实际为对象分配内存的大小。

为什么要字节对齐?

字节是内存的容量单位。但是，CPU在读取内存的时候，却不是以字节为单位来读取的，而是以“块”为单位读取的，所以大家也经常听到一块内存，“块”的大小也就是内存存取的力度。如果不对齐的话，在我们频繁的存取内存的时候，CPU就需要花费大量的精力去分辨你要读取多少字节，这就会造成CPU的效率低下，如果想要CPU能够高效读取数据，那就需要找一个规范，这个规范就是字节对齐。

为什么对象内部的成员变量是以8字节对齐，系统实际分配的内存以16字节对齐?

以空间换时间。苹果采取16字节对齐，是因为OC的对象中，第一位叫isa指针，它是必然存在的，而且它就占了8位字节，就算对象中没有其他的属性了，也一定有一个isa，那对象就至少要占用8位字节。如果以8位字节对齐的话，如果连续的两块内存都是没有属性的对象，那么它们的内存空间就会完全的挨在一起，是容易混乱的。以16字节为一块，这就保证了CPU在读取的时候，按照块读取就可以，效率更高，同时还不容易混乱。

对象的本质

objc_object结构体，里面存储isa指针和成员变量的值。

结构体的内存对齐方式

１:数据成员对齐规则：结构(struct)的第一个数据成员放在offset为0的地方，以后每个数据成员存储的起始位置要从该成员大小或者成员的子成员大小的整数倍开始(比如int为４字节,则要从４的整数倍地址开始存储)。

２:结构体作为成员:如果一个结构里有某些结构体成员,则结构体成员要从其内部最大元素大小的整数倍地址开始存储.(struct a里存有struct b,b里有char,int ,double等元素,那b应该从8的整数倍开始存储)。

３:收尾工作:结构体的总大小,也就是sizeof的结果必须是其内部最大成员的整数倍,不足的要补齐。