

KC_VIP 大师班底层原理第 1 次考试

答案解析

考试题型: 总分 200 分

- 1、选择题 (每题 5 分, 共 10 道 50 分)
- 2、判断题 (每题 5 分, 共 6 道 30 分)
- 3、简单题 (每题 10 分 共 10 道 100 分)
- 4、拓展满分题 (20 分)

提醒事项: 大家尽量自主完成, 不要百度, 不要运行, 尽量在 60-90 分钟完成
基本信息:

(1)姓名:

(2)QQ :

一、选择题(每题 5 分)

1. 在 LP64 下,一个指针的有多少个字节 分值 5 分

- ☐ A: 4
- ☒ B: 8
- ☐ C: 16
- ☐ D: 64

回答正确 (+5 分)

2. 一个实例对象的内存结构存在哪些元素 分值 5 分

- ☒ A: 成员变量
- ☐ B: superClass
- ☐ C: cache_t
- ☐ D: bit

回答正确 (+5 分)

3. 下面图片中 sizeof(struct3)大小等于

```
16 struct LGStruct1 {
17     char b;
18     int c;
19     double a;
20     short d;
21 }struct1;
22
23 struct LGStruct2 {
24     double a;
25     int b;
26     char c;
27     short d;
28 }struct2;
29
30
31 struct LGStruct3 {
32     double a;
33     int b;
34     char c;
35     struct LGStruct1 str1;
36     short d;
37     int e;
38     struct LGStruct2 str2;
39 }struct3;
```

分值 5 分

☐ A: 48

☐ B: 56

☒ C: 64

○D: 72

回答正确 (+5 分)

4. 下列代码: re1 re2 re3 re4 re5 re6 re7 re8 输出结果

```
BOOL re1 = [(id)[NSObject class] isKindOfClass:[NSObject class]]; //
BOOL re2 = [(id)[NSObject class] isKindOfClass:[NSObject class]]; //
BOOL re3 = [(id)[LGPerson class] isKindOfClass:[LGPerson class]]; //
BOOL re4 = [(id)[LGPerson class] isKindOfClass:[LGPerson class]]; //
NSLog(@" re1 :%hhd\n re2 :%hhd\n re3 :%hhd\n re4 :%hhd\n",re1,re2,re3,re4);

BOOL re5 = [(id)[NSObject alloc] isKindOfClass:[NSObject class]]; //
BOOL re6 = [(id)[NSObject alloc] isKindOfClass:[NSObject class]]; //
BOOL re7 = [(id)[LGPerson alloc] isKindOfClass:[LGPerson class]]; //
BOOL re8 = [(id)[LGPerson alloc] isKindOfClass:[LGPerson class]]; //
NSLog(@" re5 :%hhd\n re6 :%hhd\n re7 :%hhd\n re8 :%hhd\n",re5,re6,re7,re8);
```

分值 5 分

○A: 1011 1111

○B: 1100 1011

●C: 1000 1111

○D: 1101 1111

回答正确 (+5 分)

5. $(x + 7) \& \sim 7$ 这个算法是几字节对齐 分值 5 分

○A: 7

●B: 8

○C: 14

○D: 16

回答正确 (+5 分)

6. 判断下列数据结构大小

```
union kc_t {  
    uintptr_t bits;  
    struct {  
        int a;  
        char b;  
    };  
}
```

分值 5 分

- ☒ A: 8
- ☐ B: 12
- ☐ C: 13
- ☐ D: 16

回答正确 (+5 分)

7. 元类的 isa 指向谁, 根元类的父类是谁 分值 5 分

- ☐ A: 自己, 根元类
- ☒ B: 自己, NSObject
- ☐ C: 根元类, 根元类
- ☒ D: 根元类, NSObject

回答正确 (+5 分)

8. 查找方法缓存的时候发现是乱序的, 为什么? 哈希冲突怎么解决的 分值 5 分

- ☐ A: 哈希函数原因, 不解决
- ☐ B: 哈希函数原因, 再哈希
- ☐ C: 他存他的我也布吉岛, 再哈希
- ☐ D: 他乱由他乱, 清风过山岗, 不解决

回答正确 (+5 分)

9. 消息的流程是 分值 5 分

- ☒ A: 先从缓存快速查找
- ☒ B: 慢速递归查找 methodlist (自己的和父类的, 直到父类为 nil)

●C: 动态方法决议

●D: 消息转发流程

回答正确 (+5 分)

10. 类方法动态方法决议为什么在后面还要实现 `resolveInstanceMethod` 分值 5 分

●A: 类方法存在元类(以对象方法形式存在), 元类的父类最终是 `NSObject` 所以我们可以通过 `resolveInstanceMethod` 防止 `NSObject` 中实现了对象方法!

○B: 因为在 oc 的底层最终还是对象方法存在

○C: 类方法存在元类以对象方法形式存在.

○D: 先吃萝卜,瞎操心! 苹果瞎写的 不用管

回答正确 (+5 分)

二、判断题(每题 5 分)

11. 光凭我们的对象地址,无法确认对象是否存在关联对象 分值 5 分

○对

●错

回答正确 (+5 分)

12. `int c[4] = {1,2,3,4}; int *d = c; c[2] = *(d+2)` 分值 5 分

●对

○错

回答正确 (+5 分)

13. `@interface LGPerson : NSObject{ UIButton *btn }` 其中 `btn` 是实例变量 分值 5 分

●对

○错

回答正确 (+5 分)

14. `NSObject` 除外 元类的父类 = 父类的元类 分值 5 分

●对

○错

回答正确 (+5 分)

15. 对象的地址就是内存元素的首地址 分值 5 分

☒ 对

☐ 错

回答正确 (+5 分)

16. 类也是对象 分值 5 分

☒ 对

☐ 错

回答正确 (+5 分)

三、简单题 (每题 10 分 合计 100 分)

简答题无法复制 希望大家耐心 切忌浮躁 (和谐学习 不急不躁)

17. 怎么将上层 OC 代码还原成 C++代码 (2 种方式) 分值 10 分

(1)clang -rewrite-objc xxxx.m -o xxxx.cpp

(2)xcrun -sdk iphoneos clang -arch arm64 -rewrite-objc xxxx.m -o xxxx.cpp

分值 10 分

18. 怎么打开汇编查看流程,有什么好处? 分值 10 分

- 1、通过 Xcode-debug-debug workflow-always show disassembly 可以查看汇编;
- 2、查看汇编可以从更深层了解当前函数的汇编层面的执行,为 objc 源码分析提供信息避免方向性错误,结合 memory read 可以更清楚的看到寄存器之间是如何互相配合处理配合的;使用汇编查看流程,可以在不确定源码出处和执行流程的情况下,跟踪内部代码,并可以找到出处!同时,结合下符号断点的方式,能够更清晰的跟踪源码实现。

分值 10 分

19. x/4gx 和 p/x 以及 p*\$0 代表什么意思 分值 10 分

x/4gx : 输出一段内存地址,以 8 字节的形式输出 4 段;

p/x :输出一个数据结构的首地址;

p*\$0 : \$0 为指向某一个数据空间的指针,而该指令输出的是该数据的数据结构。

分值 10 分

20. 类方法存在哪里? 为什么要这么设计? 分值 10 分

对象方法存储在类中哦,类方法存储在元类中;

对象方法是由类实例化出来的,类是由元类实例化出来的;

这样设计的好处有以下 3 个方面:

1、底层不用对类方法和对象方法作区分,本质上都是对象方法,方法调用都可以理解为消息发送,只不过消息的接受者即方法的查找对象不一致;

2、这样设计更加的基于对象,类的一切信息都存储在元类中,对象的一切信息都存储在类中,类是元类的实例化对象,对象是类的实例化对象;存储也更符合面向对象的特点;

3、OC 语言设计早期借鉴了另一种 smalltalk 的语言,smalltalk 中也是这么设计的;

分值 10 分

21. 方法慢速查找过程中的二分查找流程,请用伪代码实现 分值 10 分

二分查找的前提是数据必须有序排列;2、不断的找起始位置(base)和有效数据量(count);3、当前目标位置=起始位置(base)+有效数据量(count)/2;4、查找对应位置之后不断的向前偏移,主要作用为了找到第一个符合条件的数据,即找到最前面的 category 中满足条件的方法;

```
first = 0
```

```
probe;
```

```
base = first;
```

```
for (probe = 0;count !=0;count=count/2){
```

```
    probe = base+count/2;
```

```
    if(key == probe){
```

```
        while(probe > first && key==(probe - 1)){
```

```
            probe--;
```

```
        }
```

```
        return probe;
```

```
    }
```

```
    if(key > value){
```

```
        base = probe+1;count--;
```

```
    }
```

```
}
```

分值 10 分

22. ISA_MASK = 0x00007fffffff8ULL 那么这个 ISA_MASK 的算法意义是什么? 分值 10 分

这个算法主要是为了得到 isa 中存储的 class 信息,大部分 isa 都是不纯的 isa,是一个长达 64 个 byte 位的联合体位域数据,而存储 class 信息的部分只有其中的部分 byte 位置,剩下的位置存储了其他的信息;读取的时候必须把其他无效位置的 byte 位数

据给遮盖住,所以需要使用 isa_mask,任何数据与 isa_mask 进行按位与操作,都只会保留 isamask 对应 byte 位置的数据,这个算法的意义就是遮盖,盖住不需要的地方,这样就能让 isa 存储更多的信息;分值 10 分

23. 类的结构里面为什么会有 rw 和 ro 以及 rwe ? 分值 10 分

- (1)ro 属于 clean memory,在编译即确定的内存空间,只读,加载后不会改变内容的空间;
- (2)rw 属于 dirty memory,rw 是运行时结构,可读可写,可以向类中添加属性、方法等,在运行时改变的内存;
- (3)rwe 相当于类的额外信息,因为在实际使用过程中,只有很少的类会真正的改变他们的内容,所以为避免资源的消耗就有了 rwe;
- (4)运行时,如果需要动态向类中添加方法协议等,会创建 rwe,并将 ro 的数据优先 attache 到 rwe 中,在读取时会优先返回 rwe 的数据,如果 rwe 没有被初始化,则返回 ro 的数据。

rw 中包括 ro 和 rwe,其中 rw 是 dirtymemory,ro 是 clean memory;为了让 dirty memory 占用更少的空间,把 rw 中可变的部分抽取出来为 rwe;

clean memory 越多越好,dirty memory 越少越好,因为 iOS 系统底层虚内存机制的原因,内存不足时会把一部分内存回收掉,后面需要再次使用时从硬盘中加载出来即 swap 机制,clean memory 是可以从硬盘中重新加载的内存,iOS 中的 macho 文件动态库都属于此类行;dirty memory 是运行时产生的数据,这部分数据不能从硬盘中重新加载所以必须一直占据内存,当系统物理内存紧张的时候,会回收掉 clean memory 内存,如果 dirty memory 过大则直接会被回收掉;所以 clean memory 越多越好,dirty memory 越少越好;苹果对 rw、ro、rwe 进行这么细致的划分都是为了能更好更细致的区别 cleanmemory 和 dirty memory;

分值 10 分

24. cache 在什么时候开始扩容 , 为什么? 分值 10 分

- 1、一般情况下:如果当前方法 cache 之后,缓存的使用容量会超过总容量的 3/4,那么此时就不会先插入,而是先触发扩容,扩容为原来的 2 倍,然后再插入本次的方法;
- 2、某些特殊预处理宏定义编译命令下,首次会存储满之后再开始扩容;
- 3、扩容时选用 3/4 作为负载因子是和 hash 表底层使用的链表以及红黑树的数据结构有关,0.75 是最符合泊松分布概率计算得出的数值,在这个数值下哈希表的空间和时间效率都是最高的;

分值 10 分

25. objc_msgSend 为什么用汇编编写 , objc_msgSend 是如何递归找到 imp? 分值 10 分

- (1)使用汇编响应速度快;
- (2)这个过程中使用了两个循环:

1.循环 1:通过前面获取的 mask,与要查找的_cmd 进行 hash 运算,获取下标,从而得到 _cmd 对应的 bucket 地址,然后进行向前平移查找,每次平移 16 个字节,如果找到对应的 sel,则 cacheHit;当平移到 buckets 的首地址,依然没有查找到,则进入第二个循环;
2.循环 2:首先会获取末尾 bucket 的地址,同样采用向前查找的方式,向_cmd 对应的地址进行平移查找。

分值 10 分

26. 一个类的类方法没有实现为什么可以调用 NSObject 同名对象方法 分值 10 分

这里涉及到 isa 和 superclass 的走位以及方法查找逻辑。

首先类的 isa 指向元类,在方法快速查找时,会根据类的 isa 找到元类,元类中没有该方法,就会走到 lookUpImpOrForward 慢速查找流程中。在慢速查找流程中,会进行 for 递归查找,根据 superclass 查找到元类的父类,也就是根元类,而根元类的 superclass 指向了 NSObject,所以会调用到 NSObject 的同名对象方法。

分值 10 分

四、拓展满分题 (20 分)

27. 提交一篇大师班学习期间,你写的最好的一篇博客 分值 20 分