1.将Objective-C代码转换为C\C++代码

```
clang -rewrite-objc file_name.m -o file_name.cpp
```

2.指定设备/架构/SDK版本将Objective-C代码转换为C\C++代码

```
xcrun -sdk iphoneos/iphonesimulator10.3 clang -arch arm64 -rewrite-objc file_name.m -o file_name.cpp
```

3.支持ARC、指定运行时系统版本

```
xcrun -sdk iphoneos/iphonesimulator10.3 clang -arch arm64 -rewrite-objc -fobjc-arc -fobjc-runtime=ios-8.0.0 file_name.m -o file_name.cpp
```

4.NSObject的内存本质

```
oc代码
@interface NSObject {
    Class isa;
}
@end

c++代码
struct NSObject_IMPL {
    Class isa;
};
```

5.自定义类的内存

```
struct Student_IMPL {
    Class isa;
    int _no;
    int _age;
};

Student *stu = [[Student alloc] init];
    stu->_no = 4;
    stu->_age = 5;

    struct Student_IMPL *stuImpl = (_bridge struct Student_IMPL *)(stu);
    NSLog(@"[sutImpl]:_no=%d,_age=%d",stuImpl->_no,stuImpl->_age);

// output:[sutImpl]:_no=4,_age=5
```

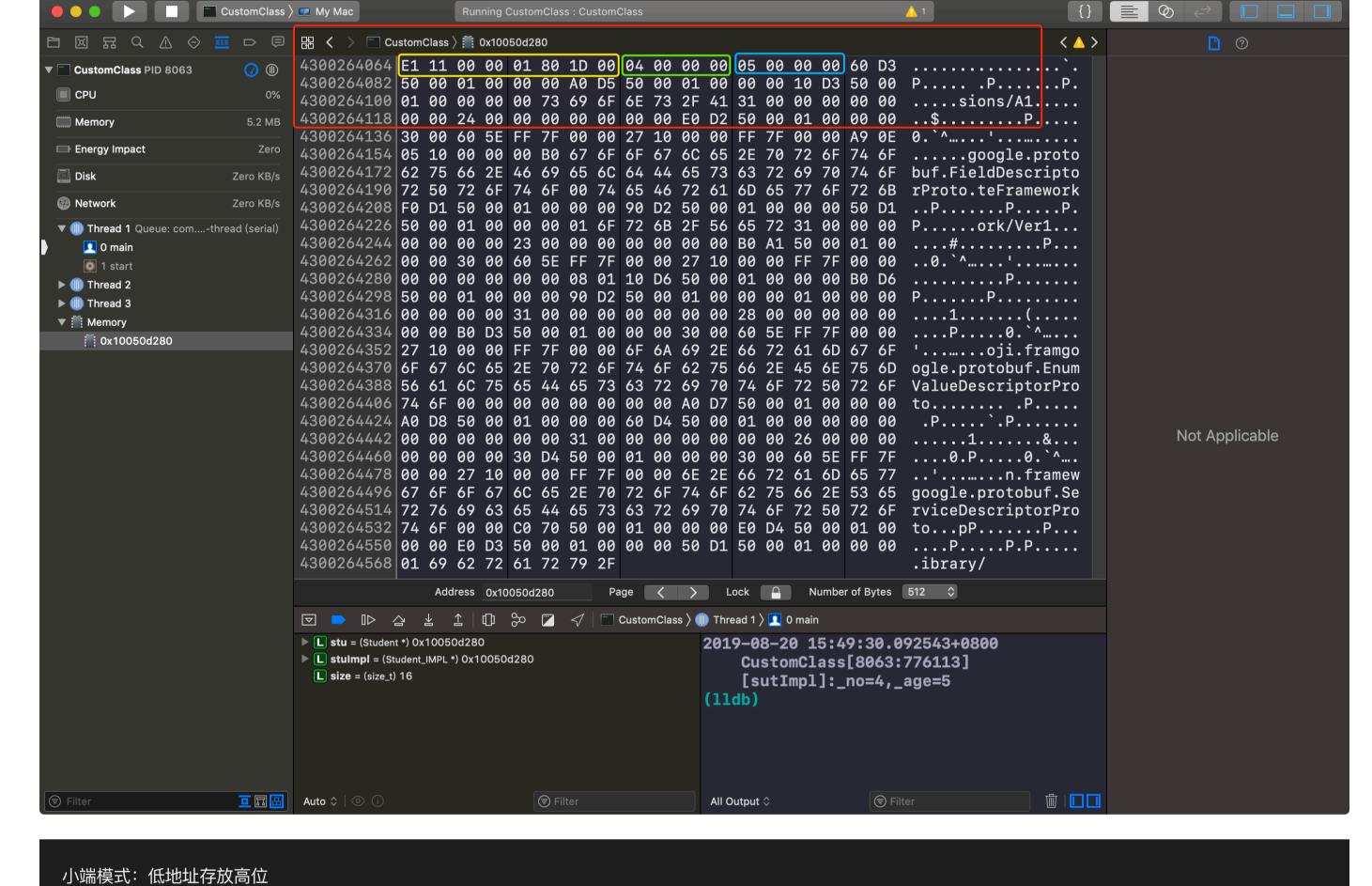
6.获取对象占用空间的api

```
size_t size = class_getInstanceSize([Student class]);
NSLog(@"%zu",size);//16
```

Parameter CustomClass → main.m → Family main()

7.窥探内存结构

```
Identity and Type
▼ CustomClass
                                       int _age;
 ▼ CustomClass
                                      };
                                                                                                                                            Type Default - Objective-C Sou... 🗘
 Products
                                   41 struct Student_IMPL {
                                           Class isa;
                                                                                                                                          Full Path /Users/Jentle/Desktop/
                                                                                                                                                xmg_object/001.OC对象的本
                                           int _age;
                                                                                                                                                质/2CustomClass/
                                                                                                                                                CustomClass/main.m
                                   45 };
                                                                                                                                       On Demand Resource Tags
                                      int main(int argc, const char * argv[]) {
                                            @autoreleasepool {
                                                Student *stu = [[Student alloc] init];
                                                                                                                                       Target Membership
                                                                                                                                       CustomClass
                                                stu->_no = 4;
                                   51
                                                stu->_age = 5;
                                   52
53
                                                                                                                                       Text Settings
                                                struct Student_IMPL *stuImpl = (__bridge struct Student_IMPL *)(stu);
                                                                                                                                       Text Encoding No Explicit Encoding
                                                NSLog(@"[sutImpl]:_no=%d,_age=%d",stuImpl->_no,stuImpl->_age);
                                                                                                                                        Line Endings No Explicit Line E
                                                size_t size = class_getInstanceSize([Student class]);
                                                                                                                                        Indent Using Spaces
                                                NSLog(@"%zu", size);//16
                                                                                                                                                                 4 0
                                                                                                                                                      4 0
                                                                                                                                                            Indent
                                                                                                                                                Wrap lines
                                           return 0;
                                   60 }
                                ▽
                                                                                   2019-08-20 15:26:58.791832+0800
                                                                                        CustomClass[7900:708427]
                                                                                        [sutImpl]:_no=4,_age=5
                                                                                   2019-08-20 15:30:46.348399+0800
                                                                                        CustomClass[7900:708427] 16
                                                                                   Program ended with exit code: 0
```



0x 00 00 00 04 == 4 0x 00 00 00 05 == 5 8.判断机器的字节序

BOOL IsBigEndian(){

```
BOOL IsBigEndian(){

int a = 0x1234;
char b = *(char *)&a; //通过将int强制类型转换成char单字节,通过判断起始存储位置。即等于 取b等于a的低地址部分
if( b == 0x12)
{
    return TRUE;
}
return FALSE;
}
```

print/p 打印po 打印对象

- memory read/数量格式字节数 内存地址 格式: x是16进制, f是浮点, d是10进制
- x/数量格式字节数 内存地址
 字节大小: b:byte 1字节, h:half word 2字节
 w:word 4字节, g:giant word 8字节
- 列子: x/4xw 0x102006870
- 10.面试题

一个NSObject对象占用多少内存?一个指针变量所占用的大小(64bit,8个字节。32bit,4个字节)