

QEMU中的对象模型——QOM（介绍篇）

原创 YuanruiZJU 最后发布于2016-12-06 13:21:20 阅读数 4185 ☆ 收藏

QEMU提供了一套面向对象编程的模型——QOM，即QEMU Object Module，几乎所有的设备如CPU、内存、总线等都是利用这一模型的实现代码位于qom/文件夹下的文件中。对于开发者而言，只要知道如何利用QOM模型创建类和对象就可以了，但是开发者只有了解了QOM的模型才能清楚如何利用QOM模型。因此本文先对QOM的必要性展开叙述，然后说明QOM的相关数据结构，在读者了解了QOM的数据结构的基础上，再介绍如何使用QOM模型创建新对象和新类，在下一篇中，介绍QOM是如何实现的。同时对于阅读代码的人来说，理解和掌握QOM是学习QEMU代码的重

为什么QEMU中要实现对象模型

- 各种架构CPU的模拟和实现
QEMU中要实现对各种CPU架构的模拟，而且对于一种架构的CPU，比如X86_64架构的CPU，由于包含的特性不同，也会有不同的CPU模型。CPU通用的属性，同时也包含各自特有的属性。为了便于模拟这些CPU模型，面向对象的变成模型是必不可少的。
- 模拟device与bus的关系
在主板上，一个device会通过bus与其他的device相连接，一个device上可以通过不同的bus端口连接到其他的device，而其他的device也可以与其他的设备连接，同时一个bus上也可以连接多个device，这种device连bus、bus连device的关系，qemu是需要模拟出来的。为了方便模拟，面向对象的编程模型也是必不可少的。

QOM模型的数据结构

这些数据结构中TypeImpl定义在qom/object.c中，ObjectClass、Object、TypeInfo定义在include/qom/object.h中。在include/qom/object.h的注释中，对它们的每个字段都有比较明确的说明，并且说明了QOM模型的法。

1. TypeImpl：对数据类型的抽象数据结构

```
1 struct TypeImpl
2 {
3     const char *name;
4
5     size_t class_size; /*该数据类型所代表的类的大小*/
6
7     size_t instance_size; /*该数据类型产生的对象的大小*/
8
9     /*类的 Constructor & Destructor*/
10    void (*class_init)(ObjectClass *klass, void *data);
11    void (*class_base_init)(ObjectClass *klass, void *data);
12    void (*class_finalize)(ObjectClass *klass, void *data);
13
14    void *class_data;
15
16    /*实例的Constructor & Destructor*/
17    void (*instance_init)(Object *obj);
18    void (*instance_post_init)(Object *obj);
19    void (*instance_finalize)(Object *obj);
20
21    bool abstract; /*表示类是否是抽象类*/
22
23    const char *parent; /*父类的名字*/
24    TypeImpl *parent_type; /*指向父类TypeImpl的指针*/
25
26    ObjectClass *class; /*该类型对应的类的指针*/
27
28    int num_interfaces; /*所实现的接口的数量*/
29    InterfaceImpl interfaces[MAX_INTERFACES];
30 };
31 其中InterfaceImpl的定义如下，只是一个类型的名字
32 struct InterfaceImpl
33 {
34     const char *typename;
35 };
```

2. ObjectClass: 是所有类的基类



4



2



```
2
3 struct ObjectClass
4 {
5     /*< private >*/
6     Type type; /**/
7     GSList *interfaces;
8
9     const char *object_cast_cache[OBJECT_CLASS_CAST_CACHE];
10    const char *class_cast_cache[OBJECT_CLASS_CAST_CACHE];
11
12    ObjectUnparent *unparent;
13 };
```



4



2



3. Object: 是所有对象的base Object

```
1 struct Object
2 {
3     /*< private >*/
4     ObjectClass *class;
5     ObjectFree *free; /*当对象的引用为0时，清理垃圾的回调函数*/
6     GHashTable *properties; /*Hash表记录Object的属性*/
7     uint32_t ref; /*该对象的引用计数*/
8     Object *parent;
9 };
```

4. TypeInfo: 是用户用来定义一个Type的工具型的数据结构，用户定义了一个TypeInfo，然后调用type_register(TypeInfo)或者type_register_s函数，就会生成相应的TypeImpl实例，将这个TypeInfo注册到全局的TypeImpl的hash表中。

```
1 /*TypeInfo的属性与TypeImpl的属性对应，实际上qemu就是通过用户提供的TypeInfo创建的TypeImpl的对象*/
2 struct TypeInfo
3 {
4     const char *name;
5     const char *parent;
6
7     size_t instance_size;
8     void (*instance_init)(Object *obj);
9     void (*instance_post_init)(Object *obj);
10    void (*instance_finalize)(Object *obj);
11
12    bool abstract;
13    size_t class_size;
14
15    void (*class_init)(ObjectClass *klass, void *data);
16    void (*class_base_init)(ObjectClass *klass, void *data);
17    void (*class_finalize)(ObjectClass *klass, void *data);
18    void *class_data;
19
20    InterfaceInfo *interfaces;
21 };
```

怎样使用QOM模型创建新类型

使用QOM模型创建新类型时，需要用到以上的ObjectClass、Object和TypeInfo。关于QOM的用法，在include/qom/object.h一开始就有一长串的注释说明了创建新类型时的各种用法。我们下面是对这些用法的简要说明。

1. 从最简单的开始，创建一个最小的type:

```
1 #include "qdev.h"
2
3 #define TYPE_MY_DEVICE "my-device"
4
5 // 用户需要定义新类型的类和对象的数据结构
6 // 由于不实现父类的虚拟函数，所以直接使用父类的数据结构作为子类的数据结构
7 // No new virtual functions: we can reuse the typedef for the
8 // superclass.
9 typedef DeviceClass MyDeviceClass;
10 typedef struct MyDevice
11 {
```



举报



```
14  int reg0, reg1, reg2;
15  } MyDevice;
16
17  static const TypeInfo my_device_info = {
18      .name = TYPE_MY_DEVICE,
19      .parent = TYPE_DEVICE,
20      .instance_size = sizeof(MyDevice), //必须向系统说明对象的大小，以便系统为对象的实例分配内存
21  };
22
23  //向系统中注册这个新类型
24  static void my_device_register_types(void)
25  {
26      type_register_static(&my_device_info);
27  }
28  type_init(my_device_register_types)
```



4



2



2. 为了方便编程，对于每个新类型，都会定义由ObjectClass动态cast到MyDeviceClass的方法，也会定义由Object动态cast到MyDeviceClass的方法。OBJECT_GET_CLASS、OBJECT_CLASS_CHECK、OBJECT_CHECK都在include/qemu/object.h中定义。

```
1  #define MY_DEVICE_GET_CLASS(obj) \
2      OBJECT_GET_CLASS(MyDeviceClass, obj, TYPE_MY_DEVICE)
3  #define MY_DEVICE_CLASS(klass) \
4      OBJECT_CLASS_CHECK(MyDeviceClass, klass, TYPE_MY_DEVICE)
5  #define MY_DEVICE(obj) \
6      OBJECT_CHECK(MyDevice, obj, TYPE_MY_DEVICE)
```

3. 如果我们在定义新类型中，实现了父类的虚拟方法，那么需要定义新的class的初始化函数，并且在TypeInfo数据结构中，给TypeInfo的class_init函数指针。

```
1  #include "qdev.h"
2
3  void my_device_class_init(ObjectClass *klass, void *class_data)
4  {
5      DeviceClass *dc = DEVICE_CLASS(klass);
6      dc->reset = my_device_reset;
7  }
8
9  static const TypeInfo my_device_info = {
10     .name = TYPE_MY_DEVICE,
11     .parent = TYPE_DEVICE,
12     .instance_size = sizeof(MyDevice),
13     .class_init = my_device_class_init, /*在类初始化时就会调用这个函数，将虚拟函数赋值*/
14 };
```

4. 当我们需要从一个类创建一个派生类时，如果需要覆盖 类原有的虚拟方法，派生类中，可以增加相关的属性将类原有的虚拟函数指针保存，然后新的函数指针，保证父类原有的虚拟函数指针不会丢失。

```
1  typedef struct MyState MyState;
2  typedef void (*MyDoSomething)(MyState *obj);
3
4  typedef struct MyClass {
5      ObjectClass parent_class;
6
7      MyDoSomething do_something;
8  } MyClass;
9
10 static void my_do_something(MyState *obj)
11 {
12     // do something
13 }
14
15 static void my_class_init(ObjectClass *oc, void *data)
16 {
17     MyClass *mc = MY_CLASS(oc);
18
19     mc->do_something = my_do_something;
```



举报



```
22 static const TypeInfo my_type_info = {
23     .name = TYPE_MY,
24     .parent = TYPE_OBJECT,
25     .instance_size = sizeof(MyState),
26     .class_size = sizeof(MyClass),
27     .class_init = my_class_init,
28 };
29
30 typedef struct DerivedClass {
31     MyClass parent_class;
32
33     MyDoSomething parent_do_something;
34 } DerivedClass;
35
36 static void derived_do_something(MyState *obj)
37 {
38     DerivedClass *dc = DERIVED_GET_CLASS(obj);
39
40     // do something here
41     dc->parent_do_something(obj);
42     // do something else here
43 }
44
45 static void derived_class_init(ObjectClass *oc, void *data)
46 {
47     MyClass *mc = MY_CLASS(oc);
48     DerivedClass *dc = DERIVED_CLASS(oc);
49
50     dc->parent_do_something = mc->do_something;
51     mc->do_something = derived_do_something;
52 }
53
54 static const TypeInfo derived_type_info = {
55     .name = TYPE_DERIVED,
56     .parent = TYPE_MY,
57     .class_size = sizeof(DerivedClass),
58     .class_init = derived_class_init,
59 };
```



4



2



点赞 4 收藏 分享 ...



YuanruiZJU

发布了15 篇原创文章 · 获赞 7 · 访问量 3万+



客户关系管理系统crm系统

crm客户关系管理系统



评论一句，前排打call...



YuanruiZJU

3年前

希望大家提出宝贵意见

QEMU中如何定义所有Device的基类和BUS的基类

阅读数 1478

本文介绍QEMU如何模拟设备、总线、主板的连接关系。

博文 来自: [YuanruiZJU的博客](#)

qemu QOM(qemu object model)和设备模拟

阅读数 169

本文所用qemu为1.5版本的，不是android emulator的。之前几篇文章介绍的都是android emulator中的设备模拟...

博文 来自: [ayu_ag的专](#)

qemu中device和driver的区别 使用9 p文件系统

阅读数 21

qemu配置中经常会出现-driver/-device的选项，可以理解成-driver是后端设备，即一个实际的物理的磁盘；device是前端设备，即一个虚拟的磁盘...

博文 来自: [weixin_30](#)



举报



(转载) 本文发源自<http://www.binss.me/blog/qemu-note-of-qemu-object-model/>, 转载请注明出处。QOM(Qem...

博文 来自: [Hugo的博客](#)



客户关系管理系统crm系统

crm客户关系管理系统

qemu device

qemu device abstraction

QEMU设备模拟

备模拟目的我们好像不会干一件事而毫无目的, 就算不停刷微信朋友圈也是为了打发你无聊的时间。其实最装B的回...

博文 来自: [tycoon的专](#)

QEMU, MMIO回调函数如何被调用

我在qemu上注册了一个系统总线设备, .parent = TYPE_SYS_BUS_DEVICE, 使用的下面的函数配置mmio中的0x100的长...

qemu如何实现面向对象模型QOM (代码讲解)

1. QOM中如何将所有的类储存起来的, 并且完整地呈现给使用者的。2. 面向对象编程, 有三个重要的特性——封装...

博文 来自: [YuanruiZJU的博客](#)

QEMU(1) - QOM

阅读数 287

Table of ContentsTypeInfo根类型TypeInfoTypeInfo链路初始化TypeInfo ->ModuleEntryDump TypeInfoTypeImpl...

博文 来自: [lwhuq的博客](#)

QEMU中的对象模型——QOM(介绍篇)_YuanruiZJU的博客-CSDN博客

...面向对象模型QOM(代码讲解)_C/C++_YuanruiZJU的博客-CSDN博客



客户关系管理系统crm系统

crm客户关系管理系统

qemu对象模型——QOM实现分析

阅读数 181

<http://blog.chinaunix.net/uid-28541347-id-5784376.html>

博文 来自: [wxx213的专栏](#)

QEMU学习笔记——QOM(Qemu Object Model) - Hugo的博客 - CSDN博客

QEMU中的内存管理介绍_YuanruiZJU的博客-CSDN博客

qemu代码分析.pdf

01-14

qemu 是使用动态二进制翻译的 cpu 模拟器, 它支持两种运行模式: 全系统模拟和用户态模拟。在全系统模拟下, qemu ...

下载



ayu_ag

32篇文章

关注 排名:千里之外



weixin_30741653

4435篇文章

关注 排名:千里之外



LoneHugo

63篇文章

关注 排名:千里之外

QEMU中协程的实现和使用_YuanruiZJU的博客-CSDN博客

qemu QOM(qemu object model)和设备模拟_ayu_ag的专栏-CSDN博客

qemu2的qom系统分析(-)对象系统

阅读数 85

前边分析machine的注册和选择, 发现如果 不了解qom系统是很难分析的。qom系统的说明在include/qom/object...

博文 来自: [woai110120130的...](#)

virtio的qemu总线与设备模型

阅读数 1万+

(很多内容是网上找的, +上我个人的一点理解, 推荐大家去看 <http://mnstory.net/2014/10/qemu-device-simulati...>

博文 来自: [majieyue的](#)

QEMU中的CPU类型设计_YuanruiZJU的博客-CSDN博客

举报

^

虚拟化中如何实现设备模拟？

在计算机虚拟化领域中，对设备进行模拟是虚拟化实现的基础。设备的模拟主要包括一下三个方面：设备状态的记录…



工位出租600元/月

出租工位

QEMU中的CPU类型设计 - YuanruiZJU的博客 - CSDN博客

qemu-kvm 对mmio的模拟

转：http://blog.chinaunix.net/uid-28541347-id-5789579.htmlMMIO和PIO的区别I/O作为CPU和外设交流的一个渠…

1.1Qemu 用户态架构

本节首先分析Qemu的初始化的顶层流程;从而引出Qemu各大功能模块的描述；最后分析Qemu与内核态KVM的通讯…

qemu问题

kvm的linux已成功用qemu安装了windows系统。 windows也已经用了一段时间，现在想通过qemu引导进入windows系…

QEMU虚拟网卡设备的创建流程

基于qemu-kvm-0.12.1.2-2.160.el6_1.8.src.rpm虚拟网卡类型为virtio-net-pcvirtio网卡设备对应的命令行参数为 -d…

QPM(量化项目管理)

一、1.首先收集原始数据。2.对收集的数据进行处理，会做出一张全生命周期的主模型，包括需求开发、需求评审、…

QEMU 设备模拟

设备模拟目的我们好像不会干一件事而毫无目的，就算不停刷微信朋友圈也是为了打发你无聊的时间。其实最装B的…

android emulator虚拟设备分析第三篇之pipe上的qemud service

一、概述以boot-properties为例，注意不需要看ANDROID-QEMUD.TXT，这个是和guest os中的qemud进行相关的…

Android属性（property）机制

1. 属性简介Android里有很多属性（property），每个属性都有一个名称和值，他们都是字符串格式。这些属性定义…

大学四年自学走来，这些私藏的实用工具/学习网站我贡献出来了

大学四年，看课本是不可能一直看课本的了，对于学习，特别是自学，善于搜索网上的一些资源来辅助，还是非常有…

在中国程序员是青春饭吗？

今年，我也32了，为了不给大家误导，咨询了猎头、圈内好友，以及年过35岁的几位老程序员………舍了老脸去揭人…

KubeOperator开源容器平台



离线和可视化方式部署和管理K8s集群，从这里开启您的Kubernetes之旅。

超全Python图像处理讲解（多图预警）

文章目录Pillow模块讲解一、Image模块1.1、打开图片和显示图片1.2、创建一个简单的图像1.3、图像混合（1）透…

为什么猝死的都是程序员，基本上不见产品经理猝死呢？

相信大家时不时听到程序员猝死的消息，但是基本上听不到产品经理猝死的消息，这是为什么呢？我们先百度搜一下…

毕业5年，我问遍了身边的大佬，总结了他们的学习方法

我问了身边10个大佬，总结了他们的学习方法，原来成功都是有迹可循的。

推荐10个堪称神器的学习网站

每天都会收到很多读者的私信，问我：“二哥，有什么推荐的学习网站吗？最近很浮躁，手头的一些网站都看烦了，…



4

379



2



108



120



1



论坛

阅读数 2213

[博文](#) 来自: [sdulibh的专栏](#)

阅读数 133

[博文](#) 来自: [liuhuanhuan517…](#)

阅读数 4377

[博文](#) 来自: [万能的终端和网络](#)

阅读数 4108

[博文](#) 来自: [ayu_ag的专栏](#)

阅读数 103

[博文](#) 来自: [zhawei jin的博客](#)

阅读数 63万+

[博文](#) 来自: [帅地](#)

阅读数 21万+

[博文](#) 来自: [启航](#)

阅: 1万+



举报

阅: 1万+



何来我，一个双非本科弟弟，有幸在 19 届的秋招中得到前东家华为（以下简称 hw）的赏识，当时秋招签订就业协议…

博文 来自： [JavaEdge](#)

4

2

2

万+

万+

万+

万+

万+

万+

万+

万+

万+

万+

万+

万+

万+

万+

万+

万+

万+

万+

万+

万+

万+

万+

万+

万+

万+

万+

万+

万+

万+

万+

万+

万+

万+

万+



备案太麻烦？美国/香港云服务器-免备案，开年限量抢购3.5折

恒创科技-香港美国服务器，低至3.5折，CN2极速直连，外贸娱乐游戏行业都在用

强烈推荐10本程序员必读的书

很遗憾，这个春节注定是刻骨铭心的，新型冠状病毒让每个人的神经都是紧绷的。那些处在武汉的白衣天使们，尤其…

博文 来自： [沉默王二](#)

为什么说程序员做外包没前途？

之前做过不到3个月的外包，2020的第一天就被释放了，2019年还剩1天，我从外包公司离职了。我就谈谈我个人的…

博文 来自： [dotNet全栈](#)

B 站上有哪些很好的学习资源？

哇说起B站，在小九眼里就是宝藏般的存在，放年假宅在家时一天刷6、7个小时不在话下，更别提今年的跨年晚会，…

博文 来自： [九章算法的](#)

昂，我24岁了

24岁的程序员，还在未来迷茫，不知道能不能买得起房子

博文 来自： [敖丙](#)

新来个技术总监，禁止我们使用Lombok！

我有个学弟，在一家小型互联网公司做Java后端开发，最近他们公司新来了一个技术总监，这位技术总监对技术细节…

博文 来自： [HollisChuang's Blog](#)

阅读量 3万+

备案太麻烦？免备案 美国/香港云服务器，开年限量抢购3.5折

恒创科技-香港美国服务器低至3.5折，CN2极速直连，外贸娱乐游戏行业都在用

字节跳动的技术架构

字节跳动创立于2012年3月，到目前仅4年时间。从十几个工程师开始研发，到上百人，再到200余人。产品线由内涵…

博文 来自： [作一个独立连续的…](#)

阅读量 2万+

在三线城市工作爽吗？

我是一名程序员，从正值青春年华的 24 岁回到三线城市洛阳工作，至今已经 6 年有余。一不小心又暴露了自己的实…

博文 来自： [沉默王二](#)

阅读量 8万+

这些插件太强了，Chrome 必装！尤其程序员！

推荐 10 款我自己珍藏的 Chrome 浏览器插件

博文 来自： [沉默王二](#)

阅读量 1万+

抱歉，我觉得程序员副业赚钱并不靠谱

我最近看到不少关于程序员副业赚钱的文章，其中出的点子有这些：1. 在网上找项目做兼职2. 录制课程，到网上平台…

博文 来自： [码农翻身](#)

阅读量 9068

学Python后到底能干什么？网友：我太难了

感觉全世界营销文都在推Python，但是找不到工作的话，又有哪个机构会站出来给我推荐工作？笔者冷静分析多方…

博文 来自： [CSDN学院](#)

阅读量 4706

反转！“只问了1个框架，就给了35K的Python岗”

学Python的程序员建议收藏！

Java C语言 Python C++ C# Visual Basic .NET JavaScript PHP SQL Go语言 R语言 Assembly language Swift Ruby
MATLAB PL/SQL Perl Visual Basic Objective-C Delphi/Object Pascal Unity3D

©2019 CSDN 皮肤主题: 大白 设计师: CSDN官方博客



YuanruiZJU

[TA的个人主页>](#)

原创 15 粉丝 28 获赞 7 评论 5 访问 3万+

等级: [博客](#) 周排名: 19万+

积分: 563 总排名: 14万+

4

2

2

万+

万+

万+

关注

私信



4



2



亿速云高防服务器低延迟免备案

亿速云高防服务器，疫情期间，免费！
前10大游戏公司CEO鼎力支持，速度！
亿速云

最新文章

QEMU中协程的实现和使用

GPU虚拟化的评价标准与实现策略

qemu中的Hbitmap数据结构

QEMU中如何定义所有Device的基类和BUS
的基类

Mesos: A Platform for Fine-Grained
Resource Sharing in the Data Center（论
文译文）

分类专栏



QEMU源码分析系列

10篇



Linux操作系统学习

1篇



虚拟化技术

4篇



QEMU-KVM虚拟机使用

1篇



分布式计算

1篇

归档

2017年1月

2篇

2016年12月

14篇

2016年5月

1篇

热门文章

QEMU虚拟机源码分析概论

阅读数 4680

QEMU中的CPU类型设计

阅读数 4613

QEMU中的内存管理介绍

阅读数 4599

QEMU中的对象模型——QOM（介绍篇）

阅读数 4183

虚拟化I/O qos——mClock算法介绍

阅读数 3377

最新评论

虚拟化I/O qos——mCloc...

tunbahuang: 您好，问一下ceph中的mclock_op
class配置，这个现在生效吗？根据单个osd的i ...



举报



主赞

qemu如何实现面向对象模型QOM...
kunli4558：接口没太听懂

qemu如何实现面向对象模型QOM...
u011364612：欢迎大家提出意见！

QEMU中的对象模型——QOM（介...
u011364612：希望大家提出宝贵意见

亿速云高防服务器送防防CC

亿速云高防服务器，20+行业领袖祁
大游戏公司CEO鼎力支持，速度快和

打开

🗨️ QQ客服

✉️ kefu@csdn.net

💬 客服论坛

☎️ 400-660-0108

工作时间 8:30-22:00

关于我们 招聘 广告服务 网站地图

京ICP备19004658号 经营性网站备案信息

🚓 公安备案号 11010502030143

©1999-2020 北京创新乐知网络技术有限公司

司 网络110报警服务

北京互联网违法和不良信息举报中心

中国互联网举报中心 家长监护

版权与免责声明 版权申诉



4



2



举报

