1. Fuchsia 构建与运行
   1. 下载需要的组件
   2. 构建
   3. 运行
   4. Fuchsia的使用操作
2. Fuchsia简介与面向对象
   1. 简介
   2. 目录结构
      1. 各目录的作用
      2. 目录的面向对象
   3. Doublylinkedlist 介绍
   4. c++11新特性
3. 启动过程分析

以qemu-arm64为例。

* 1. 启动流程，函数执行顺序
  2. Start.S分析
  3. Main.c分析

1. VmAspace

管理kernel和用户空间。

* 1. is\_aslr\_enabled(),what is aslr?
  2. 结构

VmAspace使用特定的架构的类实例。实例自身会调用自身init()为地址空间分配translation talbe从内核空间。(分两种情况：一是初始化内核空间，translation table是静态分配的，二是初始化用户空间，translation table动态分配的。)

1. RefPtr的Ｔ类型应当继承自RefCounted。
2. Page table entry uses 8bytes.

FAQ:

1. mxtl::atomic\_int 实现原理。
2. enum 的声名函义：

enum memory\_order : int {

memory\_order\_relaxed = \_\_ATOMIC\_RELAXED,

memory\_order\_consume = \_\_ATOMIC\_CONSUME,

memory\_order\_acquire = \_\_ATOMIC\_ACQUIRE,

memory\_order\_release = \_\_ATOMIC\_RELEASE,

memory\_order\_acq\_rel = \_\_ATOMIC\_ACQ\_REL,

memory\_order\_seq\_cst = \_\_ATOMIC\_SEQ\_CST,

};

1. \_\_atomic\_thread\_fence含义
2. mxtl::RefCounted 实现
3. vending weak pointers ?
4. is\_class<T> 实现
5. mxtl::RefPtr实现
6. \_\_atomic\_thread\_fence

https://gcc.gnu.org/onlinedocs/gcc/\_005f\_005fatomic-Builtins.html

1. mxtl::internal::has\_mxtl\_recycle<> uses is\_same<> in template Recyclable<>? How to understand the meaning of is\_same<>?