

# 一个更完备的入门计划

## 1. HBA.ppt

- 先综述性的看我们在干什么，这个PPT概述了我们之前所做的工作，这里是最基本的故事线
- 但是这是非常老旧的内容了，拿到该计划的同学或同事不太需要关注该PPT中**SASI/DMA的IP细节以及之后的内容**

## 2. Linux内核IO栈

- 给非存储专业看的Linux内核IO栈的入门

## 3. howhba2.pdf

- 描述了我们第二阶段如何参与到HBA的开发中来，这里面的内容是纲领性的，相当于目录了，以这里作为**核心**来把握
- 该文档的末尾有提供两个参考，可以关注一下
  - 缩略词全称的汇总
  - 一些网络好文的link

## 4. howhba2.pdf中涉及了大量的pdf资料与文档，这些可以先浅浅的过一下，然后需要动手完成的就是**开发环境**的搭建以及入门任务（`student.pdf`）的完成

# 如何参与到HBA/RAID卡驱动组的开发中

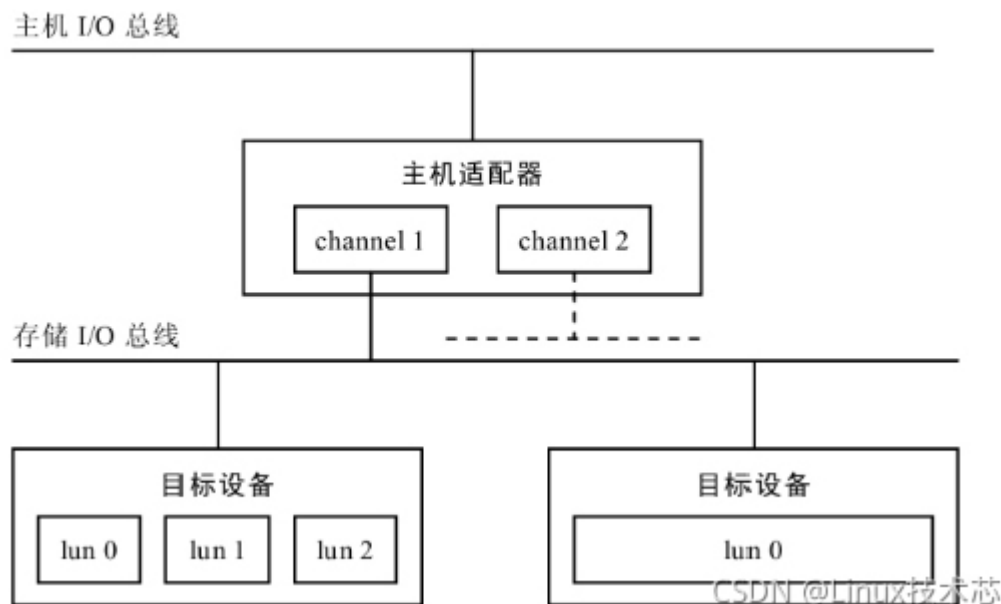
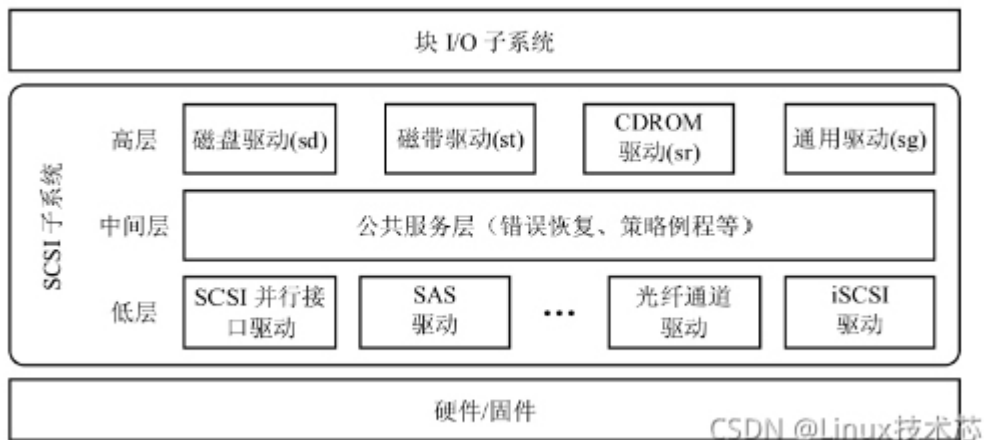
- 这是我们第二阶段的开发任务，基础知识类的学习理解在说明中都有，在该文档中不会涉及太多基础知识的细节
- 我们的产品现在主要对标博通9560 RAID卡，硬件在趋同，所以在软件的开发上我们可以参考博通的开源代码，**我们这里主要是驱动代码部分**

## 1. 时间线 —— 简单的包括第一阶段

1. HBA/RAID卡大多数是插在主机PCIe插槽上的，所以HBA/RAID驱动本质上是一个**PCIe设备的驱动**，这类卡涉及2类总线（**PCIe总线，SCSI总线**），2-3类协议（**PCIe协议，SCSI协议，NVMe协议**）
  - 博通的HAB/RAID卡后续可以接**SATA/SAS盘**，可能还有**NVMe盘**，视具体型号来定
2. 其次，这里我们看到了很多新的概念（**PCIe/SATA/SAS/SCSI/NVMe等等**）。所以接下来做了什么事呢？接下来就重点理解**PCIe/SCSI等协议的含义以及部分关键代码的实现**

- Linux kernel中会有PCIe枚举以及初始化的代码等等
- 3. 然后又去学习**SAS/ATA/IDE**这些内容的**发展历程**以及其**含义**
  - **SAS/SATA/SCSI**这些内容除了硬件代码不开源，其它资料还基本都是公开的，网上也能看到很多的资料，当然一些正版资料需要购买
- 4. 理解这些概念的同时，对**商用成熟HBA/RAID卡**进行了大量的测试
  - 测试工具主要包括 `fio/gfio/iostat/dd`，这些都是对IO子系统测试的一些常用工具，其中fio的开发者是linux内核块层维护者
    - 理解这些工具的基本原理以及熟练使用这些工具都是基本要求，以及对IO子系统常见参数的理解一定要到位，这些内容也都有对应的文档参考
  - 期间涉及这些工具所配置参数对性能的影响等等，当然作为驱动组，我们需要**深入理解linux内核代码**，这些参数以及配置我们是需要知道并理解原理的
- 5. 第一阶段的IP细节这里就省略了，我们直接进入到**第二阶段**。当然还是聚焦我们驱动组所关注的内容
- 6. HBA/RAID卡在硬件形式上不会有太大的改变，依然是PCIe插槽上的PCIe设备，但是对于一个裸金属来说，驱动作为与硬件最近的软件，驱动长啥样取决于跑在该裸金属上的内容是什么，具体我们需要cover以下的这些形式，以及我们目前的进展，**以及每一部分我们需要做到什么程度**
  - Linux —— 驱动开源
    - Linux内核操作系统精通，全面理解IO栈（文件系统，块层，驱动层，SCSI驱动，NVMe驱动...），能够独立开发debug内核，理解Linux内核启动流程，搭建Linux开发调试环境，Linux内核代码随时放在手边...
      - 更细节的内容在后续文档中**均有体现**
  - FreeBSD —— 驱动开源
    - 忽略吧，哈哈，占比少，而且开源，到了这一步再说
  - BIOS —— 博通官卡可以得到二进制代码，可以反汇编
    - 理解BIOS存在的意义，理解UEFI标准都规定了哪些内容，理解UEFI的总线设备驱动模型，理解x86指令集下的二级制
    - 能够熟练使用多种反汇编工具，伪c代码转为C代码
    - 熟练搭建BIOS驱动开发环境，理解edk2源码结构以及编译操作原理等等
  - Microsoft —— 博通官方的二进制驱动可以反汇编得到c代码，完整的发布需要Microsoft支持
    - 实施阶段，到时候我希望大家都基本理解Windows驱动的开发流程与步骤，以及开发环境的搭建等等
  - VMware —— 需要VMware支持，还未启动
- 7. 此外作为相对新兴的一些领域与方向包括 —— 用户态SCSI存储系统，IO虚拟化与用户态结合的探索等等

## 2. 以Linux为例，一句话概括一下 —— 很关键



- 我们打交道的代码是linux kernel中的SCSI层，**上层**可以理解为sd驱动，**中层**是通用逻辑，**底层**是我们的HBA卡的驱动（本质是PCIe驱动），这里**把握两个总线**
  - 我们的HBA卡是PCIe设备，向PCIe总线注册。对应的驱动就是我们写的HBA卡的驱动，即SCSI三层中的底层
  - 而HBA卡后面所接的盘是SCSI设备，它应该向SCSI总线注册，而什么驱动向SCSI总线注册呢，sd（以盘为例哈），即SCSI三层中的上层
    - **也正是sd驱动能够让用户在/dev下看到sda，随后用户才可以做文件系统或者使用该存储设备**
- ref
  - [https://blog.csdn.net/sinat\\_37817094/article/details/120357371](https://blog.csdn.net/sinat_37817094/article/details/120357371)
    - 深入浅出SCSI子系统

### 3. 文档类资料

- `git@github.com:LeapIO/wiki.git`
  - 还算有一些内容，但是维护没跟上，可以浅浅的作为一个参考

### 4. 基本学习流程 —— 这些md都在附件中 —— 代码部分第一次看都不需要一行一行的看，循序渐进多看几次

#### 纯计算机基础知识 —— 排列顺序并非阅读顺序

- `asm.pdf` —— x86寄存器使用惯例
- linux下的文本处理三剑客 —— `awk/sed/grep`
- **`gdb`的使用**
- **`git`的使用**
- 主要代码编写阅读工具，**强烈推荐vscode**
- `intro.pdf` —— 内核开发入门
- `kernel-init.pdf` —— OS初始化相关
- `vfs.pdf` —— 虚拟文件系统相关
- `NoBlockOrBlockorAsyncOrSync.pdf` —— 同步异步阻塞非阻塞
- `better-use-linux.pdf` —— 如何更好的使用linux
- `csapp.pdf` —— **程序员的自我修养**以及**CSAPP**两本书的部分内容的学习笔记

#### 我们领域相关的基础知识

- `sas.pdf` —— SATA/SAS基本概念入门以及基本发展等等
- `pcie.pdf` —— PCIe基础知识
- `dma.pdf` —— DMA基本原理
- `interrupt.pdf` —— 中断基本原理
- `io.pdf` 以及 `iostack.drawio.html`
- `fio.pdf` 以及 `MobaXterm图形化使用gfio指南.pdf` —— `fio`源码分析以及如何让你更好的利用好`fio`以及`gfio`等等
- `ssdfans.pdf` —— SSD以及NVMe入门
- BIOS相关
  - `ing`
- vmware相关
  - `iov.pdf` 以及 `v.pdf` —— IO虚拟化的入门
- Windows相关

- ing

## 一些ppt

- megaraid大概流程说明.pptx
  - megaraid驱动基本流程
- Uefi驱动模型.pptx
- Linux内核IO栈.pptx —— 非存储专业的
- chatgpt简单使用.pptx

## 5. 基本开发环境的搭建 —— 一个是linux内核的开发环境，一个是bios的开发环境，这两个其实是可以合并在一起的，充分理解一些系统是如何跑起来的

### 简单介绍一下这个环境是怎么回事 —— 既是学习环境，也是开发环境，也是生产环境

- 正常来说，对于计算机启动而言，从BIOS阶段到Linux内核（**假设OS是Linux哈**）启动阶段，直到我们能够看到计算机启动成功。这个过程涉及了大量的知识点，大量的计算机原理等等。如果能够清晰的看到这个过程，那么对于理解这个过程来说将相对容易。所以有没有办法将这些过程都展示给我们看呢，当然有了，那就是qemu模拟器。qemu是模拟器，它模拟了硬件，但是OS以及BIOS的代码都是与真机一致的。通过qemu模拟的方式我们就可以对从**BIOS加载到OS启动的过程一览无余**。这里简单介绍一下其基本过程，相关原理以及细节均位于以上的pdf等相关资料中
  - 首先需要一份**Linux内核源码**，内核源码位置在这里 ——  
<https://github.com/doubleDDDD/linux.git> —— **版本建议选择5.15**
    - 内核源码的编译等教程位于 intro.pdf （主要是讲内核开发入门的），内容包括但不限于，有些内容可能是其它文档中的哈
    - Linux内核源码的目录结构，如何配置内核，源码的编译连接过程，每一步都有什么产物，与Linux目录下的一些文件夹中的内容都有什么对应关系，内核的编译安装本

质都是在干什么

- 其中也包含了很多的概念

- ramdisk/initrd.img/grub/内核模块/内建模块/内核模块应该如何开发等等

- 内核源码编译后的产物，可以简单的理解为**Linux内核编译完之后的可执行文件，假设叫做bzImage**，具体是哪个文件，具体是什么形式，文档中都有的哈

- 这个可执行文件将作为qemu的参数使用，一个示意的用法是 `qemu bzImage`，然后呢你就相当于起了一个操作系统，但是这个操作系统相对于你的物理机来说是一个进程，是不是就可以gdb调试，是不是就能看到内核中每一条代码的执行等等

- 其实，qemu其实模拟的一个完备的硬件系统。一个类似这样简单的示意用法，`qemu bzImage` 其实qemu会启动默认的BIOS，即seaBIOS，那当我们想要学习开发BIOS的时候应该做什么样的操作呢。与Linux内核代码类似。当前的BIOS都遵循UEFI协议，虽然4大BIOS厂商的BIOS并非开源代码，但是有一个开源代码，叫做edk2，edk2是UEFI协议的其中一个开源实现，由intel主导的。实际上很多闭源BIOS也都是基于edk2开发的。闭源是商业上的问题，开源咱就可以学习技术上的问题。

- 基本原理同Linux内核 BIOS源码在这里 —— [git@github.com:tianocore/edk2.git](https://github.com/tianocore/edk2) —— **你可以使用最新的稳定版，一定选稳定版编译哦**

- 相关文档比较乱，我整理一下再给你，你做到这个地方我还没给你，要问我要哈

- edk2的源码也是C代码，需要编译出一个bios的可执行文件（**客观的不是这个叫法哈**）出来，但是代码组织，代码编译等方式与Linux内核代码并不相同，这里也可以浅浅的学习一下

- 在Linux内核可执行程序的环境下，示意用法变成了这样 `qemu bzImage bios.img`，这样就能够追到BIOS以及Linux内核代码的启动过程的每一行程序

- 具体开发环境的搭建参考下面git的README

- [git@github.com:LeapIO/debug\\_kernel.git](https://github.com/LeapIO/debug_kernel)

- 此外，还有一个知识点需要强调，即PCIe设备的穿透，这个涉及到虚拟化等概念，即将一个真实的硬件设备透传到某一个虚拟机中，完全供改虚拟机使用

- 这个涉及到我们**手把手**来搞哈

## 缩略词全称（无序供查）

1. IDE/ATA/SATA
  - IDE: Integrated Drive Electronics
  - ATA: Advanced Technology Attachment
  - PATA: Parallel Advanced Technology Attachment
2. SATA: Serial Advanced Technology Attachment
3. SCSI: Small Computer System Interface
4. SAS: Serial Attached SCSI
5. AHCI: Advanced Host Controller Interface
6. tmf: Task Management Function
7. SSP: Serial SCSI Protocol
8. SMP: SAS Management Protocol
9. STP: Serial ATA tunneling protocol
10. I\_T\_nexus: I\_T nexus is a relationship between a **SCSI Initiator Port** and a **SCSI Target Port**
11. XFER: transfer
12. DMAE: dma engine
13. TF: task info
14. MIG: Memory Interface Generator
15. IAF: IDENTIFY Address Frame
16. OAF: OPEN Address Frame
17. GPIO: general-purpose input/output
18. SQ: submit queue
19. CQ: complete queue
20. IU: information unit
21. DL: data list
22. XRDY: transfer ready
23. RES: Response
24. TD: task data
25. FIS: Frame Information Structure

## 关键link汇总（无序供查）

- <https://blog.51cto.com/wushank/1708168>
  - SAS/SATA/SSD测试好文

- [https://linuxtools-rst.readthedocs.io/zh\\_CN/latest/tool/iostat.html](https://linuxtools-rst.readthedocs.io/zh_CN/latest/tool/iostat.html)
  - Linux Tools Quick Tutorial
- <http://www.webkf.net/article/21/212.html> && <http://www.webkf.net/article/21/162.html>
  - SAS接口完全互联指南
- <https://blog.csdn.net/kunkliu/article/details/108884882>
  - XDMA驱动初始化流程
- <https://docs.xilinx.com/api/khub/documents/gU5oVrEmGVabn9W5b5r4FQ/content?Ft-Calling-App=ft%2Fturnkey-portal&Ft-Calling-App-Version=3.11.25&filename=pg195-pcie-dma.pdf>
  - XDMA文档
- `linux/include/scsi/scsi_proto.h`
  - `#define INQUIRY 0x12`
  - ...
- [https://en.wikipedia.org/wiki/SCSI\\_command](https://en.wikipedia.org/wiki/SCSI_command)
  - SCSI command wiki
  - 可以快速查阅scsi命令的opcode，比如能够快速查到0x12代表的是inquiry
- <https://www.t10.org/ftp/t10/document.04/04-136r0.pdf>
  - SCSI to ATA Command Translations
- <https://www.t10.org/ftp/t10/document.02/02-158r1.pdf>
  - Serial Attached SCSI Technical Overview
- [https://www.seagate.com/files/staticfiles/support/docs/manual/Interface manuals/100293068j.pdf](https://www.seagate.com/files/staticfiles/support/docs/manual/Interface%20manuals/100293068j.pdf)
  - scsi cmd手册，即每一条cmd的细节
- <http://softdown.elecfans.net/uploads/soft/171029/2387123-1G02Z93457.pdf?filename=>
  - Linux那些事儿系列丛书之**我是SCSI硬盘**
- <http://blog.chinaunix.net/uid-709830-id-2069884.html>
  - Linux scsi 异步扫描分析笔记
- <http://www.4k8k.xyz/article/jasonactions/116453458>
  - SCSI子系统基础学习笔记 —— SCSI设备探测
- [https://blog.csdn.net/flyingnosky/category\\_11474453.html](https://blog.csdn.net/flyingnosky/category_11474453.html)
  - CSDN —— block scsi专栏
- <https://blog.csdn.net/flyingnosky/article/details/121717421>
  - SCSI错误处理流程
- <https://blog.csdn.net/wangww631/article/details/78798637>
  - 块设备内核参数max\_segments和max\_sectors\_kb解析
- <https://blog.51cto.com/wendashuai/2585401>
  - linux磁盘队列深度nr\_requests和queue\_depth
- [https://en.wikipedia.org/wiki/SCSI\\_check\\_condition](https://en.wikipedia.org/wiki/SCSI_check_condition)
  - SCSI check condition
- <https://www.daimajiaoliu.com/daima/479a00d39100400>



- Libata错误信息解析
- [http://www.wowotech.net/irq\\_subsystem/request\\_threaded\\_irq.html](http://www.wowotech.net/irq_subsystem/request_threaded_irq.html)
  - Linux kernel中断子系统之（五）：驱动申请中断API
- [https://blog.csdn.net/Chris\\_Tsai/article/details/80869304?utm\\_medium=distribute.pc\\_relevant.none-task-blog-2\\_defaultbaidujs\\_title~default-0.pc\\_relevant\\_default&spm=1001.2101.3001.4242.1&utm\\_relevant\\_index=3](https://blog.csdn.net/Chris_Tsai/article/details/80869304?utm_medium=distribute.pc_relevant.none-task-blog-2_defaultbaidujs_title~default-0.pc_relevant_default&spm=1001.2101.3001.4242.1&utm_relevant_index=3)
  - module\_init()能够执行的原因
- <https://stackoverflow.com/questions/32334870/how-to-do-a-true-rescan-of-pcie-bus>
  - 关于PCIe的rescan
- <https://www.docin.com/p-2270323962.html>
  - MPTSAS源码阅读笔记
- <https://blog.csdn.net/zancijun1666/article/details/83387135>
  - sas控制器驱动结构粗探
- [https://blog.csdn.net/qq\\_21186033/article/details/116975283](https://blog.csdn.net/qq_21186033/article/details/116975283)
  - ACPI PCI枚举
- [https://blog.csdn.net/weixin\\_39540178/article/details/111196263](https://blog.csdn.net/weixin_39540178/article/details/111196263)
  - PCIe驱动分析。提出了需要掌握以下知识
    - PCI协议，BIOS对ACPI表格的初始化，X86的IO空间，X86某些古老设备的向前兼容，以及BIOS对PCI的枚举过程等等
    - 重点研究非ACPI方式枚举，主要看pci\_legacy\_init和pcibios\_init
- <https://blog.csdn.net/yhb1047818384/article/details/106676548>
  - 结合了ACPI与PCIe驱动的初始化
  - ACPI部分讲的不错，后面介绍了资源分配
- [https://www.cxyzjd.com/article/qq\\_39376747/112723350](https://www.cxyzjd.com/article/qq_39376747/112723350)
  - 靠谱，基于它总结学习
- <https://www.cnblogs.com/szhhb-5251/p/11620310.html>
  - PCIe枚举，深度好文
    - 在启动阶段，只有PCI host被告知了CPU
    - BIOS代码在枚举设备的时候，会去读每个设备的VID，判断读到的是不是FFFF，若为FFFF,则表示设备不存在。否则设备存在
    - DID&VID是出厂时就固定在PCIE设备配置空间中的数据，表示每个不同的设备，同理，我们在BIOS中也可通过判断这个ID值来寻找指定的PCIE设备
    - 所有的PCIe设备不会通过ACPI报告，ACPI只会报告 root host bridge
    - 剩下的设备需要PCIe驱动主动探测，能读到内容则说明有设备
    - HBA扫描后端target的拓扑结构时也是一样的方式
- <http://www.ssfans.com/?s=老男孩读PCIe>
  - 老男孩读PCIe

- [https://blog.csdn.net/u013253075/article/details/119485466?spm=1001.2101.3001.6661.1&utm\\_medium=distribute.pc\\_relevant\\_t0.none-task-blog-2~default~CTRLIST~default-1.no\\_search\\_link&depth\\_1-utm\\_source=distribute.pc\\_relevant\\_t0.none-task-blog-2~default~CTRLIST~default-1.no\\_search\\_link&utm\\_relevant\\_index=1](https://blog.csdn.net/u013253075/article/details/119485466?spm=1001.2101.3001.6661.1&utm_medium=distribute.pc_relevant_t0.none-task-blog-2~default~CTRLIST~default-1.no_search_link&depth_1-utm_source=distribute.pc_relevant_t0.none-task-blog-2~default~CTRLIST~default-1.no_search_link&utm_relevant_index=1)
  - BAR的配置过程
- <https://stackoverflow.com/questions/30190050/what-is-the-base-address-register-bar-in-pcie>
  - 一下就把bar空间理解了
- <https://cromwell-intl.com/open-source/sysfs.html>
  - sysfs
- <https://linux.die.net/man/8/lspci>
  - lspci man page
- <http://www.ssdfans.com/?p=8210>
  - 老男孩读PCIe系列
- <https://www.techtarget.com/searchstorage/definition/IDE>
  - SATA的发展
- <https://unix.stackexchange.com/questions/144561/in-what-sense-does-sata-talk-scsi-how-much-is-shared-between-scsi-and-ata>
  - SAS (Series SCSI) 的发展以及与ATA的兼容合并
- [https://www.mindshare.com/files/ebooks/SATA Storage Technology.pdf](https://www.mindshare.com/files/ebooks/SATA%20Storage%20Technology.pdf) && [https://web.archive.org/web/20161009182351if\\_/http://www.ece.umd.edu/courses/enee759h.S2003/references/serialata10a.pdf](https://web.archive.org/web/20161009182351if_/http://www.ece.umd.edu/courses/enee759h.S2003/references/serialata10a.pdf) && <https://pdfroom.com/books/sata-storage-technology-serial-ata/0q2JQeN8gxE>
  - SATA协议
- [https://www.intel.com/content/dam/www/public/us/en/documents/technical-specifications/serial-ata-ahci-spec-rev1\\_1.pdf](https://www.intel.com/content/dam/www/public/us/en/documents/technical-specifications/serial-ata-ahci-spec-rev1_1.pdf) && [https://www.intel.com/content/dam/www/public/us/en/documents/technical-specifications/serial-ata-ahci-spec-rev1\\_3.pdf](https://www.intel.com/content/dam/www/public/us/en/documents/technical-specifications/serial-ata-ahci-spec-rev1_3.pdf)
  - ACPI spec
- [https://wiki.osdev.org/ATA\\_Command\\_Matrix](https://wiki.osdev.org/ATA_Command_Matrix)
  - ATA cmd
- SAS Storage Architecture Serial Attached SCSI by Mike Jackson
  - SAS参考书
- <https://www.seagate.com/staticfiles/support/disc/manuals/sas/100293071b.pdf>
  - Serial Attached SCSI (SAS) Interface Manual Users Guide
- <https://www.snia.org/sites/default/education/tutorials/2007/spring/networking/SAS-Overview.pdf>
  - SAS Overview PDF
- <https://www.t10.org/ftp/t10/document.06/06-002r2.pdf>

- SCSI addr
- <https://www.snia.org/sites/default/education/tutorials/2007/spring/networking/SAS-Overview.pdf>
  - SAS协议类型