SCSI Interfaces Guide

Author: James Bottomley
Author: Rob Landley

Introduction

Protocol vs bus

Once upon a time, the Small Computer Systems Interface defined both a parallel I/O bus and a data protocol to connect a wide variety of peripherals (disk drives, tape drives, modems, printers, scanners, optical drives, test equipment, and medical devices) to a host computer.

Although the old parallel (fast/wide/ultra) SCSI bus has largely fallen out of use, the SCSI command set is more widely used than ever to communicate with devices over a number of different busses.

The SCSI protocol is a big-endian peer-to-peer packet based protocol. SCSI commands are 6, 10, 12, or 16 bytes long, often followed by an associated data payload.

SCSI commands can be transported over just about any kind of bus, and are the default protocol for storage devices attached to USB, SATA, SAS, Fibre Channel, FireWire, and ATAPI devices. SCSI packets are also commonly exchanged over Infiniband, I2O, TCP/IP (iSCSI), even Parallel ports.

Design of the Linux SCSI subsystem

The SCSI subsystem uses a three layer design, with upper, mid, and low layers. Every operation involving the SCSI subsystem (such as reading a sector from a disk) uses one driver at each of the 3 levels: one upper layer driver, one lower layer driver, and the SCSI midlayer.

The SCSI upper layer provides the interface between userspace and the kernel, in the form of block and char device nodes for I/O and ioctl(). The SCSI lower layer contains drivers for specific hardware devices.

In between is the SCSI mid-layer, analogous to a network routing layer such as the IPv4 stack. The SCSI mid-layer routes a packet based data protocol between the upper layer's /dev nodes and the corresponding devices in the lower layer. It manages command queues, provides error handling and power management functions, and responds to ioctl() requests.

SCSI upper layer

The upper layer supports the user-kernel interface by providing device nodes.

sd (SCSI Disk)

sd (sd_mod.o)

sr (SCSI CD-ROM)

sr (sr_mod.o)

st (SCSI Tape)

st (st.o)

sg (SCSI Generic)

sg (sg.o)

ch (SCSI Media Changer)

ch (ch.c)

SCSI mid layer

SCSI midlayer implementation

include/scsi/scsi_device.h

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\driver-api\(linux-master)(Documentation)(driver-api)scsi.rst, line

94)

Unknown directive type "kernel-doc".

```
.. kernel-doc:: include/scsi/scsi_device.h
   :internal:
```

drivers/scsi/scsi.c

Main file for the SCSI midlayer.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\driver-api\(linux-master)(Documentation)(driver-api)scsi.rst, line 102)

Unknown directive type "kernel-doc".

```
.. kernel-doc:: drivers/scsi/scsi.c
   :export:
```

drivers/scsi/scsicam.c

SCSI Common Access Method support functions, for use with HDIO_GETGEO, etc.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\driver-api\(linux-master)(Documentation)(driver-api)scsi.rst, line 112)

Unknown directive type "kernel-doc".

```
.. kernel-doc:: drivers/scsi/scsicam.c
   :export:
```

drivers/scsi/scsi_error.c

Common SCSI error/timeout handling routines.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\driver-api\(linux-master)(Documentation)(driver-api)scsi.rst, line 120)

Unknown directive type "kernel-doc".

```
.. kernel-doc:: drivers/scsi/scsi_error.c
   :export:
```

drivers/scsi/scsi_devinfo.c

Manage scsi_dev_info_list, which tracks blacklisted and whitelisted devices.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\driver-api\(linux-master)(Documentation)(driver-api)scsi.rst, line 129)

Unknown directive type "kernel-doc".

```
.. kernel-doc:: drivers/scsi/scsi_devinfo.c
   :internal:
```

drivers/scsi/scsi_ioctl.c

Handle ioctl() calls for SCSI devices.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\driver-api\(linux-master)(Documentation)(driver-api)scsi.rst, line 137)

Unknown directive type "kernel-doc".

```
.. kernel-doc:: drivers/scsi/scsi_ioctl.c
   :export:
```

drivers/scsi/scsi_lib.c

SCSI queuing library.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\driver-api\(linux-master)(Documentation)(driver-api)scsi.rst, line 145)

Unknown directive type "kernel-doc".

```
.. kernel-doc:: drivers/scsi/scsi_lib.c
   :export:
```

drivers/scsi/scsi_lib_dma.c

SCSI library functions depending on DMA (map and unmap scatter-gather lists).

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\driver-api\(linux-master)(Documentation)(driver-api)scsi.rst, line 154)

Unknown directive type "kernel-doc".

```
.. kernel-doc:: drivers/scsi/scsi_lib_dma.c
   :export:
```

drivers/scsi/scsi_proc.c

The functions in this file provide an interface between the PROC file system and the SCSI device drivers It is mainly used for debugging, statistics and to pass information directly to the lowlevel driver. I.E. plumbing to manage /proc/scsi/*

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\driver-api\(linux-master)(Documentation)(driver-api)scsi.rst, line 165)

Unknown directive type "kernel-doc".

```
.. kernel-doc:: drivers/scsi/scsi_proc.c
   :internal:
```

drivers/scsi/scsi_netlink.c

Infrastructure to provide async events from transports to userspace via netlink, using a single NETLINK_SCSITRANSPORT protocol for all transports. See the original patch submission for more details.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\driver-api\(linux-master)(Documentation)(driver-api)scsi.rst, line 177)

Unknown directive type "kernel-doc".

```
.. kernel-doc:: drivers/scsi/scsi_netlink.c
   :internal:
```

drivers/scsi/scsi_scan.c

Scan a host to determine which (if any) devices are attached. The general scanning/probing algorithm is as follows, exceptions are made to it depending on device specific flags, compilation options, and global variable (boot or module load time) settings. A specific LUN is scanned via an INQUIRY command; if the LUN has a device attached, a scsi_device is allocated and setup for it. For every id of every channel on the given host, start by scanning LUN 0. Skip hosts that don't respond at all to a scan of LUN 0. Otherwise, if LUN 0 has a device attached, allocate and setup a scsi_device for it. If target is SCSI-3 or up, issue a REPORT LUN, and scan all of the LUNs returned by the REPORT LUN; else, sequentially scan LUNs up until some maximum is reached, or a LUN is seen that cannot have a device attached to it.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\driver-api\(linux-master)(Documentation)(driver-api)scsi.rst, line 196)

Unknown directive type "kernel-doc".

```
.. kernel-doc:: drivers/scsi/scsi_scan.c
   :internal:
```

drivers/scsi/scsi_sysctl.c

Set up the sysctl entry: "/dev/scsi/logging_level" (DEV_SCSI_LOGGING_LEVEL) which sets/returns scsi_logging_level.

drivers/scsi/scsi_sysfs.c

SCSI sysfs interface routines.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\driver-api\(linux-master)(Documentation)(driver-api)scsi.rst, line 210)

Unknown directive type "kernel-doc".

```
.. kernel-doc:: drivers/scsi/scsi_sysfs.c
   :export:
```

drivers/scsi/hosts.c

mid to lowlevel SCSI driver interface

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\driver-api\(linux-master)(Documentation)(driver-api)scsi.rst, line 218)

Unknown directive type "kernel-doc".

```
.. kernel-doc:: drivers/scsi/hosts.c
   :export:
```

drivers/scsi/scsi_common.c

general support functions

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\driver-api\(linux-master)(Documentation)(driver-api)scsi.rst, line 226)

Unknown directive type "kernel-doc".

```
.. kernel-doc:: drivers/scsi/scsi_common.c
   :export:
```

Transport classes

Transport classes are service libraries for drivers in the SCSI lower layer, which expose transport attributes in sysfs.

Fibre Channel transport

The file drivers/scsi/scsi_transport_fc.c defines transport attributes for Fibre Channel.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\driver-api\(linux-master)(Documentation)(driver-api)scsi.rst, line 241)

Unknown directive type "kernel-doc".

```
.. kernel-doc:: drivers/scsi/scsi_transport_fc.c
   :export:
```

iSCSI transport class

The file drivers/scsi/scsi_transport_iscsi.c defines transport attributes for the iSCSI class, which sends SCSI packets over TCP/IP connections.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\driver-api\(linux-master)(Documentation)(driver-api)scsi.rst, line 251)

Unknown directive type "kernel-doc".

```
.. kernel-doc:: drivers/scsi/scsi_transport_iscsi.c
   :export:
```

Serial Attached SCSI (SAS) transport class

The file drivers/scsi/scsi_transport_sas.c defines transport attributes for Serial Attached SCSI, a variant of SATA aimed at large high-end systems.

The SAS transport class contains common code to deal with SAS HBAs, an aproximated representation of SAS topologies in the driver model, and various sysfs attributes to expose these topologies and management interfaces to userspace.

In addition to the basic SCSI core objects this transport class introduces two additional intermediate objects: The SAS PHY as represented by struct sas_phy defines an "outgoing" PHY on a SAS HBA or Expander, and the SAS remote PHY represented by struct sas_rphy defines an "incoming" PHY on a SAS Expander or end device. Note that this is purely a software concept, the underlying hardware for a PHY and a remote PHY is the exactly the same.

There is no concept of a SAS port in this code, users can see what PHYs form a wide port based on the port_identifier attribute, which is the same for all PHYs in a port.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\driver-api\(linux-master)(Documentation)(driver-api)scsi.rst, line 278)

Unknown directive type "kernel-doc".

```
.. kernel-doc:: drivers/scsi/scsi_transport_sas.c
   :export:
```

SATA transport class

The SATA transport is handled by libata, which has its own book of documentation in this directory.

Parallel SCSI (SPI) transport class

The file drivers/scsi/scsi_transport_spi.c defines transport attributes for traditional (fast/wide/ultra) SCSI busses.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\driver-api\(linux-master)(Documentation)(driver-api)scsi.rst, line 293)

Unknown directive type "kernel-doc".

```
.. kernel-doc:: drivers/scsi/scsi_transport_spi.c
   :export:
```

SCSI RDMA (SRP) transport class

The file drivers/scsi/scsi_transport_srp.c defines transport attributes for SCSI over Remote Direct Memory Access.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\driver-api\(linux-master)(Documentation)(driver-api)scsi.rst, line 302)

Unknown directive type "kernel-doc".

```
.. kernel-doc:: drivers/scsi/scsi_transport_srp.c
   :export:
```

SCSI lower layer

Host Bus Adapter transport types

Many modern device controllers use the SCSI command set as a protocol to communicate with their devices through many different types of physical connections.

In SCSI language a bus capable of carrying SCSI commands is called a "transport", and a controller connecting to such a bus is called a "host bus adapter" (HBA).

Debug transport

The file drivers/scsi/scsi_debug.c simulates a host adapter with a variable number of disks (or disk like devices) attached, sharing a common amount of RAM. Does a lot of checking to make sure that we are not getting blocks mixed up, and panics the kernel if anything out of the ordinary is seen.

To be more realistic, the simulated devices have the transport attributes of SAS disks.

For documentation see http://sg.danny.cz/sg/sdebug26.html

todo

Parallel (fast/wide/ultra) SCSI, USB, SATA, SAS, Fibre Channel, FireWire, ATAPI devices, Infiniband, I2O, Parallel ports, netlink...