SMALL COMPUTER SYSTEM INTERFACE

Small Computer System Interface (SCSI) was for a long time *the* technology for I/O buses in Unix and PC servers, is still very important today and will presumably remain so for a good many years to come. The first version of the SCSI standard was released in 1986. Since then SCSI has been continuously developed in order to keep it abreast with technical progress.

SCSI basics

As a medium, SCSI defines a parallel bus for the transmission of data with additional lines for the control of communication. The bus can be realised in the form of printed conductors on the circuit board or as a cable. Over time, numerous cable and plug types have been defined that are not directly compatible with one another (Table 3.1). A so- called daisy chain can connect up to 16 devices together (Figure 3.3).

The SCSI protocol defines how the devices communicate with each other via the SCSI bus. It specifies how the devices reserve the SCSI bus and in which format data is transferred. The SCSI protocol has been further developed over the years. For example, a server could originally only begin a new SCSI command when the previous SCSI command had been acknowledged by the partner; however, precisely this overlapping of SCSI commands is the basis for the performance increase achieved by RAID (Section 2.4). Today it is even possible using asynchronous I/O to initiate multiple concurrently write or read commands to a storage device at the same time.

The SCSI protocol introduces SCSI IDs (sometimes also called target ID or just ID) and Logical Unit Numbers (LUNs) for the addressing of devices. Each device in the SCSI bus must have an unambiguous ID, with the HBA in the server requiring its own ID. Depending upon the version of the SCSI standard, a maximum of 8 or 16 IDs are permitted per SCSI bus. Storage devices such as RAID disk subsystems, intelligent disk subsystems or tape libraries can include several subdevices, such as virtual hard disks, tape drives or a media changer to insert the tapes, which means that the IDs would be used up very quickly. Therefore, so-called LUNs were introduced in order to address subdevices within larger devices (Figure 3.4). A server can be equipped with several SCSI controllers. Therefore, the operating system must note three things for the differentiation of devices – controller ID, SCSI ID and LUN.

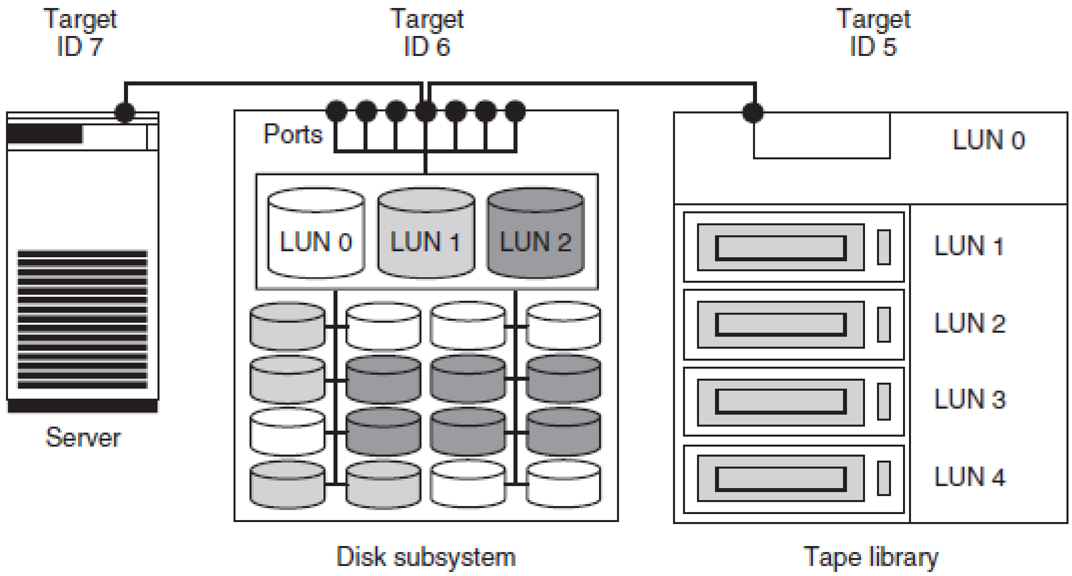

Figure 4.4 Devices on the SCSI bus are differentiated by means of target IDs. Components within devices (virtual hard disks, tape drives and the robots in the tape library) by LUNs.

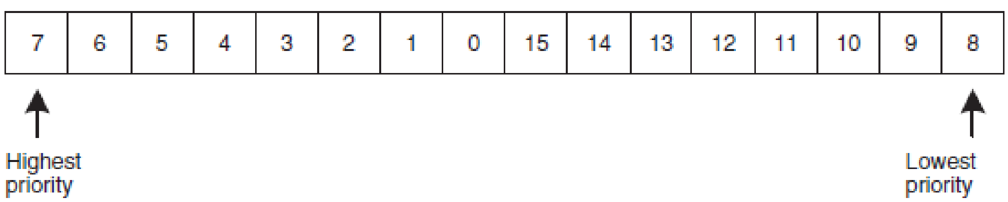

Figure 4.5 SCSI Target IDs with a higher priority win the arbitration of the SCSI bus. The priority of SCSI IDs is slightly trickier. Originally, the SCSI protocol permitted only eight IDs, with the ID '7' having the highest priority. More recent versions of the SCSI protocol permit 16 different IDs. For reasons of compatibility, the IDs '7' to '0' should retain the highest priority so that the IDs '15' to '8' have a lower priority (Figure 3.5). Devices (servers and storage devices) must reserve the SCSI bus (arbitrate) before they may send data through it. During the arbitration of the bus, the device that has the highest

priority SCSI ID always wins. In the event that the bus is heavily loaded, this can lead to devices with lower priorities never being allowed to send data. The SCSI arbitration procedure is therefore 'unfair'.

Source : http://elearningatria.files.wordpress.com/2013/10/cse-viii-storage-area-networks-06cs833-notes.pdf