The Universally-Silly Bus

Geoff Collyer

Plan 9 Foundation

ABSTRACT

USB is a disaster in terms of software complexity and interrupt load. It was supposed to establish a single connector standard and didn't even manage that. USB is a triumph of marketing over technical excellence.

1. Jack of All Trades

By attempting to solve all peripheral interface problems, USB has done a poor job of solving any of them, but is excessively complex in the attempt. (Shades of UEFI!) There are several classes of transfers to cope with all possible needs, including audio. For each device type (e.g., mouse, keyboard, audio, ether, disk) there is supposed to be a single driver required, yet in practice there are different device subtypes or just vendor-specifc nonsense, so it may be necessary to write a new USB driver for a new device, even of a 'known' type.

It doesn't help that there have been four not-quite-compatible host controller interfaces, each of which requires a substantial driver.

Instead of unifying device management, it has simply required creation of a whole parallel tree of interface and device driver support.

A thought experiment: imagine trying to shoe-horn USB support into a tiny bootstrap.

2. Failure of Engineering

USB manages to combine the worst of polled I/O and interrupt-driven I/O by requiring frequent polling, driven by interrupts. It doesn't even readily identify the interrupt cause(s). A common bug exhibited by USB device or interfaces is essentially non-stop interrupts at the maximum permitted rate, which is still probably not adequate for busy 10Gb/s Ethernet. One may end up sacrificing a CPU core to fielding largely-pointless USB interrupts.

References