

---

# Intelligent I/O

---

Operating System Seminar



Varsha Teckchandani

121060

Institute of Engineering and Technology - Ahmedabad University

## What is an I/O?

Input/output (I/O), in computing, is a communication process between a computer and the outside world.

Input refers to the signals or instructions sent to the computer. Output refers to the signals sent out from the computer. For instance, a keyboard can send physical information of the user's keystrokes, which the CPU accepts as input. That data is then processed and displayed on a computer monitor as output.

## Intelligent I/O Objective:

Input and output devices on computers, such as storage or network adapters, are accessed and controlled by the operating system through device-specific software known as device drivers. Recently, these I/O devices have become more intelligent and sophisticated and, consequently, their drivers have evolved from simple routines into highly complex, multi-layered programs. Because each operating system has its own unique I/O system and device interface, hardware vendors have been faced with the challenge of creating a different device driver for each operating system that they want their devices to support.

The objective of I2O is to provide a standards-based approach that complements existing drivers and offers a portable framework for the rapid development of a new generation of portable, I/O solutions.

## Intelligent I/O overview:

Intelligent I/O can be defined as a rule based model to perform I/O processing of the Operating system intelligently.

It follows a split driver architecture i.e. dividing the architecture logically into 2 sections

1. **Operating System Services (OSM):** Interfaces Operating System I/O infrastructure
2. **Hardware Device Module (HDM):** Interfaces with the device while running over I/O processor

I2O supports "Message Passing model" for communication between these two layers and also defines an API interface that works on multiple operating environments and I/O systems

Intelligent I/O focuses on a standard interface and permits complete implementation in user space applications that would normally require ad-hoc kernel modifications such as a Network File System. It also serves to implement features that would normally have to be embedded in an application, such as Automatic Format Conversion or Revision Tracking.

## Split Device Driver:

A device driver interfaces a particular hardware device to a specific operating system. As shown in Figure 1, the top portion of a device driver adapts the operating system calls into I/O transactions. The bottom portion of the driver contains vendor-specific code, which adapts the hardware level interface of the I/O adapter to the functions required for that particular class of device.

**For example,** all network adapters perform the same class-specific functions but have varying register interfaces.

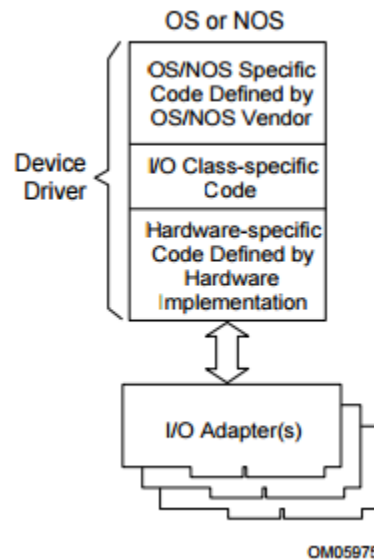


Figure 1: Typical I/O device driver

The I2O specification splits the device driver into two modules: one that contains all the OS specific code and the other for hardware-specific code. OS vendors need to produce only one specific module for each class of I/O device. Likewise, hardware vendors have to produce only a single version of the hardware device module for an I/O adapter. The device driver can be split more than once, creating stackable drivers. This enables an independent software vendor to support system expansion, independent of both the hardware and the OS.

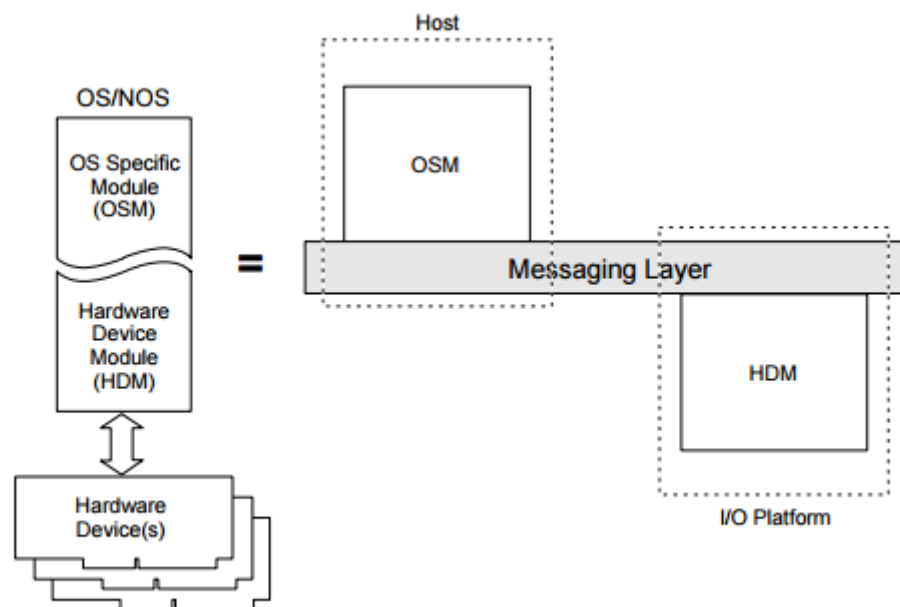


Figure 2: I2O Split device driver

Splitting the driver as shown in the figure 2 produces two modules:

**1. OS-specific module (OSM):**

It is the upper module provides the interface to the operating system. The OS vendor supplies this module, which contains no hardware-specific code.

**2. Hardware device module (HDM):**

It is the lower module provides the interface to the I/O adapter and its devices. The hardware vendor supplies this module, which contains no OS-specific code.

**Benefits:**

- Special processors used to complete I/O transactions reduces host CPU utilization.
- Handles real time requirements of I/O by exporting interrupts.
- Standardization on intelligent platforms for the benefit of all segments of the industry
  - The I2O specification enables the OS vendor to produce a single driver for each class of device and concentrate on optimizing the OS portion of the driver. Furthermore, the hardware vendor needs to produce only one version of that driver, which works for any OS that supports I2O. Thus ensuring standard product and thereby more focus can be laid to make it better and better
- Improved fault isolation and recovery, due to the physical and logical isolation of the I/O subsystem.