



THE UNIVERSITY of EDINBURGH
informatics

Operating Systems (INFR10079) 2023/2024 Semester 2

IO Subsystems

abarbala@inf.ed.ac.uk

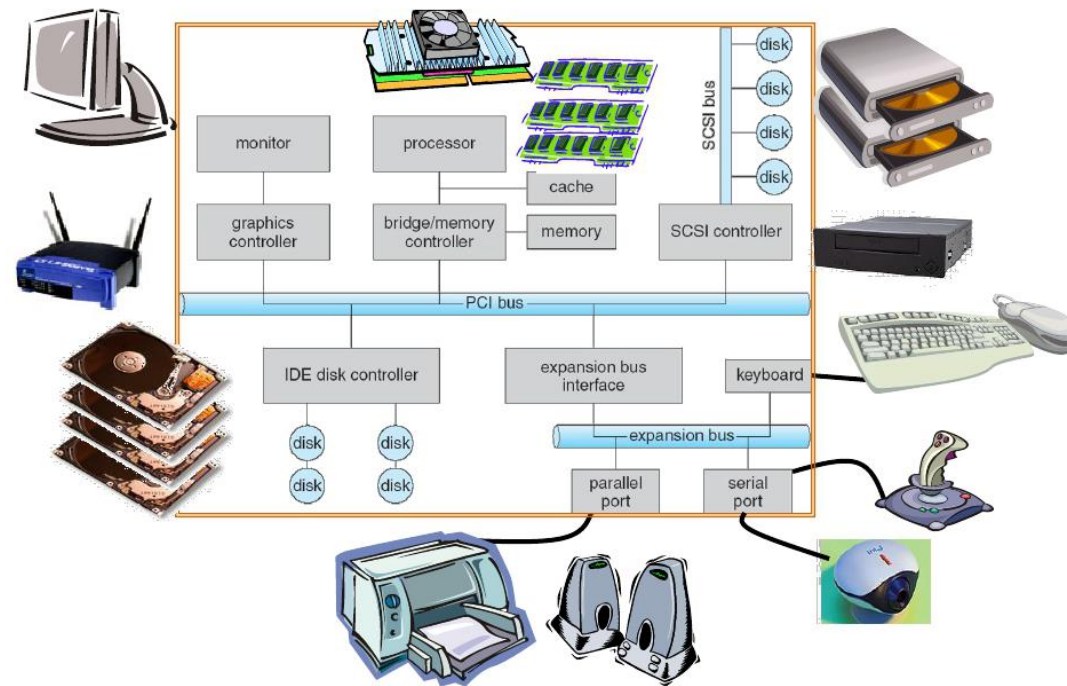
Chapter 12.1, 12.2, 12.3, 12.4, 12.5

IO Subsystems: Overview

- Applications do **IO and compute**
 - OS manages and controls IO for applications
 - Common interfaces to IO devices
 - IO Services
- IO Hardware
- CPU to device communication
 - PIO (Programmed input–output)
 - DMA (Direct memory access)
- Device Drivers
- IO Subsystem
- An IO Syscall Example

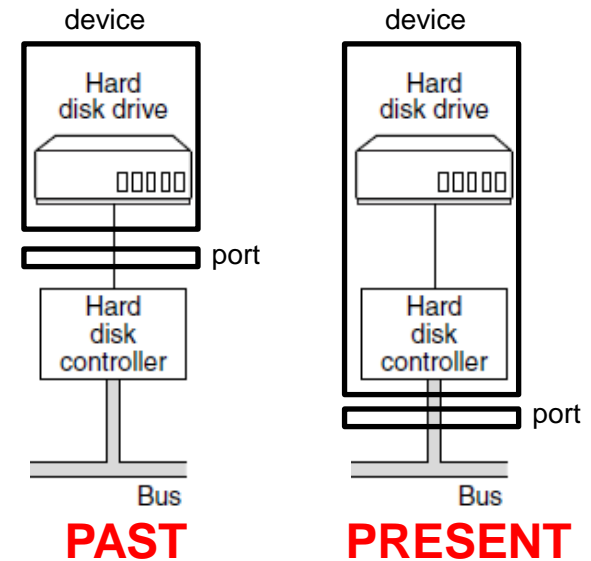
Devices

- Storage devices
 - Disk
 - Tape
- Transmission devices
 - Network connections
 - Bluetooth
- Human-interface devices
 - Screen
 - Keyboard
 - Mouse
 - Audio in
 - Audio out
- Specialized devices
 - E.g., to control a machine/equipment (aircraft)



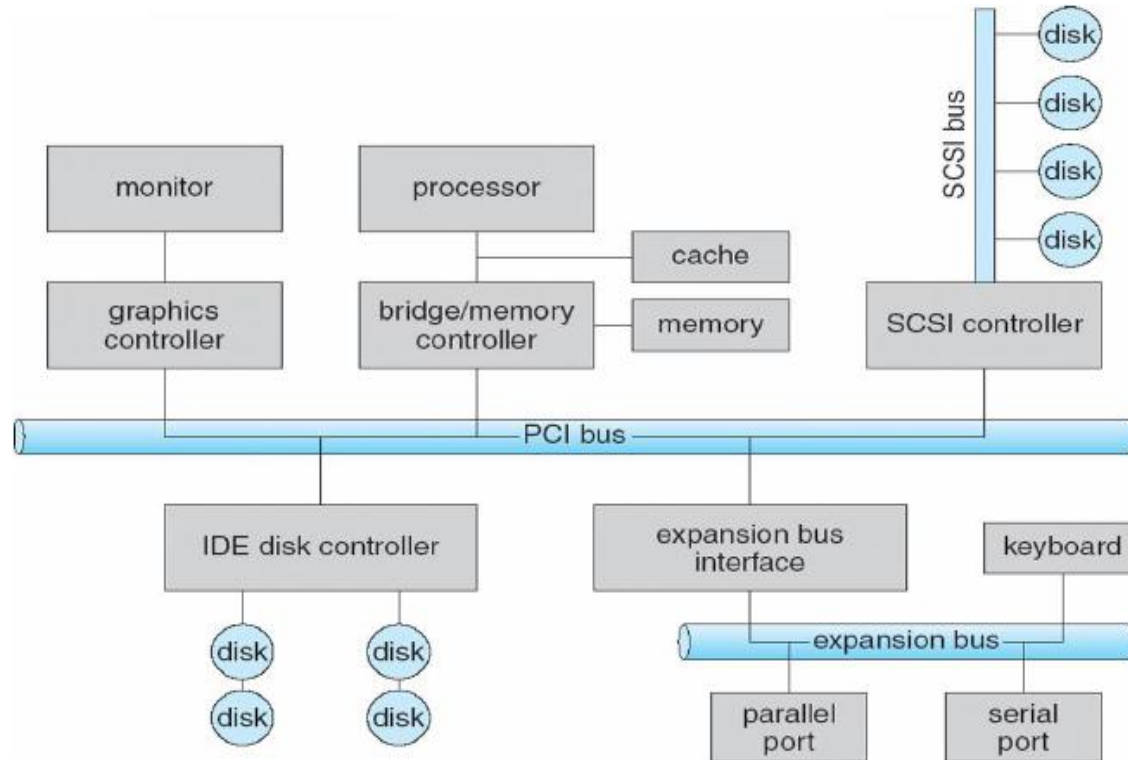
I/O Hardware #1

- Variety of I/O devices
- Common concepts
 - **Port**
 - Connection point for device (e.g., USB, parallel, serial, Ethernet)
 - **Bus**
 - **Peripheral** buses (e.g., PCI/PCIe)
 - **Expansion bus** connects relatively slow devices
 - **Device**
 - **Controller (host adapter)**
 - Electronics that operate port, bus, device
 - Sometimes integrated
 - Sometimes separate circuit board (host adapter)
 - Contains processor, microcode, private memory, bus controller, etc.



I/O Hardware #2

- Buses (cyan)
 - Handle the traffic between I/O devices and processor
- Examples
 - PCI/PCIe
 - Connects with high speed graphics, networking, etc.
 - Connects to low speed buses
 - SCSI
 - Used to be for fast devices with large bandwidth (disks, scanners, etc.)



CPU to Device Communication

- Controllers have
 - **Registers** for data and control
 - **Buffers** (memory-like areas) mostly for data
- CPU **communicates** with devices by reading and writing in registers and buffers
- Communication methods
 - **IO Ports**
 - **Memory-mapped IO**
 - **Hybrid**

I/O Ports

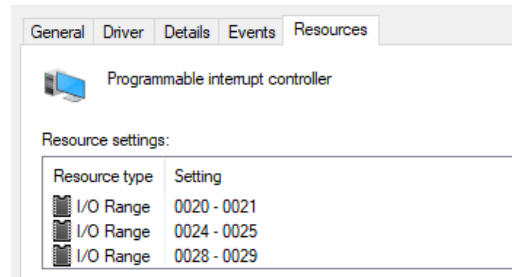
Memory Mapped I/O

Hybrid

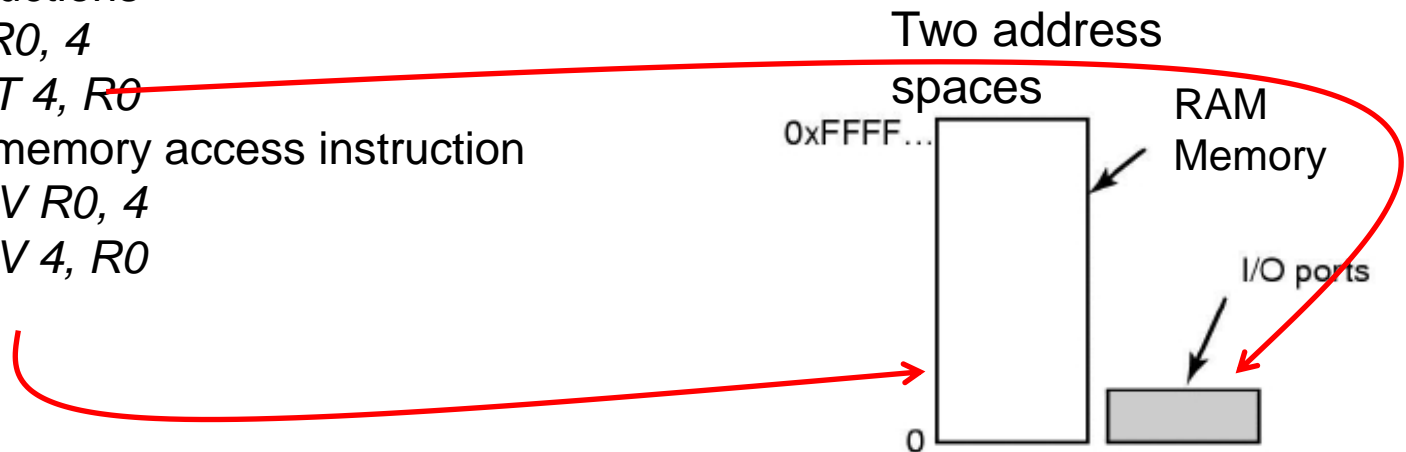
The image displays three screenshots of Windows Device Manager resource settings windows, illustrating different communication methods:

- Programmable interrupt controller Properties** (I/O Ports): The Resources tab shows three I/O Range settings: 0020 - 0021, 0024 - 0025, and 0028 - 0029.
- Trusted Platform Module 2.0 Properties** (Memory Mapped I/O): The Resources tab shows two Memory Range settings: 00000000FED40040 - 00000000FED4103F and 00000000FED40000 - 00000000FED40FFF.
- Intel(R) HD Graphics 520 Properties** (Hybrid): The Resources tab shows three settings: two Memory Range settings (00000000A0000000 - 00000000A0FFFFFF and 0000000090000000 - 000000009FFFFFFF) and one I/O Range setting (3000 - 303F).

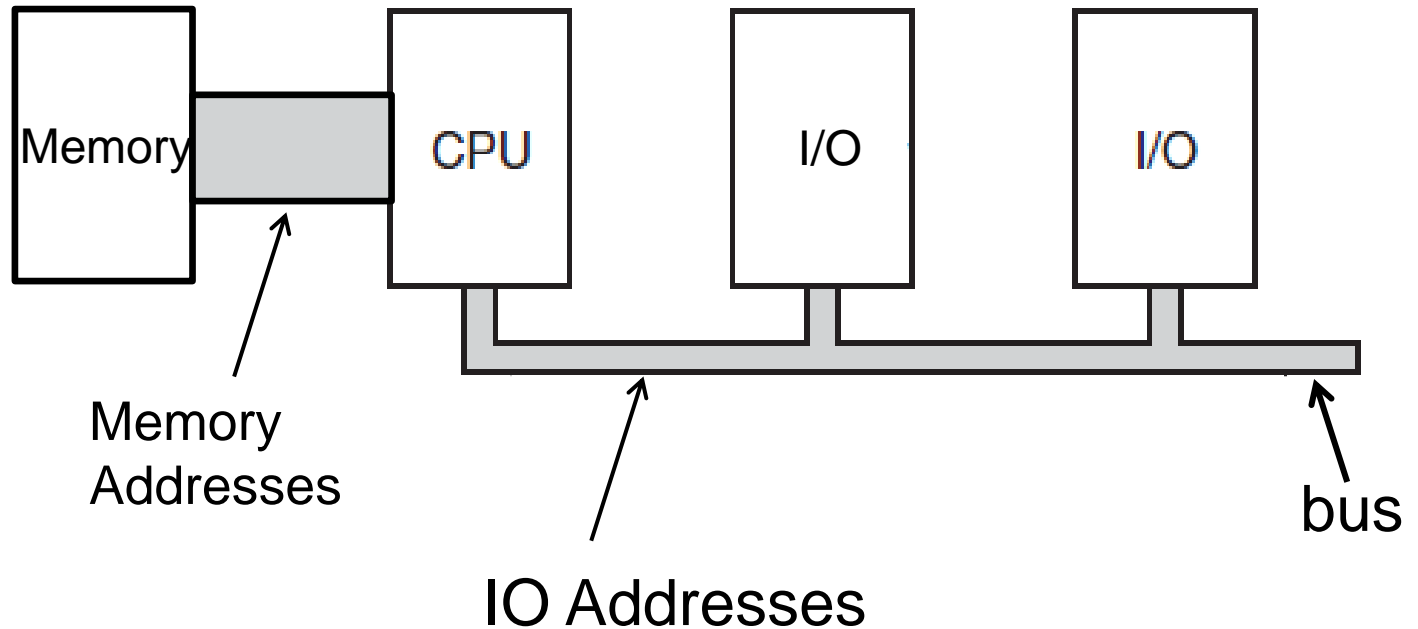
I/O Ports #1



- Each control register an I/O port number
- Special instructions to access the I/O port space
 - CPU reads in from device I/O PORT to CPU register
 - `IN REG, PORT`
 - CPU writes to device I/O PORT from CPU register
 - `OUT PORT, REG`
- Instruction are privileged (OS kernel only)
- Separate **I/O port space** and **memory space**
 - I/O instructions
 - `IN R0, 4`
 - `OUT 4, R0`
 - Similar memory access instruction
 - `MOV R0, 4`
 - `MOV 4, R0`



I/O Ports #2



I/O address range (hexadecimal)	device
000-00F	DMA controller
020-021	interrupt controller
040-043	timer
200-20F	game controller
2F8-2FF	serial port (secondary)
320-32F	hard-disk controller

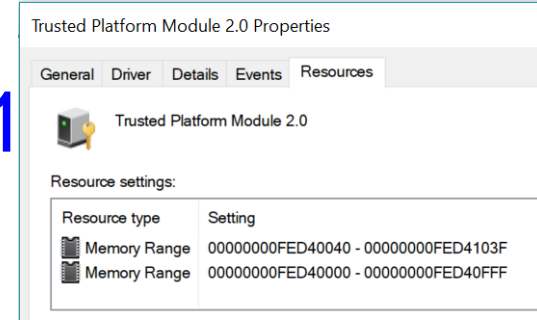
I/O Ports #3

You must be root!

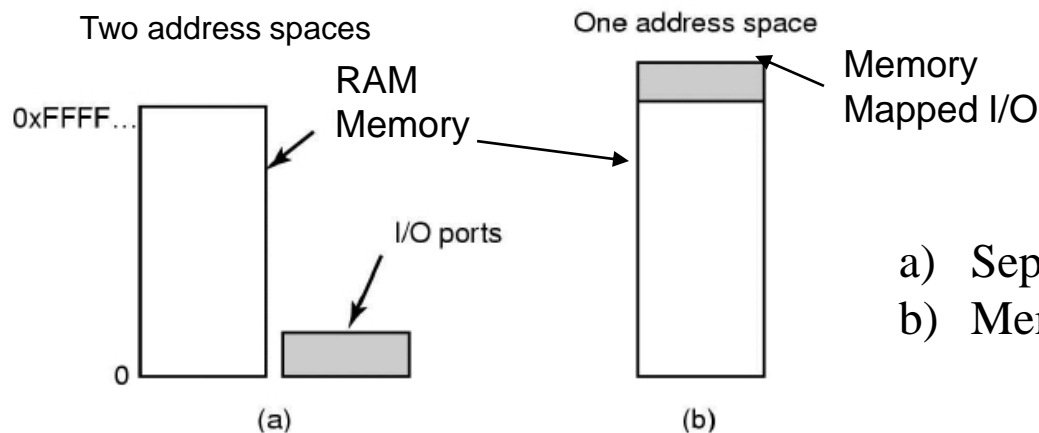
```
antonio@antonio-VirtualBox: ~  
File Edit View Search Terminal Help  
antonio@antonio-VirtualBox:~$ cat /proc/ioports  
0000-0000 : PCI Bus 0000:00  
  0000-0000 : dma1  
  0000-0000 : pic1  
  0000-0000 : timer0  
  0000-0000 : timer1  
  0000-0000 : keyboard  
  0000-0000 : keyboard  
  0000-0000 : rtc_cmos  
    0000-0000 : rtc0  
  0000-0000 : dma page reg  
  0000-0000 : pic2  
  0000-0000 : dma2  
  0000-0000 : fpu  
  0000-0000 : 0000:00:01.1  
    0000-0000 : ata_piix  
  0000-0000 : 0000:00:01.1  
    0000-0000 : ata_piix  
  0000-0000 : 0000:00:01.1  
    0000-0000 : ata_piix  
  0000-0000 : vga+  
  0000-0000 : 0000:00:01.1  
    0000-0000 : ata_piix  
0000-0000 : PCI conf1  
0000-0000 : PCI Bus 0000:00  
  0000-0000 : 0000:00:07.0  
    0000-0000 : ACPI PM1a_EVT_BLK  
    0000-0000 : ACPI PM1a_CNT_BLK  
    0000-0000 : ACPI PM_TMR  
    0000-0000 : ACPI GPE0_BLK  
  0000-0000 : 0000:00:07.0  
    0000-0000 : piix4_smbus  
  0000-0000 : 0000:00:01.1  
    0000-0000 : ata_piix  
  0000-0000 : 0000:00:03.0  
    0000-0000 : e1000  
  0000-0000 : 0000:00:04.0  
  0000-0000 : 0000:00:05.0  
    0000-0000 : Intel 82801AA-ICH  
  0000-0000 : 0000:00:05.0  
    0000-0000 : Intel 82801AA-ICH  
  0000-0000 : 0000:00:0d.0  
    0000-0000 : ahci  
  0000-0000 : 0000:00:0d.0  
    0000-0000 : ahci  
  0000-0000 : 0000:00:0d.0
```

```
root@antonio-VirtualBox: /home/antonio  
File Edit View Search Terminal Help  
root@antonio-VirtualBox:/home/antonio# cat /proc/ioports  
0000-0cf7 : PCI Bus 0000:00  
  0000-001f : dma1  
  0020-0021 : pic1  
  0040-0043 : timer0  
  0050-0053 : timer1  
  0060-0060 : keyboard  
  0064-0064 : keyboard  
  0070-0071 : rtc_cmos  
    0070-0071 : rtc0  
  0080-008f : dma page reg  
  00a0-00a1 : pic2  
  00c0-00df : dma2  
  00f0-00ff : fpu  
  0170-0177 : 0000:00:01.1  
    0170-0177 : ata_piix  
  01f0-01f7 : 0000:00:01.1  
    01f0-01f7 : ata_piix  
  0376-0376 : 0000:00:01.1  
    0376-0376 : ata_piix  
  03c0-03df : vga+  
  03f6-03f6 : 0000:00:01.1  
    03f6-03f6 : ata_piix  
0cf8-0cff : PCI conf1  
0d00-ffff : PCI Bus 0000:00  
  4000-403f : 0000:00:07.0  
    4000-4003 : ACPI PM1a_EVT_BLK  
    4004-4005 : ACPI PM1a_CNT_BLK  
    4008-400b : ACPI PM_TMR  
    4020-4021 : ACPI GPE0_BLK  
  4100-410f : 0000:00:07.0  
    4100-4108 : piix4_smbus  
  d000-d00f : 0000:00:01.1  
    d000-d00f : ata_piix  
  d010-d017 : 0000:00:03.0  
    d010-d017 : e1000  
  d020-d03f : 0000:00:04.0  
  d100-d1ff : 0000:00:05.0  
    d100-d1ff : Intel 82801AA-ICH  
  d200-d23f : 0000:00:05.0  
    d200-d23f : Intel 82801AA-ICH  
  d240-d247 : 0000:00:0d.0  
    d240-d247 : ahci  
  d248-d24b : 0000:00:0d.0  
    d248-d24b : ahci  
  d250-d257 : 0000:00:0d.0
```

Memory-mapped I/O #1

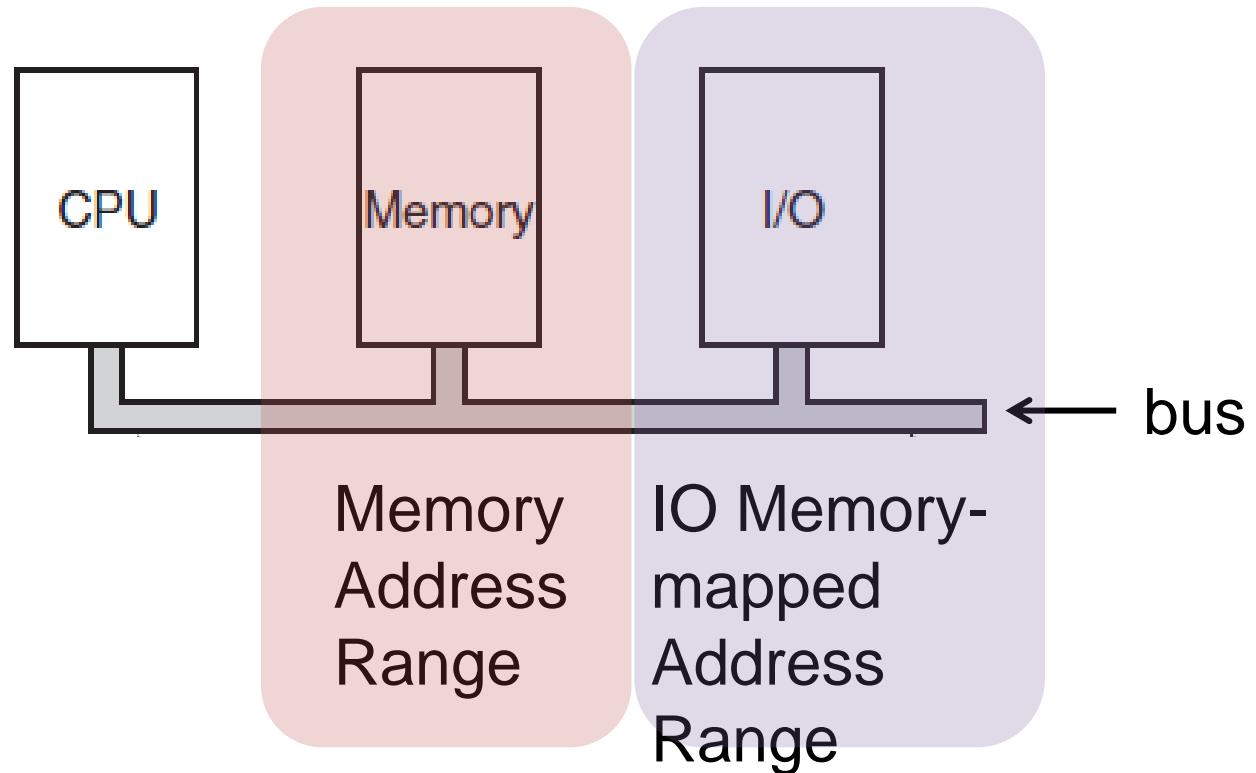


- All control registers and buffers into the memory space
- Each control register is assigned a unique memory address
 - There is no actual RAM memory for this address
- Such addresses may be at the top of the physical address space



- a) Separate I/O and memory space
- b) Memory-mapped I/O

Memory-mapped I/O #2



Example

0 ... 32GB

127.999TB ... 128TB

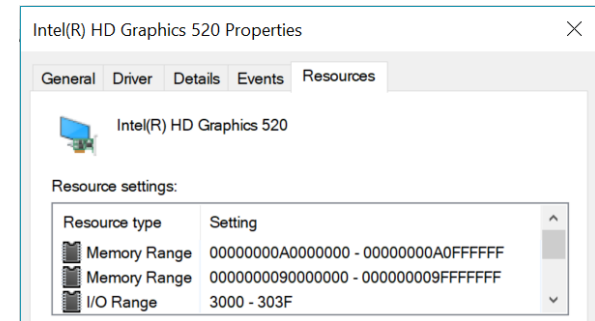
Memory-mapped I/O #3

```
antonio@antonio-VirtualBox: ~  
File Edit View Search Terminal Help  
antonio@antonio-VirtualBox:~$ cat /proc/iomem  
00000000-00000000 : Reserved  
00000000-00000000 : System RAM  
00000000-00000000 : Reserved  
00000000-00000000 : PCI Bus 0000:00  
00000000-00000000 : Video ROM  
00000000-00000000 : Adapter ROM  
00000000-00000000 : Reserved  
  00000000-00000000 : System ROM  
00000000-00000000 : System RAM  
  00000000-00000000 : Kernel code  
  00000000-00000000 : Kernel data  
  00000000-00000000 : Kernel bss  
00000000-00000000 : ACPI Tables  
00000000-00000000 : PCI Bus 0000:00  
  00000000-00000000 : 0000:00:02.0  
  00000000-00000000 : 0000:00:03.0  
  00000000-00000000 : e1000  
  00000000-00000000 : 0000:00:04.0  
  00000000-00000000 : vboxguest  
  00000000-00000000 : 0000:00:04.0  
  00000000-00000000 : 0000:00:06.0  
  00000000-00000000 : ohci_hcd  
  00000000-00000000 : 0000:00:0d.0  
  00000000-00000000 : ahci  
00000000-00000000 : Reserved  
  00000000-00000000 : IOAPIC 0  
00000000-00000000 : Local APIC  
  00000000-00000000 : Reserved  
00000000-00000000 : Reserved  
00000000-00000000 : System RAM  
antonio@antonio-VirtualBox:~$
```

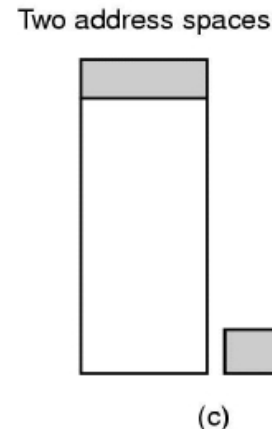
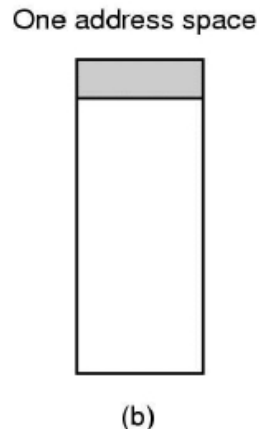
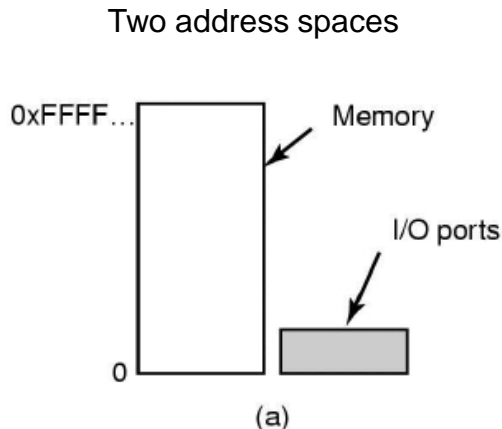
```
root@antonio-VirtualBox: /home/antonio  
File Edit View Search Terminal Help  
root@antonio-VirtualBox:/home/antonio# cat /proc/iomem  
00000000-00000fff : Reserved  
00001000-0009fbff : System RAM  
0009fc00-0009ffff : Reserved  
000a0000-000bffff : PCI Bus 0000:00  
000c0000-000c7fff : Video ROM  
000e2000-000ef3ff : Adapter ROM  
000f0000-000fffff : Reserved  
  000f0000-000fffff : System ROM  
00100000-dfffffff : System RAM  
  20a00000-216031d0 : Kernel code  
  216031d1-2206a43f : Kernel data  
  222e2000-2253dfff : Kernel bss  
dfff0000-dfffffff : ACPI Tables  
e0000000-fdffffff : PCI Bus 0000:00  
  e0000000-e1ffffff : 0000:00:02.0  
  f0000000-f001ffff : 0000:00:03.0  
  f0000000-f001ffff : e1000  
  f0400000-f07fffff : 0000:00:04.0  
  f0400000-f07fffff : vboxguest  
  f0800000-f0803fff : 0000:00:04.0  
  f0804000-f0804fff : 0000:00:06.0  
  f0804000-f0804fff : ohci_hcd  
  f0806000-f0807fff : 0000:00:0d.0  
  f0806000-f0807fff : ahci  
fec00000-fec00fff : Reserved  
  fec00000-fec003ff : IOAPIC 0  
fee00000-fee00fff : Local APIC  
  fee00000-fee00fff : Reserved  
fffc0000-ffffffff : Reserved  
10000000-11ffffff : System RAM  
root@antonio-VirtualBox:/home/antonio#
```

00001000-0009fbff : System RAM
00100000-dfffffff : System RAM
10000000-11ffffff : System RAM

Hybrid #1

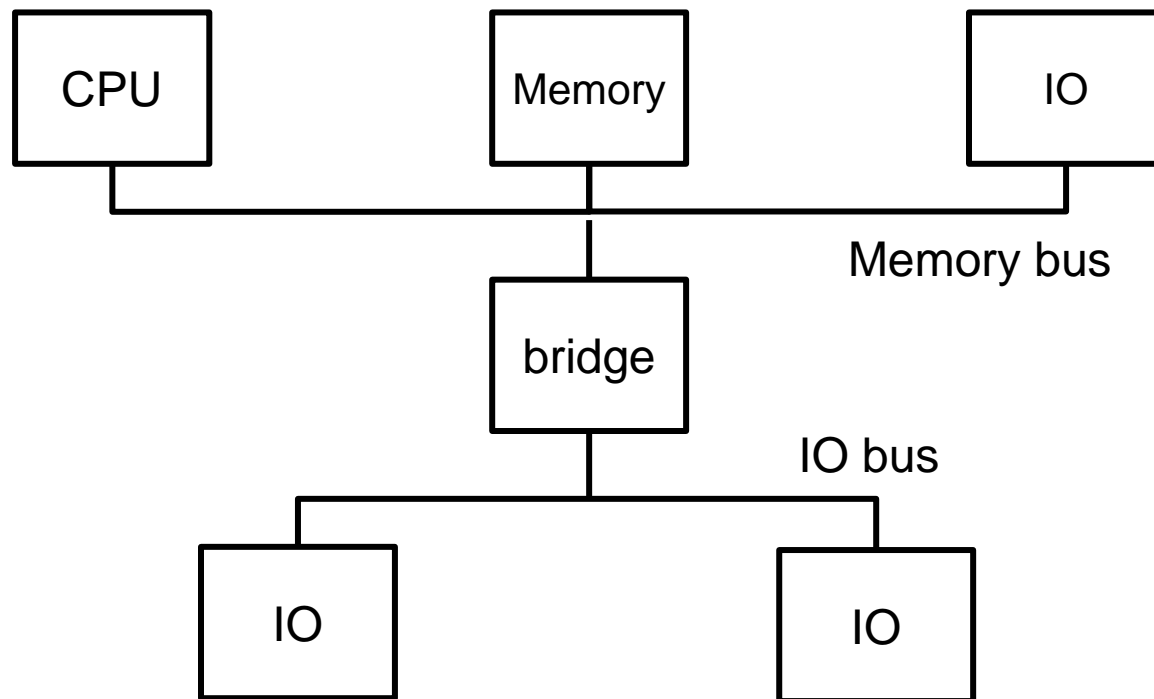


- I/O ports and memory-mapped IO
- Example
 - Memory-mapped I/O data buffers and separate I/O ports for the control registers
 - x86 CPUs, memory addresses 640K to 1M – 1 being reserved for device data buffers, in addition to I/O ports 0 to 64K – 1



- a) Separate I/O and memory space
- b) Memory-mapped I/O
- c) Hybrid

Hybrid #2



Offloaded Communication

- The CPU can request data from an I/O controller **one byte at a time**
 - Programmed IO (PIO)
 - (Previous slides)
 - This wastes CPU's time for large data transfers
 - Small data transfers are OK
- CPU offloads data transfers
- **DMA (Direct Memory Access) controller** transfers data for the CPU
 - From/to an IO Device
 - Between IO Devices

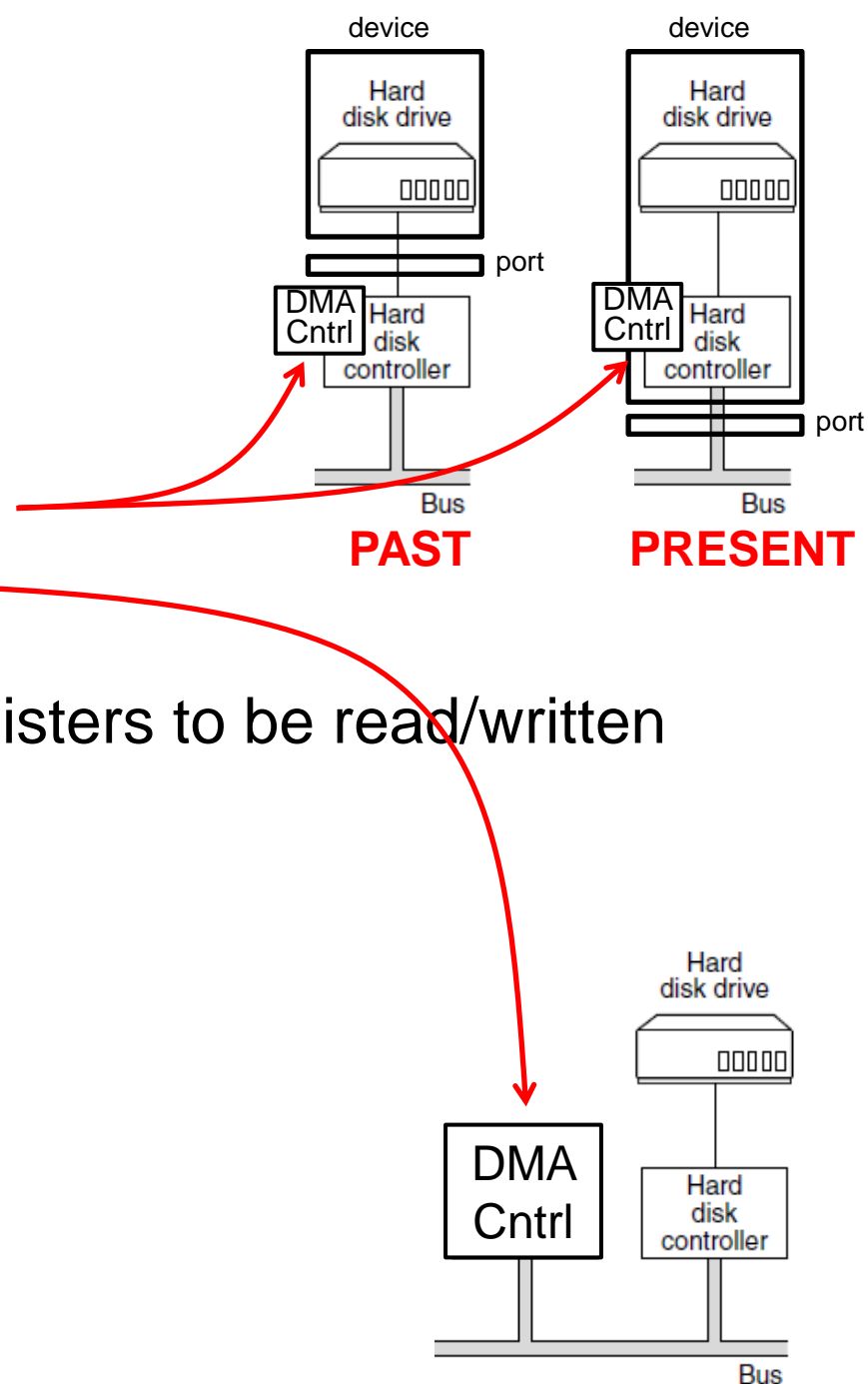
Direct Memory Access

- Requires a **DMA controller**

- On the device host controller
- On the motherboard

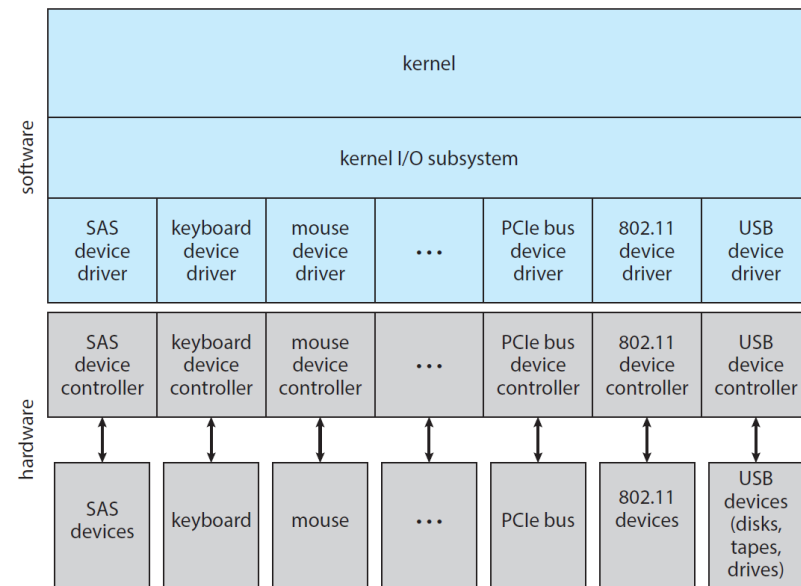
- **DMA controller** contains registers to be read/written by the software

- Memory address register
- Byte count register
- Control registers to
 - Direction of the transfer
 - Transfer unit
 - Byte burst size
 - ...



OS Device Drivers

- Great variety of devices
 - Each device vendor/model its specs
- OS deals with IO devices in a **standard and uniform way**
 - Abstraction
 - Encapsulation
 - Software layering
- Use specific interface (file)
- Encapsulate the differences in devices by **device drivers' classes**
 - Each OS its own standard
- Example
 - An application can open a file without knowing what kind of disk it is
 - Independently of the disk technology



Characterizing IO Devices

aspect	variation	example
data-transfer mode	character block	terminal disk
access method	sequential random	modem CD-ROM
transfer schedule	synchronous asynchronous	tape keyboard
sharing	dedicated sharable	tape keyboard
device speed	latency seek time transfer rate delay between operations	
I/O direction	read only write only read–write	CD-ROM graphics controller disk

IO Services Provided by the OS

- Kernel **IO subsystem** services
 - Available to applications and to other parts of the OS
- Management of the name space for files and devices
- Access control to files and devices
- Operation control (for example, a modem cannot seek())
- File-system space allocation
- Device allocation
- Buffering, caching, and spooling
- I/O scheduling
- Device-status monitoring, error handling, and failure recovery
- Device-driver configuration and initialization
- Power management of I/O devices

Putting Everything Together: Life Cycle of an IO Request

