Computer-System Architecture

- I/O devices and the CPU can execute concurrently.
- Each device controller is in charge of a particular device type.
- Each device controller has a local buffer.
- I/O is from the device to local buffer of controller.
- Device controller informs CPU that it has finished its operation by causing an interrupt.

Common Functions of Interrupts

- Interrupt transfers control to the interrupt service routine generally, through the interrupt vector, which contains the addresses of all the service routines.
- Interrupt architecture must save the address of the interrupted instruction.
- Incoming interrupts are disabled while another interrupt is being processed to prevent a lost interrupt.
- A trap is a software-generated interrupt caused either by an error or a user request.
- An operating system is interrupt driven.

Interrupt Handling

- The operating system preserves the state of the CPU by storing registers and the program counter.
- Separate segments of code determine what action should be taken for each type of interrupt.
Device-Status Table

<table>
<thead>
<tr>
<th>Device</th>
<th>Status</th>
<th>Request</th>
</tr>
</thead>
<tbody>
<tr>
<td>card reader 1</td>
<td>idle</td>
<td>request for line printer</td>
</tr>
<tr>
<td></td>
<td></td>
<td>address: 35546 length: 1372</td>
</tr>
<tr>
<td>line printer 3</td>
<td>busy</td>
<td>request for disk unit 3</td>
</tr>
<tr>
<td></td>
<td></td>
<td>file: xxx operation: read</td>
</tr>
<tr>
<td></td>
<td></td>
<td>address: 43946 length: 20000</td>
</tr>
<tr>
<td>disk unit 1</td>
<td>idle</td>
<td>request for disk unit 3</td>
</tr>
<tr>
<td></td>
<td></td>
<td>file: yyy operation: write</td>
</tr>
<tr>
<td></td>
<td></td>
<td>address: 63458 length: 500</td>
</tr>
<tr>
<td>disk unit 2</td>
<td>idle</td>
<td></td>
</tr>
<tr>
<td>disk unit 3</td>
<td>busy</td>
<td></td>
</tr>
</tbody>
</table>

Direct Memory Access (DMA) Structure

- Used for high-speed I/O devices able to transmit information at close to memory speeds.
- Device controller transfers blocks of data from buffer storage directly to main memory without CPU intervention.
- Only one interrupt is generated per block, rather than the one interrupt per byte.

Storage Structure

- Main memory – only large storage media that the CPU can access directly.
- Secondary storage – extension of main memory that provides large nonvolatile storage capacity.
- Magnetic disks – rigid metal or glass platters covered with magnetic recording material
  - Disk surface is logically divided into tracks, which are subdivided into sectors.
  - The disk controller determines the logical interaction between the device and the computer.

Moving-Head Disk Mechanism
Storage Hierarchy

- Storage systems organized in hierarchy.
  - Speed
  - Cost
  - Volatility

- Caching – copying information into faster storage system; main memory can be viewed as a cache for secondary storage.

Migration of A From Disk to Register

- Use of high-speed memory to hold recently-accessed data.
- Requires a cache management policy. Caching introduces another level in storage hierarchy. This requires data that is simultaneously stored in more than one level to be consistent.

Dual-Mode Operation

- Sharing system resources requires operating system to ensure that an incorrect program cannot cause other programs to execute incorrectly.
- Provides hardware support to differentiate between at least two modes of operations.
  1. User mode – execution done on behalf of a user.
  2. Kernel mode (also monitor mode, supervisor mode or system mode) – execution done on behalf of operating system.

Dual-Mode Operation (Cont.)

- Mode bit added to computer hardware to indicate the current mode: kernel (0) or user (1).
- When an interrupt or fault occurs hardware switches to kernel mode.

- Privileged instructions can be issued only in kernel mode.
I/O Protection

- All I/O instructions are privileged instructions.
- Given the I/O instructions are privileged, how does the user program perform I/O?
- System call – the method used by a process to request action by the operating system.
  - Usually takes the form of a trap to a specific location in the interrupt vector.
  - Control passes through the interrupt vector to a service routine in the OS, and the mode bit is set to kernel mode.
  - The kernel verifies that the parameters are correct and legal, executes the request, and returns control to the instruction following the system call.

Memory Protection

- Must provide memory protection at least for the interrupt vector and the interrupt service routines.
- In order to have memory protection, add two registers that determine the range of legal addresses a program may access:
  - base register – holds the smallest legal physical memory address.
  - Limit register – contains the size of the range
- Memory outside the defined range is protected.

Use of a System Call to Perform an I/O

- Must ensure that a user program could never gain control of the computer in kernel mode. Otherwise, undesirable actions can be done e.g. a user program that, as part of its execution, stores a new address in the interrupt vector.

A Base And A Limit Register Define A Logical Address Space
**Protection Hardware**

- When executing in kernel mode, the operating system has unrestricted access to both kernel and user’s memory.
- The load instructions for the `base` and `limit` registers are privileged instructions.

**CPU Protection**

- **Timer** – interrupts computer after specified period to ensure operating system maintains control.
  - Timer is decremented every clock tick.
  - When timer reaches the value 0, an interrupt occurs.
- Timer commonly used to implement time sharing.
- Time also used to compute the current time.
- Load-timer is a privileged instruction.