Name : Andrias Sigid W.R
Class : X
NIM : L200144010
Assignment : Operating Systems (Resume)

Input Output

Direct Memory Access (DMA)

is to Used to avoid being programmed I / O for large data movement and to erform CPU bypass for data transfer directly between I / O devices and memory and requires DMA controller.

Interface Aplikasi I/O

1. System call I / O encapsulates devices in more general classes
2. Device-driver layer hides the difference between the I / O controllers of the kernel
3. Devices have various dimensions :
    a. Character-stream or block
    b. Sequential or random-access
    c. Sharable or dedicated
    d. Speed atau operation
    e. read-write, read only, or just write

Block and Character Device

1. Block devices included into it are disks
   Drives
   The commands include read, write, seek
    a. Raw I/O or file-system access
    b. Allow Memory-mapped file access
2. Character devices include keyboards, mice, serial ports
    a. The commands include get, put
    b. The layered libraries are located at the top of the editing line

Network Device

1. Varied from block and character used for interface
2. Unix and Windows NT/2000 input the socket interface
    a. Separate network protocol from network operation
    b. Enter the select function
    c. The approach is quite varied (pipes, FIFOs, streams, queues, mailboxes)

Clock and Timer

1. Provide current time, elapsed time, timer
2. If the programmable interval timer is used for timing, the interrupt is performed periodically
3. ioctl (on UNIX) mask the unusual aspects of I / O like clock and timer

Blocking and Nonblocking I/O

1. Blocking – process is suspended until I/O is complete
    1. Easy to use and understand
    2. Sometimes not suitable for all needs
2. Nonblocking - I/O returned
    1. User interface, data copy (buffered I/O)
    2. Implemented through multi-threading
    3. The return is done quickly by counting the bytes read or written
3. Asynchronous – process executed during I/O execution
    1. Difficult to use
    2. Signal I/O subsystem performs the process when I/O completes

Subsystem Kernel I/O

1. Scheduling
    1. Some I/O requests are ordered through the device sequence
    2. For some Operating Systems quite reasonable
2. Buffering – store data in memory when transfer between devices
    1. To overcome the speed is not worth it
    2. To overcome the size of the transfer is not worth it
    3. For the management of "copy semantics
3. Caching – fast memory handle data copy
    1. Just for copy
    2. Key in performance
4. Spooling - handle output on the device
   device can serve itself one request at a time, for example, printing (printing)
5. Device reservation – provides exclusive access to the device
    1. System calls for allocation and deallocation
    2. Be careful in deadlock

Error Handling

1. The operating system can restore disk readings, unavailable devices and write failures due to transients
2. The return of an error number or code occurs when the request for I / O fails.
3. System error log gives an error report

Data Kernel Structure

1. The kernel maintains the status information of the I/O components including open file tables, network connections, device character status
2. Many complex data structures for track buffers, memory allocations, and dirty blocks.
3. The use of object-oriented methods and message passing methods for I / O implementation

I/O Request for Hardware Operation

The process of reading the file from disk:

1. Specifies the device that handles the file
2. Translate a name to a representative device
3. Physical reading of data from disk via buffer
4. Make the data available so it can be requested for the process
5. Returns control to the process.