

C502 – Operating Systems Tutorial *

Device Management – Solutions

Note: The solution notes below only briefly list (some of) the key points that should be included in an answer. They are by no means complete. In an exam, you are expected to spell out the solution more fully and include a detailed explanation of your reasoning.

1. In which of the four I/O software layers (user-level I/O software, device-independent OS software, device drivers and interrupt handlers) is each of the following done?
 - (a) Computing the track, sector and head for a disk read
 - (b) Maintaining a cache of recently used blocks
 - (c) Writing commands to the drive registers
 - (d) Checking to see if the user is permitted to use the device
 - (e) Converting binary integers to ASCII for printing

Answer:

- (a) *Device driver (or hardware): This requires familiarity with the actual disk layout used, so only the hard disk driver will have the necessary information. For modern hard disks, the mapping will often be done entirely in hardware by the disk controller because only the built-in disk hardware may be familiar with the location of bad blocks etc.*
 - (b) *Device-independent OS layer: A block cache is useful across a wide-range of block I/O devices. By putting this functionality into the device-independent layer, it can be reused/shared by multiple different devices.*
 - (c) *Interrupt handler (or device driver): If this is a quick but time-critical operation, then it can be performed within the interrupt handling routine. If updating the drive registers requires more time (or is device-specific), then it would be done by the device driver.*
 - (d) *Device-independent OS layer: Performing access control checks for devices is functionality that is applicable to many devices. Therefore, it makes sense to factor it out and provide it in the device-independent layer. The OS can then enforce access control policy uniformly across all devices.*
 - (e) *User-level I/O layer: Typically conversion between data representations for I/O will be done by a user-level I/O library that an application can linked against. This gives the user maximum flexibility in choosing the right conversion strategy. In addition, the conversion can be done cheaply in user space because it does not need any privileged kernel access.*
2. Explain what direct memory access (DMA) is and why it is used. Although DMA does not use the CPU, the maximum transfer rate is still limited. Consider reading a block from disk. Name **three** factors that might ultimately limit the rate of transfer. (Exam question 2015-16).

Answer: *Direct Memory Access (DMA) is a technique to relieve the CPU from the task of handling low-level device I/O. A special hardware device called a DMA controller takes over the responsibility of handling I/O. Simple DMA controllers can only handle a single read at a time, whereas more sophisticated ones support multiple reads and putting the results at different memory locations (scatter-gather DMA).*

The transfer rate of blocks from a disk to memory using DMA will be limited by:

- 1 – Transfer rate supported by the hard disk, i.e. how fast the disk can read blocks and put them on the bus.*

*with thanks to Morris Sloman

2 – Transfer rate supported by the bus and potential contention for bus access.

3 – Access bandwidth supported by the memory and, again, potentially contention with the CPU for memory access.

3. What is spooling? Why is a printer spooling system better than direct user access to printers?

Answer: *Spooling is the allocation of all interaction with a given device to a single process; other processes that want access to the device must go through the spool process. A non-spoiled printer is vulnerable to two or more people printing on it at once: in such cases the result will be gibberish. A spoiled printer does not suffer from the problem, since it has better means of communication with client processes (e.g. depositing and noticing a file in the spool directory). Spooling also frees processes from having to supervise the printing process; the `lpr` command can return without waiting for the file to be printed. This is much better than the usual situation on non-spoiled single-tasking systems.*

4. An operating system has to support I/O devices with very diverse properties. Complete the following table, as exemplified below, using your best guesses.

Device	Type (Character/Block)	Operation (Read, Write, Seek)
Clock		
Keyboard		
Mouse		
56k Modem	C	R, W
ISDN line		
Laser Printer		
Scanner		
52x CD-ROM		
FastEthernet		
EIDE (ATA-2)disk		
ISA bus		
Fire Wire (IEEE 1394)		
USB 2.0		
XGA Monitor		
Gigabit Ethernet		
Serial ATA disk		
SCSI Ultrawide4 disk		
PCI bus		

Answer:

<i>Device</i>	<i>Type (Character/Block)</i>	<i>Operation (Read, Write, Seek)</i>
<i>Clock</i>	<i>?</i>	<i>?</i>
<i>Keyboard</i>	<i>C</i>	<i>R</i>
<i>Mouse</i>	<i>C</i>	<i>R</i>
<i>56k Modem</i>	<i>C</i>	<i>R, W</i>
<i>ISDN line</i>	<i>C</i>	<i>R, W</i>
<i>Laser Printer</i>	<i>C</i>	<i>W</i>
<i>Scanner</i>	<i>C</i>	<i>R</i>
<i>52x CD-ROM</i>	<i>B</i>	<i>R, S</i>
<i>FastEthernet</i>	<i>C</i>	<i>R, W</i>
<i>EIDE (ATA-2) disk</i>	<i>B</i>	<i>R, W, S</i>
<i>ISA bus</i>	<i>C</i>	<i>R, W</i>
<i>Fire Wire (IEEE 1394)</i>	<i>C</i>	<i>R, W</i>
<i>USB 2.0</i>	<i>C</i>	<i>R, W</i>
<i>XGA Monitor</i>	<i>C</i>	<i>R, W</i>
<i>Gigabit Ethernet</i>	<i>C</i>	<i>R, W</i>
<i>Serial ATA disk</i>	<i>B</i>	<i>R, W, S</i>
<i>SCSI Ultrawide4 disk</i>	<i>B</i>	<i>R, W, S</i>
<i>PCI bus</i>	<i>C</i>	<i>R, W</i>

5. Write a C program that implements the copy (cp) command. Your program should be invoked as:

mycp <source file> <destination file>

- Make sure that you use the correct Linux I/O calls. How efficient is your implementation compared to the standard cp command? You can use the **time** command to measure execution times for various file sizes. If there is a performance difference, can you explain it?
- The **strace** command can be used to trace the system calls that a program makes. Compare the system calls between cp and mycp. Again, can you explain the differences?

Answer: *This is left as an exercise.*

- Any performance difference can probably be explained by the larger block size (32KB) that **cp** uses by default.
- The standard **cp** command probably does additional calls to **stat64** and **fstat64** to check for the existence of the source and destination files and ensure their successful access.