211 — Operating Systems – Tutorial
Device Management

Peter Pietzuch <prp@doc.ic.ac.uk>

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.
2. What is the difference between
   (a) a *device driver* and a *device controller*?
   (b) a *block-oriented* device and a *character-oriented* device?
3. What is *memory-mapped IO*? Why is it sometimes used?
4. An alternative to using interrupts for I/O is *polling*. Are there any circumstances when using polling is a better choice?
5. 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.
6. What is *spooling*?

   Why is a printer spooling system better than direct user access to printers?
7. 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	**Data rate**	**Type** (**C**haracter/**B**lock)	**Operation** (**R**ead, **W**rite, **S**eek)
Clock			
Keyboard			
Mouse			
56k Modem	*7 KB/sec*	*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			

8. Explain how one can provide an *asynchronous* I/O API on top of a *blocking* I/O system call interface.

   You have to implement a web server that should handle thousands of concurrent incoming connections. What would be the advantages of using a non-blocking I/O interface for this?

9. Write a C program that implements the *copy* (`cp`) command. Your program should be invoked as

   `mycp <source file> <destination file>`.

   (a) Write your program on a sheet of paper. Make sure that you use the correct Linux I/O calls.

   (b) Now try running your program on a computer. 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?

   (c) 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?