Frequently Asked Computer Systems Questions On the Scholar Webboard

What are standard file formats and why do we need them?

Standard file formats are agreed formats for storing and transmitting information. They make it possible for information created in one application to be read by a different application.

Different types of information will have different standard file formats. Eg. Text in its simplest format is stored as **ASCII** code. ASCII is a file format which is almost universally readable by any text based application. It does not contain any other information about how the text is laid out however. If you want to store information about fonts, size, alignment, margins, tables etc, then you need a more complicated standard file format like **Rich Text Format(RTF)** Most word processors will have the option to store files as RTF in addition to their own proprietary format. This means that you can create a document using one word processor and be able to import and edit it in a different word processor. **Hypertext Markup Language (HTML)** is another standard file format popular for text as it allows web browsers to share information.

There are standard file formats for almost every type of information. Some formats were originally designed by software developers for a particular application which was able to achieve sufficient market dominance for its file format to become the standard. Other standard file formats have come about as a result of an agreement between organisations such as the International Standards Organisation (ISO)

Application	**Standard File Format**
Web Browser	Hypertext Markup Language (HTML)
Spreadsheet	Symbolic Link Format (SYLK)
Database	Comma Separated Values (CSV)
Word Processor	Rich Text Format (RTF)
Bitmapped Graphics	Graphical Interchange Format (GIF) Joint Photographic Expert Group (JPEG)
Vector Graphics	Scalable Vector Graphics (SVG)
Movie	Motion Picture Expert Group (MPEG)

What is a buffer and what function does a buffer perform when information is being transferred between an interface and a peripheral?

A buffer is dedicated RAM which is used to store information in transit between the CPU and a peripheral or vice versa. Buffers vary in capacity, and may be part of the interface, or installed in the peripheral.

In the case of a keyboard, the buffer only needs to store a few characters at a time while the interface is waiting for the processor to give it the attention it needs to transfer the key-presses in the time since the processor last checked the buffer. In the case of a printer, a buffer might be large enough to store the images of several pages of text.

Where printers are concerned, buffers can make the transfer of information from CPU to peripheral much more efficient because a large block of information can be stored in the printer buffer in one go. The printer then uses its buffer to retrieve the information to be printed. This means that the processor can get on with other jobs until the printer has finished dealing with the information in the buffer and requests another transfer. If there was no buffer in the printer, information would have to be transmitted between CPU and printer one byte at a time, taking up a lot more processor time.

What is the difference between a peer-to-peer and a client-server network?

A peer-to-peer network is one where the machines connected together all have roughly equal status. They may share resources such as a printer, a hard disk or an Internet connection, but these resources are probably not all concentrated on one machine. A peer-to-peer network is usually a small network of up to 5 machines and is likely to be installed in a place where there is no need for internal security, such as a family home or a small office. Peer-to-peer networks can be set up using most modern operating systems.

A client-server network is one where a large number of machines are connected together and where there is a need for dedicated servers which are not used as workstations, but which deal with sharing network resources, network services such as email, an intranet and network security. A client-server network may have several hundred machines connected together, and may have one or more network managers who deal with security, backup, and application installation. The servers on a client-server network will be running a network operating system which is different from the workstations connected to them.

Peer-to peer:

Advantages	Disadvantages
Cheap to install	Poor security
Resource sharing	No dedicated server so performance may decrease when a machine's resource is accessed
No need to employ a network manager	No central backup
	No central filing system

Client-server:

Advantages	Disadvantages
Dedicated servers	Expensive to install
Good Security	Need to employ a network manager
Centralised backup and services	Everything depends on the servers

What is the purpose of a cache and how does it improve processor performance?

A cache is an area of fast memory which acts as an intermediary between the processor and the main memory. Although data can move around the processor very quickly, the system bus which connects the main memory (RAM) to the processor is becoming a bottleneck, and time is wasted waiting for data to be read from or written to the main memory.

Cache memory is usually Static Random Access Memory (SRAM) which is faster than Dynamic Random Access Memory (DRAM) and although the cache is smaller, data can be read from or written to it more quickly.

When data or instructions are read from memory, the memory locations following it are copied into the cache. This means that the next instruction can then be read from the cache instead of from the main memory, thus saving time. If the data required is not in the cache then the main memory is accessed, and although in this case two memory reads takes up a little more time, the overall saving is still significant.

When data or instructions are written back, the cache may be written to and the main memory updated later, (write back cache) or both may be written to at the same time (write through cache) Write back cache is about 10% faster than write through cache.

How do you calculate the storage capacity needed for a graphics file?

To calculate the storage requirements for a graphics file you need to know:

The number of pixels needed, and the colour depth of the graphic.

The number of pixels can be calculated from the total physical size of the graphic – usually given in square inches, and the number of pixels or dots per inch (DPI)

300DPI = 300 X 300 pixels per square inch
600DPI = 600 X 600 pixels per square inch

The colour depth can be calculated from the total number of colours used. This will tell you how many bits are needed to store that number of colours.

Colours used = 2 ^ bits needed

2 colours = 1 bit
4 colours = 2 bits
8 colours = 3 bits
16 colours = 4 bits

256 colours = 8 bits
16 million colours = 24 bits

Once you have calculated the total number of bits required to store the graphic, you need to divide by 8 to express the answer in bytes, divide by 1024 to express the answer in Kilobytes and divide by 1024 again to express the answer in Megabytes

A worked example:

A graphic 3 inches by 2 inches at 600DPI uses 256 colours – how much storage spade is needed for the graphic?

3 X 2 X 600 X 600 = 2160000 pixels

2160000 X 8 = 17280000 bits

17280000 / 8 = 2160000 bytes

2160000 / 1024 = 2109 Kb

2109 / 1024 = 2.06 Mb

What is the difference between the Address bus the Data bus and the Control bus?

The address bus is used to identify a memory location to be read from or written to. It is a one-way bus because all it does is sets up the memory location. The number of lines in the address bus determines the number of addressable memory locations available.

The total number of addressable memory locations = 2 ^ the number of lines in the address bus.

The data bus is used to transfer data from the CPU to memory (Write) or from memory to the CPU (Read) The data bus is a two-way bus. The number of lines in the data bus determines the word size of the computer (the size of instruction it can read at any one time)

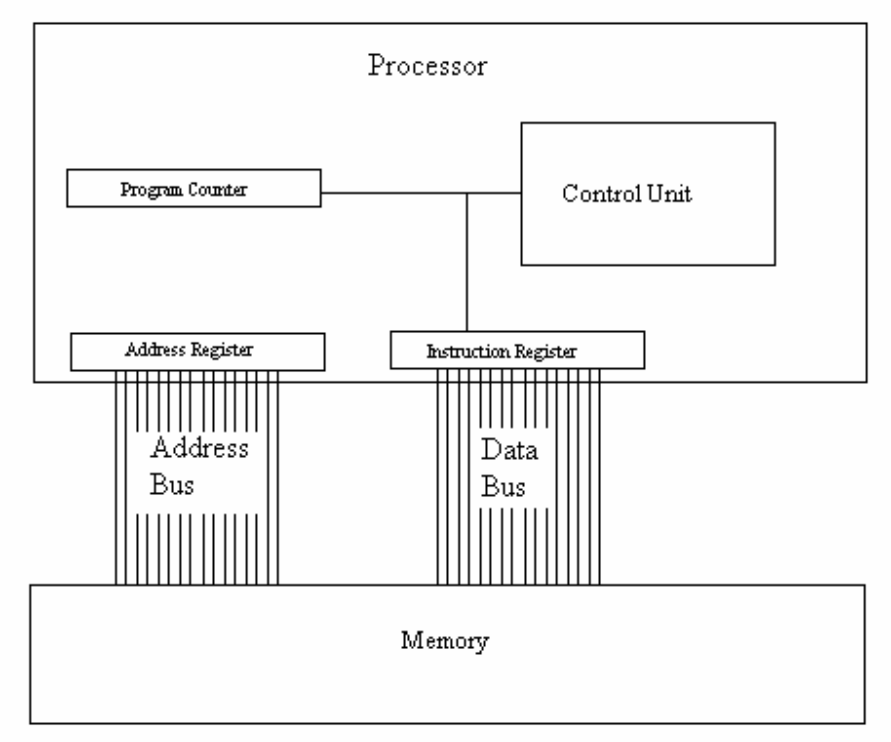
The word size of a computer = 2 ^ the number of lines in the data bus

The control bus is not really a bus at all, but has a number of lines which perform functions like **Read**, Write, **Interrupt** and many others.

What is the Fetch – Execute cycle?

The fetch execute cycle is the cycle which the processor performs in order to read, decode and execute an instruction in memory.

1. Contents of program counter is copied to Memory Address Register (on Address bus)
2. Control bus activates read line
3. Instruction is copied from memory location to Instruction Register (via Data bus)
4. Instruction is decoded
5. Program counter is incremented ready for next instruction
6. Instruction is executed



How do you calculate the total amount of memory available to a processor?

You need to know the width of the address bus, which determines the total number of addressable memory locations, and the width of the data bus, which determines the size of each memory location. Multiplying these two figures together will give the total amount of memory which can be addressed by the processor.

Worked example:

A processor has a 32 bit address bus and a 24 bit data bus. How much total memory can this processor address?

2^32 = 4294967296 locations

Data bus = 24 bits = 3 bytes

Total amount of memory = 4294967296 X 3 = 12884901888

12884901888 / 1024 / 1024 / 1024= 12 Gb

This is the total amount of memory which can be addressed by this processor. In practice it is unlikely that this much memory will be fitted.