

The human computer

Debriefing memory

How realistic was this?

Not so bad.

One important concept that was illustrated here is that a given cell of memory (a person) has no idea what its neighbors contain. So it's not the task of the memory to ensure that consecutive cells read as a correct program. If it doesn't work, not its problem.

Another important thing to remember is that numbers, letters or instructions are stored exactly in the same way: binary numbers. So the memory has no way to know which is which: it is the responsibility of the programmer to write the instructions so that they access what they should (e.g. a JMP should jump to the location of an executable instruction).

Finally, if the memory is unreliable, e.g. writes a wrong value or at the wrong place when instructed to do something which was yet correct, the whole function of the computer can be hampered. More importantly, the error is nearly impossible to detect and recover from, since nothing resembles a binary value more than another binary value. If the JMP now links to a wrong cell, or the ADD tries to add an instruction to a number, so be it.

Of course, in reality, memory is more complex.

In our role-playing, memory cells had a varying size. Cells storing a number stored 2 hexadecimal digits (1 byte), while cells storing instructions stored the whole lot (several bytes). Normally, memory is organized by groups of the same size, usually 2, 4 or 8 bytes (16, 32 or 64bit-architectures). Each of these groups has an address. Incrementing the address by one unit therefore make a jump of several bytes in the memory.

The CPU contains a special set of memory cells called registers that can be read and written to much more rapidly than the main memory area. There are typically between two and one hundred registers depending on the type of CPU. Registers are used for the most frequently needed data items to avoid having to access main memory every time data is needed. Since data is constantly being worked on, reducing the need to access main memory (which is often slow compared to the ALU and control units) greatly increases the computer's speed.

The type of memory that we role-played here is called random-access memory (RAM). This type of memory can be read or written in. There is another important type of memory found in computers: ROM (read-only memory). This type of memory cannot be easily written in or erased, and often contains the very first instructions that a computer should read in order to start up (“the boot”).

Although it is similar, the memory that we role-played here should not be confused with a hard-drive. The first is the main memory of the computer, the one which is used by the CPU to execute programs and store very basic information (not the type that is manipulated by the user). The second, hard-drive, is not compulsory for the computer to work. There are plenty of computers with no hard-drive (a Apple iPhone, for instance). The hard-drive is not part of the main computer, it is rather a device used for to receives information from the outside world and sends results back (this is called an input/output device, other examples of which are keyboard, screen...). For instance, the user’s mp3s are stored on a hard-drive, and the computer can access them. However, in order to play the corresponding music, the computer will have to load this data into its main memory (RAM).

In the memory we role-played here, instructions were stored as a series of letters (M... O... V). In reality, instructions are not stored as words, but as a unique instruction code (the “Op-code”), which is defined by the constructor of the computer’s CPU (e.g. there are Intel-defined codes for each instruction that can be run on an Intel Pentium processor).

Remember:

CPU (central processing unit) = control unity + arithmetic and logic unit (ALU) + registers + basic input and output devices

Most basic computer= CPU + memory

A CPU is usually constructed on a single integrated circuit called a micro processor (e.g. Intel Pentium III). CPU and memory are usually incorporated with other devices on an electronic circuit called the motherboard.

For more information, you can have a look at the role-sheets and debriefing information of the parts played by other people in the class.