

Bits & Bytes

Arkansas' Premier Computer Club



January 2019

Bella Vista Computer Club - John Ruehle Center

Highlands Crossing Center, 1801 Forest Hills Blvd Suite 208 (lower level), Bella Vista, AR 72715

Website: <http://www.BVCompClub.org>

Email: editor@bvcompclub.org

HOW TO FIND US

All meetings are on the lower level of the Highlands Crossing Center in Bella Vista. You may use entrance A on the West side or entrance C on the South side and take the elevator or stairs to the lower level. Turn left (West) to reach the General Meeting room, right for the John Ruehle Training Center. Additional information is on our web site.

MEETINGS

Board Meeting: January 14, 6:00 pm, John Ruehle Training Center

General Meeting: January 14, (2nd Monday), 7:00 pm, Community Room 1001.

Program: Joel Ewing will present a program on "Cutting the TV Cable Cord"

Bring a guest! New Members and Guests are always welcome at the General Meeting

Genealogy SIG: January 19 (meets 3rd Saturday of the months).

MEMBERSHIP

Single membership is \$20; \$10 for each additional family member. Join by mailing an application (from the web site) with check, or complete an application

HELP CLINICS

Saturday, January 5, 9am – noon
Wednesday, January 16, 9am – noon
Saturday, February 2, 9am – noon

Help clinics are a free service for BVCC club members, held in the Training Center

Bring your tower, laptop, tablet or smartphone for problem solving.

CLASSES

"Computer Security for Regular People, Part 1" – Justin Sell, Tuesday, January 15, 6:30 – 8 pm
Part 2 will be offered again on 3rd Tuesday in February.

"Why, When and How to Backup Your C Drive" Wednesday, January 23, 9 am - 11 am

"Slow PC? Let's Upgrade or Buy New"-- Tuesday, January 29, 9 am - 11 am

Advance sign up required for classes: Contact Grace: email to edu@bvcompclub.org, text 469-733-8395, call 479-270-1643, or sign up at the General Meeting on September 10. Classes are **free to Computer Club members** and are at our John Ruehle Training Center.

Check the monthly calendar and announcements for any last minute schedule changes at <http://bvcompclub.org> .

COMPUTER FUNDAMENTALS

By Dan Douglas, President, Space Coast PCUG, FL
March – June 2018, The Space Coast PC Journal
www.scpcug.org
datadan (at) msn.com



This merges into one article a series of four “Computer Fundamentals” articles written by Dan Douglas during 2018. The goal was to explain fundamentals of computers, programming and usage that will hopefully deepen your understanding of how it all works and why things are the way they are.

What is a Computer

Computing is all about taking some input, doing something with it and producing some output. How this evolved into today’s pervasive technology is amazing.

Herman Hollerith was a US Census worker leading up to the 1880 census. At that time all census data was counted by hand and he recognized a need for a tabulating machine to reduce the time it took to summarize the information. The input was the census forms, a summary of the data is what needed to be done with the input data and the output was the census reports. He invented a tabulating machine that used punch cards that were like those already in use by automated loom machines. The company that he formed, the Tabulating Machine Company, became a great success until the 1910 census when a Census Bureau technician improved on the Hollerith design and removed the monopoly that had been in place. The Tabulating Machine Company later became a company you may have heard of: International Business Machines Corporation (IBM).

If we jump forward a hundred years, what do we have for 1980 state of the art computing? Computers that take up a large room with special power and cooling requirements. Large cables connecting the main processing unit to storage devices that were typically disc drives, tape drives, or some other medium. Input was in the form of punched cards (still!), data stored on storage media or operator terminal screens. Output took the form of storage media as in updated files or printed report output, or as a screen display of information. Computer programs were no longer coded on punched cards, but rather as files (think .exe files) that were executed by operator commands or by automation tools. Networking was done by dedicated circuits where security was important from point A to point B. There is a mainframe Operating System (OS) in control of the processing unit. This is the large company environment of ‘mainframe’ computers still in use throughout the world today.

What came after that was essentially the miniaturization of that mainframe environment into a desktop sized machine. We have a similar OS with either Windows or Mac that runs today’s PCs. The PC uses a Central Processing Unit (CPU) typically made by either Intel or AMD that performs the instructions that programs (those .exe files) tell it to process. The input of today takes the form of icons that you click or touch on a display screen, or by automated tasks that run programs periodically. Storage is on your hard disc drive or USB memory stick or SD card. Output can be information displayed on your screen or a report that you print or files that you save on your hard drive.

Here are the core components that every ‘computer’ type of device have always had:

- a CPU or processing engine of some sort to interpret instructions and to do something - one or more input sources (a button to press, a screen to touch, a sensor, a file to read, etc.)
- one or more programs to take some information or data and do something with it
- one or more outputs (printer, files stored on some media, a screen display, etc.)

Next, we'll look at programming and how your OS controls everything.

How Your Operating System Controls Everything

Now we will examine two core parts that make any computer more than an assortment of electronic components:

- the operating system (OS)
- a program (app or application)

Both are software, that is many lines of programming code that performs a set of functions that must be loaded onto a hardware platform to operate. Hardware is the real set of electronic components that make up a computing device. A typical hardware desktop has the following core parts:

- a motherboard
- a central processing unit or CPU
- one or more memory chips
- storage devices (DVD/CD/hard drive, etc.)
- a video display
- input devices – keyboard, mouse, touchscreen, etc.

The OS is the programming code that allows the hardware to do something. Without an OS you get a blinking cursor and nothing else. It is often called the system software to differentiate it from application software. The OS allows the hardware to process application programs written in any number of languages to operate on a given hardware platform. In essence, the OS translates certain instructions from the app to perform specific tasks using the hardware. The OS controls hardware devices using 'drivers' which translate instructions into commands that a specific piece of hardware can perform. The OS also controls the allocation of the system memory, usually called Random Access Memory or RAM. RAM is the memory sticks or chips that every computing device contains that is required to load instructions into to trigger some activity. Typically, a desktop or laptop computer contains around 4GB today. All the most popular operating systems can also allocate a file on a disk drive to serve as 'virtual memory' which effectively doubles the amount of memory available for the OS to assign to program tasks.

An app is the piece of software that works in conjunction with the OS to allow a user to process some input, do something with it and produce some output. Each app on your smartphone is a specialized piece of software designed to do something such as look up the cheapest gas stations to my location or tell me what the movie times are at the local cinema. The user interface on a PC typically uses icons (those shortcuts on your desktop) for the user to click on, using a mouse or a keyboard or to touch on a touchscreen, to start a specific program. The OS intercepts the start action and loads the program into the correct amount of memory for it to execute. The program will ask for certain files to be opened or created and the OS uses device drivers to access the storage media or the network to create or access the file information. The program will then take some action using that data, such as displaying some information, calculating something using the data, or printing a map. Perhaps you will choose to print the results of the data being processed in which case the OS will use a device driver to pass the desired report to the printer in a format that that specific printer can understand and handle as a printed report. As a final step you may instruct the program to save any updates that you made at which point the OS will again access the storage media and tell it to write the data in a file.

The Evolution of Operating Systems and Basic File Systems

Prior to the mid-seventies, the term Computer typically referred to one of two types of computing devices. The oldest was the early types of accounting or tabulating machines. These machines were the foundation of the IBM computing business for many years ranging from the 1940's to as late as 1976. They were relatively simple machines that could read numbers from punch cards, manipulate the numbers and print the results. Some of the

'programming' consisted of boards that had patterns of electrical circuits that were selected to perform predetermined functions (add, subtract, etc.) by connecting them by placing jumper wires between functions. The operating system on these machines was essentially fixed in place for a single purpose – to perform accounting functions.

The series of computers during the 1960's contained what we would consider today to be a proper operating system that required the more advanced capabilities of the post vacuum tube generation of computers. The IBM 360 series of computers started with a DOS/360 OS, which just like a PC DOS system, could only perform one task or program at a time. All the memory installed on the computer was essentially used by the executing program. IBM soon evolved the DOS/360 OS to OS/360 which became the most popular multi-programming system. There were two main variations of this:

- Multiple Fixed Task (MFT)
- Multiple Variable Task (MVT)

MFT allocated the physical memory into fixed regions or partitions. This allowed two or more programs to run simultaneously – true multi-tasking! The computer operator had to make sure that each program had its memory requirements known prior to trying to run it or it would not be able to load properly. It was also possible through operator commands to reallocate the memory as needed to create one large partition for the really large programs. MVT was used on the larger mainframes where memory could be allocated as needed by the program dynamically and each program ran in its own memory space. This OS has evolved many times since the 1970's but the essentials are the same.

A technique of using virtual memory reached the mainstream OS during this period. Virtual memory used external storage space, such as a disc file, to extend the physical memory space to typically double the memory available. Each program uses only a small portion of program code at any point in time, so virtual memory allocates the required memory into virtual pages that can be written to the disc file when not needed and then swapped back into real memory when required. Your PC or Mac uses this same process today.

File creation really matured with the release of OS/360 which allowed an organization to name the files into directory layers like a project.user.filename structure. A computer file is a method for recording data in a computer storage device. Just like words can be stored on a page, information be written to a computer file. Here's how Wikipedia describes computer files: "There are different types of computer files, designed for different purposes. A file may be designed to store a picture, a written message, a video, a computer program, or a wide variety of other kinds of data. Some types of files can store several types of information at once.

By using computer programs, a person can open, read, change, and close a computer file. Computer files may be reopened, modified, and copied an arbitrary number of times.

Typically, files are organized in a file system, which keeps track of where the files are located on disk and enables user access."

In a Windows OS, the file name is usually of the form name.type. The name can be used to easily identify what the contents of the file may be about, and type is used to determine the correct program to be used to access that file. A file called 'my best song.mp3' tells the Windows OS to open this file with the default program used to play .mp3 file types, such as with iTunes or the Windows media player program.

There is a lot more to files, such as what are databases and the various methods of accessing and storage of files (formatting, NTFS, FAT, FAT32, etc.) but that's a topic for the more technically inclined audience.

Common Program Languages That Programmers Have Used

Now we will examine just a few of the more common well-known programming languages out of the hundreds developed. Do you realize that a player piano is one of the early forms of programming? Programming started by being a way for a human to instruct a machine to do something.

In this case, what started out as a non-programmable device, the piano which had been around for centuries, became a programmed computer when modified to strike keys when a sensing mechanism detected holes in a roll of paper. The position of the hole told the machine which key, or keys, out of the possible 88, to strike to produce a note.

This kind of programming was tied to a certain type of automation machine and could only work on that specific machine that it was designed for. You couldn't take a roll from a player piano for example and use it to automate a drum playing the same tune. Wikipedia lists this as the first known example of programming - "the automatic flute player described in the 9th century by the brothers Musa in Baghdad."

Very early actual computers used no programming at all, but rather connected bits of circuitry to calculate and produce a result. This evolved into the 1st generation of programming languages, often referred to as machine language coding.

These programs were written in either decimal or binary format and read into the computer through toggle switches, punched cards or on magnetic tape. They were simply the expression of the old connected bits into logical operations doing the same task. An example would be to take the contents of storage area 1 and add it to the contents of storage area 2.

The 2nd generation of programming languages are called Assembly languages. They were tied to a specific architecture and instruction set of a brand or model of computer but did provide for human readable code that saved many hours of programming time and reduced the chance of errors.

Next came the first of the 'high-level' languages which formed the 3rd generation of programming languages. In 1954 FORTRAN was invented and is still used to today as a high-performance language. In many cases it reduced the complexity of a program by a factor of 20. FORTRAN is derived from the words formula translator and is representative of the main focus of FORTRAN: to aid with numeric computation and scientific task procedures.

The 3rd generation of languages used a compiler function which would take the relatively slow performing high level language and translate it into machine language for faster execution. During the late 50's there was a recognized need for non-scientific languages for use in the business community. Grace Hopper developed a program language called FlowMatic which addressed this concern and led directly to the development of Cobol (Common Business Orientated Language) in 1959, which became one of the most popular programming languages which is still in use today. The Department of Defense (DoD) standardized on COBOL in the 1960's and required it to be supported on every Computer purchased by the DoD. The goal was to have a portable language base that would allow for a program to be employed in multiple instances without having to rewrite it for each installation.

It is only during the late 1990's that programming languages such as C and C++ finally evolved to where languages such as Java became the primary language that truly allowed for 'write once, run anywhere' capabilities. This is why web pages can run on any computer using standard browsers independent of whether the hardware is from Apple, Dell, HP or any other company.