Bits & Bytes

Arkansas' Premier Computer Club

September 2025

The Bella Vista Computer Club - John Ruehle Center

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

Website: http://BVComputerClub.org **Email: BVCCeditor@bvcomputerclub.org**

MEETINGS

Board Meeting: September 8, 2pm, in John Ruehle Training Center, Highlands Crossing Center.

General Meeting: September 8, 3pm. Program: "A Look At Artificial Intelligence", presented by Dave Floyd, AI Consultant, and includes how to use Generative Pre-trained Transformers (GPTs) in everyday life, their strengths and weaknesses; and using AI for art, video, and music content creation. .

We will meet in-person in John Ruehle Training Center, Highlands Crossing Center, lower level, 1801 Forest Hills Blvd, Bella Vista, or you may attend the meeting on-line via Zoom. Zoom access information is published on our website.

Visitors or Guests are welcome.

Consider attending by Zoom if you are unable to attend in-person.

HELP CLINICS

September 6, 9am - noon at John Ruehle center
Septemher 17, 9am - noon at John Ruehle center
Members may request Remote Help on our website at https://bvcomputerclub.org at menu path Member Benefits ►Remote Help .

MEMBERSHIP

Single membership is $30; $15 for each additional family member in the same household.

Join on our website at https://bvcomputerclub.org at menu path Get Involved ►Join/Renew, by mailing an application (from the web site) with check, or complete an application and pay in person at any meeting.

CLASSES

(At BVCC Training Center)

"Introduction to GIMP", Thursday, September 18, 1pm - 4pm, with Joel Ewing.

Advance sign up required for each listed class: For reservations: email to bvccedu@bvcomputerclub.org, or sign up at the General Meeting. Classes are free to Computer Club members.

Check the monthly calendar and announcements for any last minute schedule changes at https://bvcomputerclub.org .

NEW OR RETURNING BVCC MEMBERS

We are pleased to welcome the following new members or members returning as BVCC members after an absence:

Darlene Kuta	Jan Olson	Lee Hartwell
Charllene Hartwell	Marge Arnsdorff	John Terrill

THE ORIGINS OF PROGRAMMABLE DIGITAL COMPUTERS (PART 2 OF 2)

By Joel Ewing, President Bella Vista Computer Club
Bits & Bytes, September 2025
htttps://bvcomputerclub.org
president (at) bvcomputerclub.org

(continued from Part 1 in previous newsletter)

The Evolution of Operating Systems

The earliest computers were dedicated to one application program at a time. Each program was manually provided to the computer by some relatively slow mechanical process, such as loading from punched paper tape or punched cards. Then as the application ran, it would typically require data to be read from a similar source and output printed or data saved to an external media. It soon became apparent that much expensive processor time was wasted waiting on slow input output devices.

The earliest Operating Systems, also sometimes named "monitors, were designed to speed up the input and output process of an expensive computer by allowing its fastest devices, typically magnetic tape or a disk storage device to be used for staging input and output, with a slower, cheaper computer system used to transcribe the data between the faster media and slower devices like card equipment or printers.

The hardware architectures for large computers eventually evolved to where input/output operations were performed by many semi-autonomous specialized computers (which IBM called I/O "channels") with independent access to computer memory and which could act concurrently and manage multiple actions on devices under its control.

Frequently a single programs or application could not effectively use all the memory or other resources of a large computer system, so Operating Systems evolved to allow "multi-programming", where multiple programs or applications could reside in memory at the same time, switching resources among the programs as needed to improve hardware resource usage and the total amount of useful work that could be done. Sometimes a single application might involve multiple independent computational processes that could be allowed to proceed in an overlapped fashion, and this generalization of the multi-programming concept to also include multiple tasks from the same application was called multi-tasking (also called multi-threading in the UNIX and PC world). When there were more tasks waiting for a general-purpose processor than processors available, a processor was quickly switched between different tasks giving the illusion of parallel processing through time-sharing. When the physical hardware had more than one general-purpose processor available so that multiple tasks could physically run at the same time, that was also called multi-processing.

In order to safely support and control multi-programming and multi-tasking environments, the hardware architecture must provide the means for the Operating System to prevent one application from altering the Operating System itself, altering other applications, or from directly controlling resources that were allocated to other applications. The Operating System also has to be designed to strictly observe the required protocols and software "locks" so that two different tasks could not attempt to modify the same exclusive-use software resource at the same time. Failure to handle shared resources properly invariably leads to random, timing-dependent failures that are difficult track down and fix.

One of the most powerful features of modern mainframes is their ability to run large numbers of scheduled batch "jobs", each of which may have multiple program steps, in a mostly automated fashion, taking into account dependencies on other jobs and events, with verification of successful job completion, handling notifications in the event of failure, and providing support for restarting failed jobs after problem resolution. Literally thousands of jobs can be run daily and reliably monitored and controlled under the oversight of only two or three "operators". Most users or customers that interact with a large mainframe only see and judge performance based on their individual transaction responses; but a large part of the work is typically done behind the scenes in batch processing runs that may have daily, weekly, monthly, annual, or on-demand schedules.

The PC Revolution

The Space Race in the 1960's undoubtedly deserves some of the credit for the development of microprocessors, because ways were needed to make computers very small and use less power for use in space craft. Motivation also came from the desire to build more complex and faster computers while keeping size and power consumption manageable. It turns out that making transistors physically smaller can not only make them take up less space but increases their speed and reduces power consumption. The problem is that just making the devices small also makes all the interconnections between devices more difficult. The solution was to fabricate an entire circuit with multiple transistors, resistors, and capacitors, and all their interconnections on the same silicon chip, creating an "integrated circuit" (IC). As techniques improved, it became possible to create large-scale integrated circuits with thousands of transistors on a single silicon microchip. There is some dispute who should be credited with "inventing" a complete computer processor (or "microprocessor") on a single chip, but the first that was a commercial success was the Intel 4004 in 1971, a 4-bit processor with 2,300 transistors, widely used in electronic calculators and for control applications. By 1980 a number of different companies had produced more than a dozen different microprocessors, including the 16-bit Intel 8086, the first of the x86 family of processors that now powers most of the "IBM-compatible" PCs today, and the 32-bit Motorola MC68000 in 1979 that was used in the Apple Lisa and Macintosh and the Atari ST computers.

Although home hobbyists were playing with home computer kits as early as 1974, the first Personal Computer that was a commercial success was the Apple II computer in 1977, which supported a monitor, a keyboard, two 5¼" diskette drives and a printer. In 1979, Visicalc released the first spreadsheet program, for the Apple II computer. This allowed companies to cost-justify purchasing an Apple II and was a key factor in its success. The Apple II was popular enough that cheaper clones were manufactured by other companies – until the courts ruled that computer code "published" in computer read-only memory chips was also protected by copyright laws.

Apple II Computer

By 1980 it had become clear that the market for personal computers was hundreds of thousands of units, and IBM decided to enter the market. IBM's existing design and marketing strategy for large business computers was a poor fit in the rapidly changing PC environment. To produce a competitive product in a short time frame and reach non-corporate buyers, IBM was forced to use off-the shelf and non-proprietary components and software, sell through non-IBM retail outlets (ComputerLand and Sears), and publish enough of the specs so that 3rd parties could design interface cards, peripheral devices, and programs for the computer. The IBM PC became available in August 1981, with an Operating System and BASIC programming language interpreter from Microsoft. Sales greatly exceeded IBM's expectations, with more than 750,000 machines being sold in 1983. Within a year four times as many software packages were available for the IBM PC than for the Apple Macintosh, and many other companies competed to sell hardware components. The IBM PC effectively established an industry standard for Intel x86 microprocessor-based computers.


IBM PC

The design decisions made by IBM meant that IBM didn't own many patents for the IBM PC, and within a few years their open documentation of many features enabled other companies to market 100%-compatible clones that eventually dominated the IBM-compatible PC market. By 1990, IBM's revenue share of the global PC market was estimated to have decreased to only 16%. Other manufacturers could adapt to new technology faster than IBM, and the IBM-compatible clone manufacturers and 3rd-party software manufacturers became the ones that established future standards for IBM-compatible-PC hardware and software.

Apple computers took a different path and kept their hardware design closed, eliminating most hardware cloning and 3rd-party hardware enhancements. This made Apple computers more expensive but also more stable, as there were fewer possible hardware configurations and combinations, and hardware design quality was totally under Apple's control. The Apple Macintosh introduced in 1984 featured the first commercially successful graphics user interface (GUI) adding a graphic desktop with icons, and a mouse for control. The Macintosh II introduced in 1987 expanded the GUI to display 256 different colors on a color monitor. The greater hardware capability to support the graphics, plus the impact of Apple's proprietary hardware priced many of these system above $5K – out of the reach of most individuals, but very popular in a niche market of professionals needing to manipulate digital images or videos.

Microsoft was inspired by Apple's graphic user interface to create their own GUI running under MS-DOS in the early 1990s, with Windows 3.0 in 1990 and Windows 3.1 in 1992. This was the last MS Operating System designed for a 16-bit processor – probably because the 16-bit x86 architecture quickly proved inadequate to support graphics and image processing with higher resolutions and full-color support. In 1995, Windows 95 minimum requirements included a 32-bit x86 processor.

There were some other serious contenders for Operating Systems for x86 architectures, Open Source Linux with X11 windowing in 1991 (similar to Unix but free, and popular in educational and scientific environments), and OS/2 & OS/2 Warp (1987-1996) (superior in design and more stable than Windows 3.1, but largely restricted to corporate networks). By various agreements, Microsoft was able to pressure many equipment manufacturers to bundle Windows with hardware, and by the late 1990's Microsoft controlled 90-95% of the PC Operating System market.

Unix systems (AT&T System V, BSD-based systems) were also widely used by 1990 in academia and technical computing areas. Unix was closely involved in the evolution of the Internet and Open Systems and in the setting of standards in scientific, CAD and graphics applications. The first graphic web browser was created for Unix using X11 windowing in 1992.

Mainframes Versus PC Server Platforms

By the mid 1990's the power of microprocessor chips was increasing year by year and power of PC servers increasing with little increase in cost to the point where pundits began predicting the imminent replacement of mainframe computers by farms of PC-based servers within five years. That prediction periodically resurfaces, but continues to prove wrong because mainframe technology hasn't stood still either.

Just in terms of raw processing power, by the mid 1990's there were microprocessors that could execute instructions at rates faster than mainframe CPUs, but there were a number of reasons that wasn't a valid comparison. Mainframe instructions tended to work with larger data items than the x86 architecture, a great advantage when working with large numbers of corporate data records or with scientific calculations that required greater precision. Mainframe aggregate input/output data transfer rates were much higher, and a significant part of the management of physical I/O transfers was offloaded from the CPU to I/O channel processors. Memory had more access paths, wider data buses, and higher bandwidth so even high I/O rates didn't impact CPU memory access. The mainframe Operating System was designed to dynamically manage the CPU resource so that "favored" applications could get adequate response times even when the CPU utilization was close to 100% – the typical operating systems on x86-based servers could have transaction response issues even at 50% utilization. To support critical applications, mainframe hardware was designed with redundancy, designed to catch and diagnose hardware failures, and in many cases allow for continued operation and dynamic repairs.

The possibility of inexpensive PC server farms as an alternative to mainframes was attractive, especially for a small business with modest budget, since you could start small and incremental increases for hardware and software appeared low-cost compared to mainframes. The problem was that if you had one large application that required the total power of many servers, there may not be any practical way to efficiently split that application across multiple servers. If a business had a potential for significant growth, multi-server approaches proved more likely to hit a point where further upward scaling was no longer possible without changing to a completely different approach. It all depended on whether the application was a good fit for a server farm and how much coordination overhead among the servers was required. There were success stories in moving application from mainframes, and also some significant failures – cases where a non-mainframe solution ended up costing more rather than less, and even cases where the migration had so be abandoned because of poor performance.

If you have an 18,000 lb log to pull, would you rather have the help of one elephant or 100 husky dogs that could each pull 180 lbs?

The comparison metrics began to change significantly in 1994. Until 1994 IBM large mainframe multi-processor systems required special computer rooms with raised floors to accommodate under-the-floor chilled air, power cables, chilled-water hoses, and I/O device cables. The technology had kept improving from the 1960's with denser and more complex integrated circuits, but achieving high-end performance required a bipolar transistor technology that used more power and was physically larger than the slower CMOS solid state technology in PC microprocessors. High power meant higher heat production from processor chips. That in turn required computer centers with circulating chilled water systems to remove heat from a separate circulating water system in the computer that cooled heat sinks inside the processor to dissipate the heat. The isolated internal water system was needed to keep the computer's non-conducting distilled water free of contaminants.

By 1994 IBM was finally able to produce large IC chips with smaller CMOS transistors that performed at speeds close to that of the bipolar mainframe technology. In 1994 IBM delivered their first mainframe using that power-efficient CMOS technology. Overnight the size of large IBM mainframes dropped to the size of a large refrigerator, power requirements dropped, and the need for computer room chilled water systems for cooling was eliminated. Both the hardware costs and operational costs for mainframe CPU power was substantially reduced.

A single CPU of the first generation of IBM CMOS mainframes was still somewhat slower than the older technology, and that could be a negative for some workloads; but in 1997 that changed when generation 5 of the the CMOS system became available with CPUs that were finally faster than all IBM's older mainframes. All the technical reasons for not switching to the newer processors were gone, it was now just a matter of corporate finances and contracts.

The speed and capacity of IBM CMOS systems has continued to be enhanced. In 2025, an IBM z17 has hardware-accelerated AI capabilities, can be "partitioned" into a maximum of 85 independent machines, each capable of running an Operating System (including Linux), can be configured to support several hundred processors of various types, and many terabytes of memory that can be allocated among the partitions. In effect, one mainframe server box can now run its own large server farm.

IBM z9 and IBM z10 Processors (2011 jce)

It is no longer generally believed that mainframes will be made obsolete by farms of cheap small servers; or for that matter that mainframes will be enhanced to the point that the PC and other server platforms are no longer needed. Each has its own strengths and weaknesses, and best practices in corporate computing is to use a hybrid approach with both. Application software should not be arbitrarily replaced just because it is old if it is still performing a useful function and can be maintained. Application software should not be migrated to a totally different architecture just because it is running on older less-efficient hardware without first considering a simpler hardware migration to more cost-effective versions of the same hardware architecture.

It is unclear from published reports, but a recent on-line article about the wide-spread flight delays encountered by United Airlines on August 6, 2025 may be another of those cautionary computer application migration tales. The report blames the delays on problems with United's Unimatic computer application, which was referred to as a "legacy" system from the 1970's. "Legacy" is frequently a pejorative used for what a writer considers "obsolete" applications running on "obsolete" mainframes. According to ChatGPT, United's Unimatic system was an IBM mainframe application running under z/OS. Unimatic was supposedly phased out beginning in 2012-2015 and fully retired by 2017, to be replaced by cloud-based and web-based applications. That this recent failure was attributed to Unimatic indicates that 8 years after its intended retirement it had not yet been possible to replace some of the functionality of the mainframe-based application, and that United is still paying mainframe costs plus costs for the new systems. ChatGPT indicated that in Q1 2024 United's CFO confirmed that elimination of the mainframe was still a goal, but that at that time as much as 30% of the computer workload was indeed still running on the mainframe. It is unfortunately common that when a company commits to a massive migration plan to a different computer platform, by the time problems, unanticipated costs, and delays are encountered the budgets for the older systems have already been reduced: the old systems may no longer be kept up to date, and experienced support personnel for the old systems have either left for new jobs or been diverted to the new systems, making the old system both more vulnerable to failures and more difficult to fix.

The Future

The latest mainframes and the latest PC microprocessors both advertise hardware AI support. There are some applications where statistical inferences are useful, and no doubt AI assists will prove useful for those. There are many other applications (payroll management, tracking financial assets and transactions, general record keeping, etc.) where results must be 100% correct and statistical inference is of little use, except for spotting variations that might signal suspicious activity. AI hardware is not going to end the need for conventional digital processors, but most likely be a co-processor used by a conventional processor.

Another area hyped by the press is quantum computing. If some serious technical issues can be resolved, a hybrid conventional/quantum super computer could in theory be constructed that could solve problems that would take too long to solve today. Fault-tolerant, large-scale quantum computers are still believed to be decades away. It will probably take at least another 5 to 10 years just to see if a large-scale quantum computer is possible, or if quantum computing will continue to be isolated to a few narrow applications. Just to be on the safe side, new encryption techniques are being devised that would resist decryption by future quantum processors. If they become commercially viable, the most likely scenario is that a quantum processor would also become a co-processor of a conventional processor.