Personal Technology Journey 1970-2021
John Hastings - 4/15/21

Preface

I got my first professional programming job in 1971 so this year marks my 50th year as a computer programmer, teacher, sales representative, database designer, website developer and computer consultant.

This history describes my personal technology journey and also discusses various computers and technologies that I had firsthand experience. My goal is to help the reader gain some appreciation of the phenomenal amount of change in computer technology that has occurred in the past 50 years.

At times, I present considerable technical detail that may not interest some readers. My main concern is to show the rapid change in the technology. I encourage you to read the **TECHNICAL ADDENDUM** at the end of this document for technical descriptions and general information. Take time to look at the pictures to get an idea of how computers have changed not only in speed and power but also in shape and construction.

In the Addendum there is some general information about various computer systems, programming, computer languages, and other technical information.

Most readers may not understand some or all of the technical material, but my hope is that even a quick reading of this information will provide insight into how much computer technology has changed and how it impacts all our lives.

Science fiction author Arthur C. Clarke created three "laws" about technology. His Third Law states that "Any sufficiently advanced technology is indistinguishable from magic."
(https://en.wikiquote.org/wiki/Arthur_C._Clarke)

My interpretation is that any technology you do not understand is magic to you. This is not a problem if you are able to do the correct "magical incantations" to get the magic to work! The problem usually comes when the magic fails to work, and you have to consult with an experienced technology "magician".

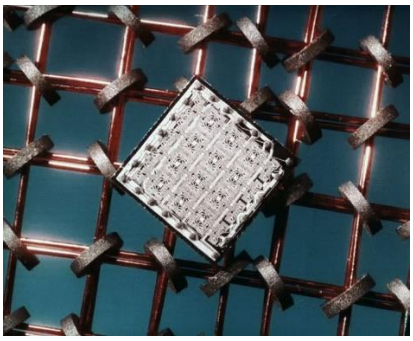
Do not be embarrassed to ask for help. Good magicians will try to help you learn something about the magic so you can improve your technology magic skills.

My journey starts at a time before there were handheld calculators or personal cell phones or flat TVs or personal computers. Computers were in the news but only a very few people worked with computers. Most of those that did work with computers used terminals to type information into forms and get reports. Computers were so expensive that only governments and very large companies could afford them.

It is hard even for me to fully comprehend how much change the computer industry has brought about in my 50 years. The first computer I learned on had 16,000 bytes of memory and had several parts each the size of my desk or larger. My current iPhone has 256 BILLION bytes of memory and fits in my shirt pocket.

1970 Things Get Interesting

It turns out that 1970 was an important year in the development of computers. Intel created the first "computer on a chip" or what is now called a microprocessor setting the stage for a computer cheap enough to be a personal computer. In 1970, Fairchild created the first "memory on a chip" or integrated circuit RAM.



RAM Memory - Magnetic Core (Donuts) vs Integrated Circuit

The Magnetic Core memory in the picture was made up of tiny donut shaped magnets with wires threaded through the hole. This was done by people and the process could not automated so it was very expensive and had very limited capacity. The silver square is the same type of memory, but it was "printed" on the chip called an Integrated Circuit (IC). This change allowed the manufacturing process to be automated and enabled the size of memory to grow at the same time the costs were dramatically reduced.

Computers then and now are made of a set of IC chips each doing a special task (computing, RAM and ROM memory, external device controllers, and other support chips). These external device controllers manage things like printers, monitors, mice, WiFi, disk drives, and other things. All of these IC chips are mounted on a single circuit board that has wires (printed on the board) that run from chip to chip to allow signals to flow as needed.

The development of the Transistors and Integrated Circuits was the most important and driving force behind the computer revolution. See the Addendum for information about Moore's Law.

HP 35 Calculator introduced in 1971

The HP calculator was revolutionary especially for scientist and engineers. I have a collection of slide rules that I used in college math classes, but they have been gathering dust since the pocket calculators before I got my first programming job.

Literally, in my lifetime the technology achieved or exceeded what was considered science fiction in 1970. During those years computers were so limited that some of the things we take for granted were things I did not think would happen in the next 50 years, but they did!

Things like giving voice questions to Siri or translating between different languages or identify people's faces, were considered so complex and would require such huge computer capability (blazing speed, tons of memory, extremely sharp display screens, etc.) that no one really expected them to be a reality by 2020. See the Epilogue for a more detailed discussion of this.

Many everyday items like refrigerators and toothbrushes have been made "smart" by including a computer in them and connecting them to the electronic highway called the Internet. This process is known as the "Internet of Things" (IoT) and would only be possible because the computer has become very small, very cheap, Internet WiFi, and very reliable.

My Computer Journey Begins

I was introduced to computers at Southwest Texas State University (SWTS) in San Marcos, Tx in 1970. I enrolled in Jan 1970 to complete my undergraduate degree in Mathematics with a public-school teaching certificate. The school is now called Texas State University.

My first semester at SWTS I had time for an elective and choose an introduction to Computer Programming courses offered by the Math Department. At that time, the University of Texas in Austin and Texas A&M had a computer degree plan (undergraduate at least) but SWTS did not have a Computer Department and only offered about 7 electives courses in the Math Department and a few in the Business Department.

I started taking classes the second semester that they were offered. I am not sure if I took enough courses to qualify for a minor in Computer Science, but I was close by the time I graduated in Dec 1971.

In 1970, most computers were either large and expensive "mainframe" computers or special purpose computers. Computers required many specially trained operators, programmers, designers, and repair technicians to function. They also required special air conditioning and electrical systems to operate.

Prior to my arrival, SWTS had acquired two computers for teaching purposes – a government surplus computer called the Athena which was designed solely to launch military rockets and an IBM 1130 aimed at "low cost" education and engineering markets.

The introductory course taught the FORTRAN programing language which was intended for use in science and engineering applications. The business courses were taught using COBOL programing language which was the standard in accounting and business applications.

FORTRAN and the IBM 1130

FORTRAN ran on the IBM 1130 and used punched cards for program and data input. FORTRAN stands for Formula Translation and was widely used both in education and industry at that time. It is still in use today partially because of the expense to convert working programs written in FORTRAN 40-50 years ago into a modern computer language.

Since the computer courses had a lab time, I was able to go to the lab and keypunch my programs onto cards which I feed into the computer. After the program was finished, I looked at the printed output and determined if the results were what I expected or if there was a "bug" or error that needed to be corrected.

Helen was working at the university library and we only had one car so many days I was able to spend time in the computer lab doing extra programs or reading computer books from the library till it was time to meet her after work.

I did not think about it at the time, but I realize now how important that time in the lab was as I was learning this new skill. Learning any skill takes practice and repletion. As time went on, I found I did not have to look at a manual or textbook to know what the programming commands were needed. In fact, the actual process of "writing code" became almost automatic so I could focus on the problem and the steps required to create a solution (software program).

My instructor was an Electrical Engineer who had worked in industry prior to coming to SWTS. I am really fortunate that he became a mentor to me and was encouraging me to explore and push myself to do extra learning activities .

Also, during that first course I realized that programming was fairly easy for me. In a way programming "made sense" to me. Computer courses are like Math in that there are very specific terms, rules, procedures, etc. that must be followed exactly to get the computer to produce the desired result.

I had grown up playing card games and board games which I think helped me with problem solving and strategy skills which helped me learn computers. My love of math and science also fit well with the process of learning technology and computers.

A distinct memory of that class is that the textbook had a chapter at the end of the book that described and gave the code for a 3-dimensional tic-tac-toe game (4 levels each with a 4x4 board which allow the winner to do vertical, horizontal, and multi-level of 4 tokens in a row) that you could play against the computer.

I typed the program on to cards and put it in the computer so I could play against the computer (I think I tied a few times but never won a game). The book not only listed the code but spent considerable time describing the logic behind the code. It was the first "non-trivial" program I worked with.

I took some other programming courses that used FORTRAN, but the details are not important. What was important is that over 18 months of courses, I got lots of programming practice including being selected to be a computer lab assistant.

Machine Language and the Athena Guidance Computer

The IBM 1130 was a "current technology" computer in 1970 but the Athena was definitely obsolete by the time it got to SWTS (see the short history of the 1130 and Athena computers in the Addendum). It did not have a language like FORTRAN and all instructions and data were entered as numbers (machine language).

All computers work by interpreting numbers to do the various operations such as add, input, test, etc. Computers also store "data" such as the actual numbers to be added (like the number 15) The Add instruction might be something like 650402 which might mean Add the contents of memory register 1 to memory register 2. On the Athena you could either enter the numbers and data by

- pushing buttons on the large console one at a time and then hit "go"
- loading them in from small Drum Memory
- feeding them from punched paper tape on the Teletype

Feeding program instructions and data into a computer by entering a series of numbers one by one was very tedious, very prone to errors, and very hard to test. Early computers had programs called Assemblers which would take commands like "ADD Register A to Register B" and the Assembler software to do the lookup to determine the correct number the computer needed.

In other words, programs were written in an alphabetic Assembler Language that software could translate into the Machine Language that would be the final program to run. The Athena was so primitive that there was no Assembly language available for it.

An interesting aspect of using the Athena computers was that the console showed all of the internal operations using the push button lights on the panel. It also had a knob that allowed you to slow down the speed of the computer so you could watch the different operations take place in slow motion.

I enjoyed the challenge of working with the Athena so much that I wrote some utility programs for the machine on my own time. I have always thought that having used a really primitive computer was a big help later on because I had a better understanding of the inner workings of all computers than most programmers.

This educational experience with both an 1130 "personal computer" and the primitive Athena was very helpful in several jobs I was to have later in my career.

Heat Transfer and the Hewlett Packard 2100

I was taking some final courses in Summer School to complete my degree when I got a change to interview for a full-time programmer position with a company in San Marcos.

The company sold products that were used to heat the industrial pipes and other equipment in refineries, pipelines, chemical plants, and other industrial applications. For example, to pump sulfur through a pipe system the pipes must be kept above 240F degrees to keep the sulfur liquid.

The FORTRAN program was designed to take information from the customer such as pipe size, insulation type and size, min/max temperatures required, and other parameters and calculate potential solutions using the company's products. I worked closely with the head of research and the engineering department to refine and improve the accuracy of the calculations.

The company was moving their operations from Houston and the current programmer was a student who did not want to transfer to SWTS, so they needed someone immediately. I interviewed and was hired but I had to quit school so I could travel to Houston for training so that delayed my graduation till December 1971.

When I started, the company was using a Time-Shared computer service with a teletype connection and were paying by the hour for services. Moving to San Marcos meant that their costs would increase due to the long-distance phone charges required to connect to the Houston Time-Shared computer service.

After a few months on the job, I was tasked with recommending a purchase of a computer that we could install in our offices to run the program. After researching the market, the finalists were IBM, Hewlett-Packard, and Data General. We selected Hewlett Packard 2100 (see information below) which was purchased for about $80,000 and was housed in a single refrigerator size cabinet. Fortunately, it did not require special electrical or air conditioning, so it was housed in a storage room across the hall from my office.

The HP 2100 had 16K of core memory (RAM), two 2.5 MB 15" hard disks (one fixed and one removable), a punch tape reader (instead of cards), and a teletype for input/output (30 characters/second). Spools of paper tape were sent from HP which contained all of the HP supplied software including the FORTRAN program.

Even though the loading of the software and the general operation of the system was cumbersome, the system worked very well for the purpose that it was required to do.

After about 4 years, I realized that the next step for the company was to bring in a computer system for the business/accounting part of the operation. I received an offer to move to Austin to take a challenging technical position which I did because I was more interested in expanding my scientific programming than starting to work in the business area of the company.

The company found an excellent person to replace me. He was able to do both the engineering and the business programming for the company.

Sonar Signals Processing and HP

The new job was with a company that did contract computer work for the US government especially in the sonar area. I was assigned to assist with a research project designed to evaluate a new type of sonar system on a joint US/French research project. The project was classified as secret so I could not find any reference to it on Google.

The requirement for our system was to capture the signals generated by the sonar and other data (i.e. ship speed and heading). These signals were sent to an HP 2100 computer on the ship for storage and analysis during each sonar test run. This type of system is known as Real Time Data Acquisition and Control because the data is continuously sent to the computer and must be recorded before the next bit of data arrives. On this and other Data Acquisition systems the readings from the instruments can arrive fractions of a second apart so the system was very busy getting a data reading and storing it on the disk before the next piece of data arrived.

This type of software is vastly different from heat calculations that I did at the previous company. Data Acquisition systems are discussed in detail in the Addendum.

The same HP 2100 was the computer used and I was one of three programmers and one electrical engineer on the project. We developed a set of programs using FORTRAN and HP Assembler that stored the data to a disk and provided data analysis tools which would do some signal processing and graphing operations while the sonar test ship was still at sea.

Prior to this system, data was stored on magnetic tapes on the ship which then had to be taken to a large computer on shore for analysis. This process would delay the availability of the data for the research staff and so our system was the first system that would allow on ship processing of the sonar data.

Given the requirements of the system, we had to use a special programming technique called microcode to allow the HP 2100 to keep up with the stream of data coming in and do all the other functions required. When the system was not taking in data on a test run, it was used to make calculations and plot graphs which was very useful in determining the performance of the sonar and other equipment.

Data Control Systems and HP

Shortly after this project ended, I was approached about a job in Houston again with HP computes and Real Time Data Acquisition. I joined the company as a technical expert to assist another company who had contracted to develop software for my company's systems.

They had a variety of clients including pipeline control, electric utility control, and industrial plant control. These systems had sensors and controls in the field to allow operators at a central location the ability to monitor things like temperature, flow rate, pump and valve status, and other parameters. Operators could also use buttons on a console attached to the computer at a main control location to change equipment in the field like turning on a pump or closing a valve.

Air Quality Monitoring and HP

After about 3 years in Houston, we returned to Austin where I continued to work with HP and other small computers and an Air Quality Monitoring system developed by the company. In addition to the HP computers, the company had created their own computer from scratch especially for Air Quality Monitoring and I worked on it for a time. This small computer was developed for the Air Quality and other applications but was never sold to the general market, so I don't have any information on it.

This was the late 1970s and personal computers were beginning to appear. The company I worked for had just upgraded their large, central mainframe computer to a UNIVAC 1108 which used multiple terminals connected to it for programmers and scientists to use to run their programs.

A co-worker knew about the new Apple II and a spreadsheet program called VisiCalc that he thought would be perfect for a project he was working on at the time. Most people thought of the Apple II as a "toy" and not a serious, capable business machine.

The co-worker went to his VP to ask for the $3000 required to purchase the Apple II, VisiCalc and accessories. The VP turned him down saying that the company had a brand-new UNIVAC that could do anything so he would have to use that.

Actually, VisiCalc was a breakthrough program that did not run on anything but an Apple II at the time. That was my first realization that the market was about to change and that the word "personal" in Personal Computer was a revolutionary concept. Having a computer that you could command to do what you wanted was a game changing event.

Since I only wrote some small utility programs for the UNIVAC, I do not consider myself a mainframe programmer and have not included the UNIVAC in the descriptions of computers below.

Computer Retail and Apple II, IBM PC, and Kaypro

After about 10 years doing data acquisition systems, I reached another career crisis of feeling like I needed a break from the programming environment.

By some unexplainable reason, I was looking in the want ads and found an opening for a Computer Store Manager position. I don't know to this day what possessed me to answer the ad. I had no experience with retail, computer sales, management, advertising, purchasing, etc. but there I was in the interview.

The store had been in operation a year and had an Apple dealership. The silent partner bought out the active partner and hired me on a Friday to be the new store manager. I went to the store the next day to be "trained" by the active partner and I took over as the manager and only employee on the following Monday.

After Monday it was obvious that this was more than a one-person job, so Helen agreed to come with me each day to answer phones, track paperwork, and other administrative duties. We worked together the whole time we worked in the computer store (she did start making a salary after a while).

At that time the vast majority of customers had no real knowledge of the computer or what it could do so my approach to sales was to focus on helping the customer understand how the computer could be useful in helping them solve specific problems.

I started by asking why the person wanted a computer. What problem did they want the computer to help them solve? My goal was to be able in the span of 30-40 minutes to fine one major thing that would make the computer with purchasing and then doing a demo on the computer of a software package that would solve their problem.

If they told me that they needed to setup a company budget report then I would use a spreadsheet to create a simple budget from scratch while they watched me. I would show them how the computer would make calculations, run totals, etc. so they could see it in action.

This came to be known as "Consultative Selling" and properly applied there was no need to do any "selling" or "convincing" because the customer could see for themselves that it was valuable for them. The store backed the

initial sales demo up with classes and telephone help to get the customers to a functioning level as quickly as possible.

After a about 3 years, we both hit job burnout because the store continued to grow and the workload was pretty intense. We hired staff, knocked out walls, increased inventory, etc. so when we left the store had 3 locations and about 30 employees. I realized that I was spending all my time on the business of running a business and did not have time for teaching, sales, meeting customers, and other aspects of the operation that I enjoyed.

In 1981, IBM announced the IBM Personal Computer. We immediately saw an impact as customers wanting a "business PC" started buying IBM exclusively. The store owner attempted to get an IBM dealership but was denied. We also failed to get IBM clone dealerships for Compaq or others when they became available so that continued to hurt sales.

We knew we had to have more than Apple products, so we applied for a Kaypro CPM computer dealership. Kaypro had a "luggable computer" (think really heavy suitcase) and ran CPM operating system which was similar to the IBM system but not compatible with IBM operating system so PC software would not run on them.

The CPM computers had a wider variety of business software than Apple and the system was actually cheaper since the monitor and disk drives and some software was included. Eventually Kaypro did manufacture an IBM PC compatible computer with a 10MB hard drive but it was "too little too late" in the computer marketplace.

At this point I want to suggest that the Apple II, IBM PC, and CPM systems were very close to the capabilities of the IBM 1130 and the HP 2100 of the early 1970s in terms of their speed, amount of RAM, size of hard disk, etc. The major differences were the use of ICs, lower cost and smaller size. As personal computers, many of the tasks like word processing or spreadsheets were fast enough and users did not care because small computers were so much faster than any manual method they had been using.

Since personal computers incorporated sound, graphics, mouse, and other ease of use features, many people found that they preferred these machines to the larger, faster mainframe machines that they used at work. Mainframe software was also noted for the difficulty of learning and using most of their programs. My co-worker was just ahead of his time in his request for an Apple II.

During this time, I happened to be in an Austin bookstore and walked by one wall and noticed that all computer books. I was amazed to realize that 1) there were enough people in Austin who would be interested in those books and 2) the range of books from simple, introductory level to the very technical, very specific computer area.

Computers must have reached a kind of critical mass for a bookstore to stock so many titles for what I had though was a limited audience. Also during this time there were a number of Computer Clubs that met monthly to share information or the latest advance or gizmo.

Personal computers were moving from toy to indispensable roles in business and organizations.

Hewlett-Packard Computer Publication

After we left the computer store, I did some consulting which led to a job as the Managing Editor of a monthly computer newspaper/magazine. The newsletter was sent free to all HP Computer users that we could get on our mailing list. The publication was funded by advertising and was in a large newsprint format. They actually employed both Helen and I so we worked together again.

Here again I found myself in a new world of publishing and had to quickly learn the publishing and printing business and how to do professional reporting and editing. We used PC computers for all our administrative tasks, but the actual layout was done by a contract professional graphic designer.

PC Desktop Publishing was very new, and we did not have it so I had to send stories to the commercial printing company via modem. They printed galley proofs and sent them back by courier. The final layouts were done manually by cutting and pasting the galley articles into the mockup for the commercial printer.

The job lasted about 9 months because I realized that I really wanted to teach computers at the College level and I needed to have a Master's degree to qualify.

Back to Retail and Off to School

I knew I could go back into computer store sales and earn enough to be able to have a flexible schedule for my schoolwork. Helen and I managed to get rehired by our previous employer and this time we worked in the same store but not in the same positions.

Unfortunately, that company was unable to survive the changes in the marketplace and filed for bankruptcy a few months after we came back. I was able to get a sales/Assistant Manager job at another computer store and stayed there until I was able to finish my degree in Instructional Technology (basically studying how to apply all types of technology to student learning in the classroom).

Helen got a job at Austin Community College library during this time which helped put me through school for the once again.

ACC and Independent Computer Consultant

I applied at Austin Community College for a computer teaching position, but full-time positions were not available, so I accepted part-time teaching assignments each semester. For about 10 years, I taught part-time as many courses as I could year-round which generally was 3 courses per semester plus summer school.

To supplement my teaching, I started a consulting company for programming and computer support. I worked for a wide variety of clients but focused mainly on creating business systems using database or file system software. Some of the clients included a dress shop, an art gallery, a weekly political newsletter, several state-wide teacher associations, and an insurance company.

During this time the Internet "came of age" starting in 1983 several computer networks were linked together. I did understand that having a "Personal" computer was going to be revolutionary, I did not expect the Internet to be a factor in people's lives very quickly.

I assumed it would be very expensive to create all of the wiring to link then computers together. It was also evident that with a home/business connection of a phone modem to the Internet would be too slow for most things we have today such as downloading pictures, movies or music.

After this part-time setup, I was finally hired full-time at ACC and became a campus Computer Department Head and taught several courses. During that first year, I learned of a proposal for the creation of a new college-wide department dedicated to assisting all faculty in using technology and innovative methods in their teaching.

The department was called the Educational Catalyst Center (EC2) and I was interviewed and hired as the first (and only) Director starting the next school year. We created a faculty only computer lab at the main

administrative building and conducted a variety of training in the lab and on the various campuses around Austin.

We also held faculty retreats and technology fairs to allow faculty to describe how they were using technology or innovation. A great deal of discussion focused on the future direction of education including using technology, innovative teaching styles, and preparing students for the future. We also discussed how we might prepare ACC for that future.

During this time the Internet and WWW were starting to become important, so I taught myself webpage design and then taught a number of classes for faculty wanting to create their own instructional web pages.

After a couple of years of operation, there was a change of Presidents and the EC2 was disbanded, and I was offered the position of Head of Faculty Development. I served in that capacity for another 18 months.

ACC used PCs for administration and instruction for the most part with a few Apple computers in some situations. During this time, I was mostly a user and teacher of computers and not a programmer or developer.

I also decided I could no longer keep my skills current in PC and Mac, so I decided to focus on PC systems and applications along with developing websites.

Round Rock ISD

After being at ACC for 15 years, I decided it was time to look for another opportunity and applied for a job in the Administrative office of RRISD. A new function was being created under the Assistant Superintendent for Instruction and I was hired along with two other people to focus on Instructional Technology for the district.

Our mission was again to help teachers learn to use all kinds of technology in the classroom. In many ways it was similar to the Educational Catalyst Center at ACC.

By this time, personal computers were cheap enough and powerful enough to be in many homes, most educational institutions, and in the majority of workplaces. Most students had some computer skills (often more than their teacher) and so our focus was not introducing students to computers or multimedia but trying to enhance their learning using technology.

We explored various instructional software packages, campus to campus video streaming, alternate teaching/learning styles, and other ways to improve student learning.

Consulting and retiring

For the past 20 years or so, I have been a freelance computer consultant. I have worked alone and mostly out of my home office. Although I have done a lot of different jobs from one-on-one training to major projects, my primary expertise is in creating filing systems (databases) and websites which utilize online forms with a database.

I found clients by word of mouth and was happy that I could choose both clients and projects that were interesting for me and valuable for the client. I must confess that starting with the computer store manager position, I gained experience with business software and business uses of computers including accounting systems which has been very useful.

The following are examples of the systems and organizations that I have consulted for over the years:

- Dress shop
- Sewing machine store
- Art gallery
- Political publication
- Insurance company
- Movie poster company
- Several state-wide teacher associations
- Several church websites
- Religious publication archive
- Legal case tracking website

Since computer technology continues to develop very rapidly, I realized that I needed to focus on things that I was most interested in and had the most experience. In the early days, I felt pretty comfortable assisting people with all types of personal computer systems. This was especially true during the time I worked in the computer stores.

It is really amazing to me that I am still able to work for paying clients in the 50th year of my professional life. I no longer look for new work or spend time learning new systems or computer languages, so I guess that makes me “semi-retired”.

My career has been varied, challenging, and rewarding. I would have never imagined all the twists and turns that would happen over the years when I took those computer classes in 1970.

Helen have been very patient and supportive of all of the job changes and the times when I was on sales commission or consulting and did not have a steady income.

The agony and ecstasy of programming for me

This article helped me realized that I need to say something about my creative relationship to programming as an art form (https://www.wired.com/story/healing-power-javascript-code-programming/). Since this appeared in the Wired technology magazine, it is definitely heavy on some of the details, but I believe him when he talks about being "healed" by programming.

Life in general and human interaction specifically can be difficult, confusing, and frustrating and that is on a good day. Computers (like pets in a way) are consistent, predictable servants/companions. Computers are frustrating and inscrutable at times. They are both simple and complex even to those who try to immerse themselves in this world.

If I write lines of code, I know that I am the person who made the computer do what it was told to do. If I mess up, then it is my fault and generally the computer will tell me (not blaming me just telling it like it is).

In a way it is the same satisfaction that all artists and builders get when they can step back and say, "I created this". It is a tangible sense of accomplishment. For me there is also the satisfaction of knowing that I have helped other people make their job easier or faster or better than it was before.

There have been times when I experienced what athletes call "in the flow" where time is suspended, and the code seems to arrive on the screen effortlessly and elegantly. It is a mystical and magical happening that is

really hard to describe. Programming can at times be boring, tedious, mechanical, and not all that creative but still the end results are something of value.

For many it is an addiction and a substitute for the messy human social world.

May your computers always be nice to you. Remember someone was trying to make your life better when they wrote those lines of code.

Epilogue

I feel very lucky to have started my computer work when and where I did. I now realize how important it was for me to have started with computers in 1970. Here are some of the things that I think contributed to my computer experiences and the overall success or challenges of computers over the years:

- Transistor replacing vacuum tubes
- Integrated circuits (ICs) which put many transistors on a small chip very cheaply
- Integrated circuit memory replacing core memory
- Complete computer logic (CPU) put on an IC chip (Microprocessor)
- The increased speed and dropping price of IC components over time (Moore's Law)
- Steady increase in performance and capacity of computer accessories with regular price decreases (printers, disk drives, monitors, scanners, etc.)
- Apple, IBM and others making computers "personal"
- Standardization in software, communications, and peripheral devices
- Creation of Operating Systems that promoted software development by many small developers
- Development of tool software like word processing, spreadsheets, and databases which allowed users to customize the system to solve their specific problems without professional help
- Computer languages like BASIC, FORTRAN, HP assembler, Univac Athena machine language, C++, Pascal, HTML, CSS, FileMaker, dBase, Java, and ColdFusion (I have written programs and/or taught all of these languages)
- Developer software tools like databases, program editors, text search software, and text file compare software.
- Apple Lisa and Mac introduce the Graphical User Interface (i.e. Windows/Mouse/Fonts/Pictures)
- Computer retail stores to both sell and education people about personal computers,
- Computer training and use in all levels of education,
- Computer games and game consoles
- Creation of the Internet, World Wide Web, search engines, email, and eCommerce
- Smart phones and tablets which are electronic "Swiss Army Knives" capable of an amazing number of features and functions (of course all driven by the tiny computer inside)
- The "dark side" of computers including spam, hackers, stealing IDs, location tracking, and other privacy/security concerns.
- Satellites and GPS systems.
- Cell, Bluetooth, and WiFi wireless communications.
- Lasers in CD/DVDs, fiber optic cable, printers, and other devices.
- Google and Facebook becoming gigantic companies eclipsing companies like IBM, Ford, GE, and others.
- Cloud (large computer systems accessed through the Internet) for computing and storage.

- Internet of Things (IoT) with computers/WiFi in ordinary items like refrigerators, thermostats, video doorbell, and even toothbrushes (yes, I have one)

Remember that almost all of the things on this list did not exist 50 years ago. The speed of invention is not slowing down so it is hard to imagine what the future holds.

Some things have happened or are in the process of happening that I did not think would happen in my lifetime. I did not think that computers would be fast enough or "smart" enough or cheap enough to do these things until many years into the future.

Here are some things that I would not have guessed would have happened by 2020s if I had been asked 20-30 years ago:

- Automatic language translation either from text or speech
- Facial recognition and other forms of biometric identification
- Artificial Intelligence (AI) systems used for everyday and commercial purposes
- Speech recognition and speech output like Siri or Alexia
- Self-driving vehicles (completely autonomous vehicles may still be some way off)
- Robots which incorporate AI in manufacturing, agriculture, and other areas
- 3D printing of products and possibly inexpensive housing
- Electric vehicles becoming the "wave of the future" in transportation
- Solar panels and green energy sources
- Quantum and optical computers
- Virtual and augmented reality
- Drones for military applications and commercial package delivery
- Software that writes software without human programmers

These 50 years might compare to the first 50 years of airplane development. Starting with the Wright Brothers plane in 1903 that was barely airworthy to jet powered planes like the BOAC Comet which was the first jet passenger plane in 1952.

Of all the jobs I have had over the years I think I enjoy writing programs and teaching the most. I found programming to be very satisfying because there is a useful product at the end of the process. My goal in every project was to build a system that would meet or exceed the customer expectation, be easy to use, and would be easy to fix or enhance as needed.

Teaching afforded me to chance to pass along my passion and expertise to people who were just beginning their career.

I exited high school thinking I would be a Math teacher in a high school. My education up until my last couple of year was focused on that path. Something happened in 1970 that made all the difference and led to 50 years of interesting and challenging jobs and projects.

In 50 years, computers have become indispensable parts of our lives. Our society now has smartphones, huge computer data centers which form the "cloud", large systems which run Artificial Intelligence software, and tiny computers embedded in all sorts of everyday appliances.

In 2020, the world experienced a pandemic which instantly impacted how we live and work. Suddenly people had to use computers, Internet, Zoom, email, and other tools to work from home or attend school from home.

As difficult as it has been to adjust to this situation, we should be grateful that computers had matured to the point that we could make these shifts. If this had happened in 1970 we would have faced an even more challenging task of keeping ourselves healthy while continuing to function as a society.

TECHNICAL ADDENDUM

This section gives an overview of the computers that I worked on over the years. The most important thing to realize is that basically each of these systems have the same components and require programs (software) to function.

See the Hardware and Software descriptions in the next section to get a description of these basic, universal aspects of computers.

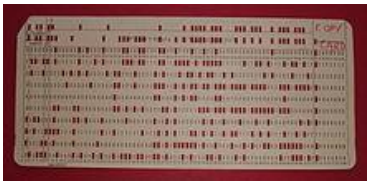
The IBM 1130 Computer System Description

The IBM 1130 was an entry level computer designed to be sold to smaller organizations and educational institutions. The installation at SWTS had the main console which had the computer and hard drive on the right side of the console cabinet, a card reader, and an 1132 line printer.

Introduced in 1965, the 1130 is considered an early third generation computer. It used punched cards for software programs and data entry, used Solid Logic Technology (transistors), included a 1-megabyte disk drive, and probably had 16,000 BYTES of RAM. The primary output device was the IBM 1132 printer which had a top speed of 80 lines per minute (it made a lot of noise but was pretty fast for the time). The 1130 was the first IBM computer to rent for less than $1,000 a month or could be purchased for $32,280.

IBM 1132 line printer


Keypunch machine for making Punched Cards (common computer input method)


Example Punched Card


IBM 1-megabyte disk cartridge (15”)

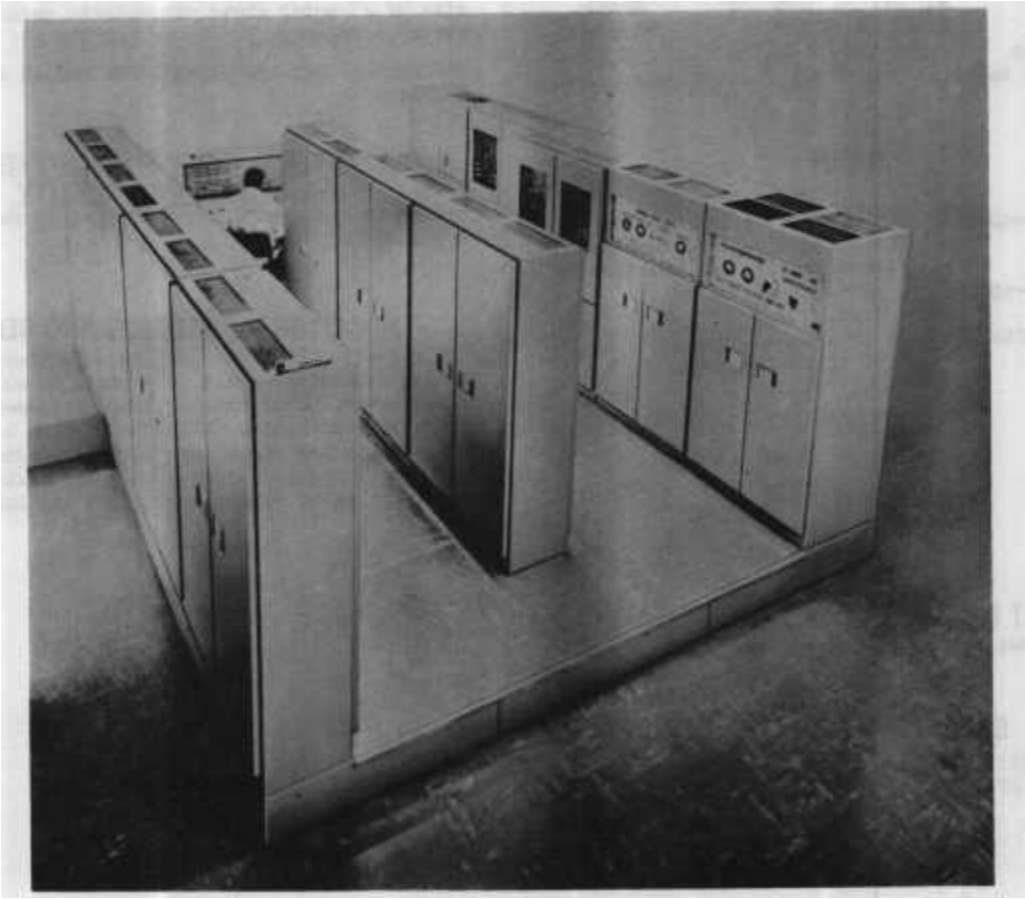
The Athena Guidance Computer System Description

The Athena Guidance Computer was given to SWTS by President Johnson when it was decommissioned by NASA. This computer was specially designed for missile launch and had to be modified after arriving at SWTS to make it function as a viable teaching tool.

Athena Operator Console

The Athena computer had **256 words of 24-bit core memory** to be used as a data work area and an **8192-word drum** for the storage of the program and data. The Athena was completed in 1957. It occupied 370 square feet

and weighed 21,000 pounds. It was found to have an average time between hardware failures of 48 days, twenty times better than the original specifications. The U.S. Air Force deployed the liquid fuel Titan as an interim measure pending the completion of the solid fuel Minuteman ICBM. In the late 1960s, the Air Force gave one of the original Athena computers to the electrical engineering department of Carnegie Mellon University. It was used for various class projects and later donated to the Smithsonian Institution.



Athena (note console in back and raised floor for AC cooling)

The Athena was the first all transistor computer produced in significant numbers (prior computers used vacuum tubes which were slower and less reliable). It was completed in 1957 and cost $1.8 million when it was new.

The following is an extended quotation from a paper titled **Computing History at Southwest Texas State University** by Dr Grady Early (https://digital.library.txstate.edu/bitstream/handle/10877/3813/fulltext.pdf?sequence=1&isAllowed=y).

In 1967, the Department of Mathematics acquired a Univac Athena, an obsolete guidance computer for the old Titan I missile system, from the Air Force through the auspices of President Lyndon B. Johnson. Although this Athena had originally been scheduled to go to a University in California, SWT President James H. (Jim) McCrocklin called the White House and asked President Johnson to send it to SWT instead. He did. The Athena was acquired for shipping costs and was assembled by Dr.Henry N. McEwen, Professor of Mathematics and Computer Science, and Mr. David E. (Dave) Hufferd, an electrical engineer hired by the department to maintain the Athena. In addition, Mr. Marcus M. Muirhead was hired to serve as half-time computer center director and half-time Computer Science faculty member. He served until September 1971, when Mr. John. I. Prewitt was hired to replace him.

The Athena was a huge (11 1/2 ton) computer which was very noisy because it had to have its own generator to produce 400 hz power and its own air conditioning unit to keep it cool enough to operate. [Ed It made a lot of noise when starting up and the generator was in a separate room] The Athena was assembled in the Laurel Hall computer center where it shared space with the 1401.

The Athena was a very interesting machine in its own right. It was the first large-scale, all solid-state computer ever built. It had limited use as a Titan. I guidance computer (1961-1962). But it was used by NASA at Vandenburg Air Force Base and at Cape Kennedy for ground guidance of the early CommSat Corporation communication satellites (Transit, Tyros, Echo). This service extended from 1962 until about 1970. The Athena did not actually communicate with the satellite. It guided the Thor booster to get the delivery vehicle free of the ground. When the booster stage broke away, on-board computers guided the satellite into orbit.

An interesting problem associated with the Athena concerned writing to the magnetic storage drum. The Athena had 256 24-bit words of core memory and 8K 17-bit words of drum memory. Programs were entered on the drum manually, then executed in Run mode. The drum could be read, but not written, at Run time. Unfortunately, this prohibited the writing of self-modifying code; i.e., creating a loop. Thus, one of Hufferd and Muirhead's first chores, to turn the Athena into a relatively practical academic computer, was to add four hardwired instructions to the instruction set to allow writes to the drum.

Another problem concerned output. The only human-readable output device was a 10-key Remington adding machine with mechanical fingers which were activated to strike the appropriate keys to produce output. Communication with the Athena really improved when Hufferd and Muirhead rigged the Athena to produce output on an ASR 35 Teletype [Ed output speed was 30 characters per second].

In 1968, academic computing was enhanced by a single terminal tied to the University of Texas Computation Center. Again, problems were encountered. Muirhead collaborated with the University of Texas Computation. Center staff to write the software that would allow our terminal to communicate over a telephone line to the CDC 6400 at the University of Texas. But occasionally, for no apparent reason, the communication software would crash. One day when a crash occurred, Muirhead, on impulse, picked up the phone and said, "Hello." After a pause, the operator said, meekly, "You sure do have some funny signals on the phone line." The operator, curious about the signals on the phone line, would patch in and say 'hello'. The 6400 thought the signal came from our terminal, but it could not interpret that signal, so it terminated the connection.

In the Fall semester, 1969, the Department of Mathematics rented an IBM 1130 minicomputer which could run Assembly Language and Fortran programs. In addition, the IBM 1130 was connected to the CDC 6400/6600 at the University of Texas so that more exotic, number-crunching programs could be run. Other problems arose. One night, William (Bill) Patton, a student operator, called Muirhead at home to report that the 1130 was down. Muirhead told Patton to call IBM for service, but Patton insisted that Muirhead come to the computer center. When he arrived, Muirhead discovered that someone had dropped a wad of bubble gum into a box of printer paper. The gun was transported, with the paper, into the printer. Muirhead and Patton worked long and hard to remove the gun from the printer. Alas, they were finally forced to call IBM to restore the printer to service.

Encouraged by McCrocklin, the Department of Mathematics offered its first Computer Science course (Fortran programing) in the Spring semester of 1968. This course in Fortran programing was designed by McEwen and taught by Muirhead- and Hufferd. Each student purchased a set of pre-punched cards designed especially for exercises in the textbook (a CDC card kit). For each exercise, the student would select the appropriate cards, assemble them into a program, and turn them in to the instructor who would then send all of these card decks to Dallas on the bus for processing by a computer there. Computer printouts would be returned in the same way. McEwen is certainly not the appropriate person to approach with complaints about turnaround time. (As an aside, it was noticed that printouts, sent first class, took two days to get to San Marcos. The card decks, sent bulk rate, only took one day. After requesting that printouts be returned with the cards, one day turnaround was achieved. IH 35 was an unusual but effective communication link.)

The Athena served the department well as a teaching tool for machine language programing and for computer organization until 1971. In 1972, it was placed on the surplus property list and sold to McEwen and Dr. Grady G. Early, Director of Computer Science. Parts of the Athena still serve as concrete models of core memory and drum memory devices. Incidentally, McEwen and Early were long in the good graces of an administration which was not completely sure that they had not acquired a Titan I missile to complement the Athena.

Computer Science as a minor for the Bachelor of Science degree was proposed in 1967, approved in 1968, and first appeared in the University catalog of 1969-1970. The curriculum consisted of seven courses: Math 2308, 2318, 2328, 3308, 3318, 3328, 3338, and required the calculus as a pre-requisite. This curriculum certainly reflected the then prevalent bias toward scientific computing.

For more information check https://en.wikipedia.org/wiki/ATHENA_computer and http://www.silogic.com/Athena/Athena.html.

Hewlett Packard 2100 System Description

*HP 2100 system including Computer, Paper Tape Reader, Dual Hard Disk, Tape Drives, Line Printer, and Teletype console [Ed **I am not in this picture**. Note the paper tape rewinder which is hanging off the table just to the right of the tape drives which was used to quickly rewind long spools of program paper tapes.]*

The Central Processing Unit (CPU) of the HP 2100A and nearly identical HP 2100S are constructed of discrete logic Integrated Circuits and are the last 2100-series minicomputers manufactured by HP that make use of core memory. The 2100A has a maximum memory capacity of 32K, 14 I/O channels, dual-channel DMA and

hardware multiply/divide. The introduction of the 2100A instantly made all previous HP minicomputers obsolete.

The HP 2100A entirely omits all register displays (except for the switch register), marking a turning point in the evolution of computers which spread across the entire industry through the 1970s. Without register displays, traditional machine language programming from the front panel became impossible and no longer necessary.

I remember a story from the HP sales representative in Houston who had to support systems sold in Texas and adjoining states. A system was delivered to a customer and the HP person was not able to be there when the system was initially turned on at the customer site.

To start the 2100 for the first time, the user had to enter a short “bootstrap” program by entering about 15 commands into the system using the front panel (the top row of buttons in the picture). The customer was a computer novice (obviously most people were back then) and so the HP person was giving instructions to the person over a long-distance phone call.

This required him to tell the person how to set the buttons by saying “on” or “off” in sequence. After the person had entered several numbers there was a pause and the customer asked the HP person, “Is that right to left or left to right?”.

The HP guy knew right away it was going to be a very long day! The customer was entering the buttons backwards!

Apple II System Description

The Apple II used a MOS 6502 chip for its central processing unit. It came with 4 KB RAM but could be extended up to 48 KB RAM. It included a BASIC interpreter and could support graphics and a color monitor or a TV. External storage was originally on cassette tape, but later Apple introduced an external floppy disk drive. Among the Apple II's most important features were its 8 expansion slots on the motherboard. These allowed hobbyists to add additional cards made by Apple and many other vendors who quickly sprung up. The expansion boards included floppy disk controllers, SCSI cards, video cards, and CP/M or PASCAL emulator cards.

For all its firsts, it had a major deficiency in that only supported uppercase letters. If you bought an Apple II in 1977, you could only type on it in capitals. It wasn't until 1983 and the Apple IIe that it had the ability to show lowercase too.

In 1979 Software Arts introduced the first computer spreadsheet, VisiCalc for the Apple II. This "killer application" was extremely popular and fostered extensive sales of the Apple II. The Apple II went through several improvements and upgrades. By 1984, when the Macintosh appeared, over 2 million Apple II computers had been sold.

Apple II computer, 2 Floppy Disk Drives, CRT Monitor

All the items to make the system work were sold separately including things like joysticks, printers, software, etc. At the time computer makers required retail sellers to be authorized dealers with a defined sales area (no mail order sales were permitted).

Since personal computers were very new, sales staff had to determine what the customer wanted to do with the system, what machine and accessories were required, and what software would need to be used. We offered free delivery and setup, individual or group classes on how to use the computer, repair, and other services.

The Apple II was the first personal computer many people ever saw. Its price was within the reach of many middle-class families, and a partnership with MECC helped make the Apple II popular in schools. By the end of 1980 Apple had already sold over 100,000 Apple IIs.

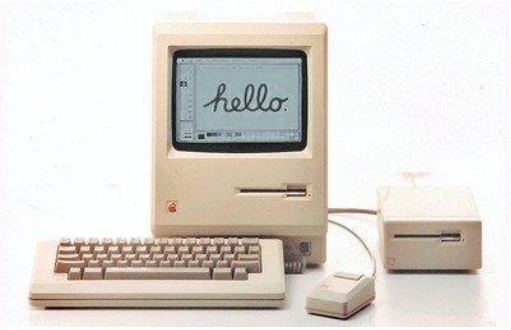
Initial Apple II used regular cassette players for programs and data. The Disk II single-sided floppy drive used 5.25-inch floppy disks; double-sided disks could be used, one side at a time. The disks stored 113.75 KB on one side of each disk. For business uses, the program was put in the A disk and the data was written to the B disk.

Complete Apple II system typically sold in retail stores had 8K or 16K of RAM and cost from $1,500 to $2,500 during the early 1980s. Apple continued to introduce new models like the II+, IIe, IIc and III that were all compatible with existing Apple systems.

Here is an interesting video overview of the Apple II and especially some of the competitors. It also talks about the importance of Apple sales to schools and shows some of the games that made it popular with kids. A*pple II Computer History and Review* https://www.youtube.com/watch?v=Mjd7pN57nU4

Apple Macintosh System Description

On January 24, 1984, former Apple CEO Steve Jobs introduced the first Macintosh at Apple's annual shareholder's meeting in Cupertino, California, debuting the new computer equipped with a 9-inch black and white display, an 8MHz Motorola 68000 processor, 128KB of RAM, a 3.5-inch floppy drive, and a price tag of $2,495.



Apple Macintosh

The Mac was a completely new approach to computers focusing on Graphical User Interface, ease of use, mouse control, fonts, windows that could be moved and sized, and many other features. The Mac was derived from the Apple Lisa which had all these features but with a $10,000 price tag.

Just like VisiCalc driving the sales of Apple II, the arrival in 1985 of the combination of the Mac, Apple's LaserWriter printer, and Mac-specific software like Boston Software's MacPublisher and Aldus PageMaker enabled users to design, preview, and print page layouts complete with text and graphics—an activity to become known as Desktop Publishing.

Initially the Mac was acknowledged to be underpowered (mostly lack of RAM memory), light on business applications and lacked a hard drive that business needed for fast data storage. However, the Mac was a great leap forward and the Mac sold itself to customers who could see how easy it was to use compared to the IBM PC and CPM systems at the time which still used typed commands and limited or no graphics.

The Mac has continually been enhanced and upgraded since 1984 and is the core of Apple's computer business today. The Mac would come to have most of the capabilities of the PCs including the ability to run PC software so businesses could have the benefit of both environments.

PCs got the Graphical User Interface and mouse when Microsoft introduced Windows 1.0 which closed the "ease of use" gap with Mac. Windows 1.0 was very limited when compared to the Mac which was designed at the outset to be totally graphical. Windows did not seriously reach comparable GUI status until Windows 3.0 was introduced in 1990 and Windows 95 in 1995.

Apple Lisa and Mac were derived from the Xerox Alto computer which was created in 1973 by Xerox Research department. This article covers the history and development of the Mac (https://en.wikipedia.org/wiki/Macintosh)

IBM PC System Description

The model IBM PC Model 5150 was introduced in August 1981. The system unit featured Intel 8008 CPU, working at speed 4.77 MHz, and an optional math co-coprocessor 8087. The RAM was 64 KB (the very first ones had only 16 KB), 256 KB max. (then later 640 KB max.) The ROM was 64 KB, included built-in language IBM BASIC (Special Microsoft BASIC-80 version). The keyboard was a full stroke 'clicky' 83 keys with 10 function keys and numeric keypad. The display was monochrome, working in text mode (40 or 80 char x 25 lines) or 2 CGA graphic modes: (320 x 200 and 640 x 200). The sound was a tone generator with built-in speaker. The I/O ports were five internal slots for enhancements and external connections, a monitor, parallel Centronics printer connector, and a cassette connector. The built-in media was one or two 160 KB 5.25" disk-drives. Three Operating Systems are provided—MS-DOS, CP/M-86, USCD Pascal.

Several factors have contributed to the 5150 PC's longevity are its flexible modular design, open technical standard making information needed to adapt, modify, and repair it readily available, use of few special nonstandard parts, and rugged high-standard IBM manufacturing, the last of which provided for exceptional long-term reliability and durability. Having the IBM logo was a deciding factor for many customers especially business users.


IBM PC with 2 floppy disks and dot matrix printer

Microsoft MS-DOS became the preferred Operating System so most software was written to run under DOS. Since Microsoft licensed DOS to IBM but retained the rights to sell to others, Compaq, Dell, and many other PC manufacturers were able to build and sell PC clones. Apple systems were proprietary and so they were able to prevent the development of Apple computer clones.

Initially the IBM PC was not an outstand technology leader and was considerably more expensive than Apple and other computer companies' products. In the business world, IBM had clout, so sales were made and software developers realized that they had to write their business software for the IBM if they were going to sell it in large numbers.

IBM initially underestimated the demand for the computer and that probably was part of the reason that our store was not given a dealership. In many ways, IBM was able to use the experience of Apple and others to develop the PC quickly and cheaply. Things like using existing commercially available computer chips, providing "slots" for new hardware to be added later, the operating system was similar to the CPM system used on other machines, etc.

When Microsoft introduced Windows 3.0 the PC caught up to the Mac in terms of the Graphical User Interface and mouse control. Both systems were comparable as far as software including Desktop Publishing, business applications, and normal office functions.

This is an interesting video about the development of the original IBM PC some of which I did not know:
The Original IBM PC 5150 - the story of the world's most influential computer
https://www.youtube.com/watch?v=0PceJO3CAGI

Kaypro System Description

Kaypro's first market-ready computer, in 1982, was called the Kaypro II. (The Kaypro 1 never made it to market.) It was a "portable (in name only) that ran CP/M on a 2.5 MHz Z80 CPU, with 64 Kb of RAM. It featured a fold-down keyboard and built-in monochrome CRT display and weighed 29 pounds. It also came bundled with a lot of expensive productivity software (Perfect Software's office suite of PerfectWriter, PerfectCalc, PerfectFiler, and PerfectSpeller) at a comparable $1,795 price.

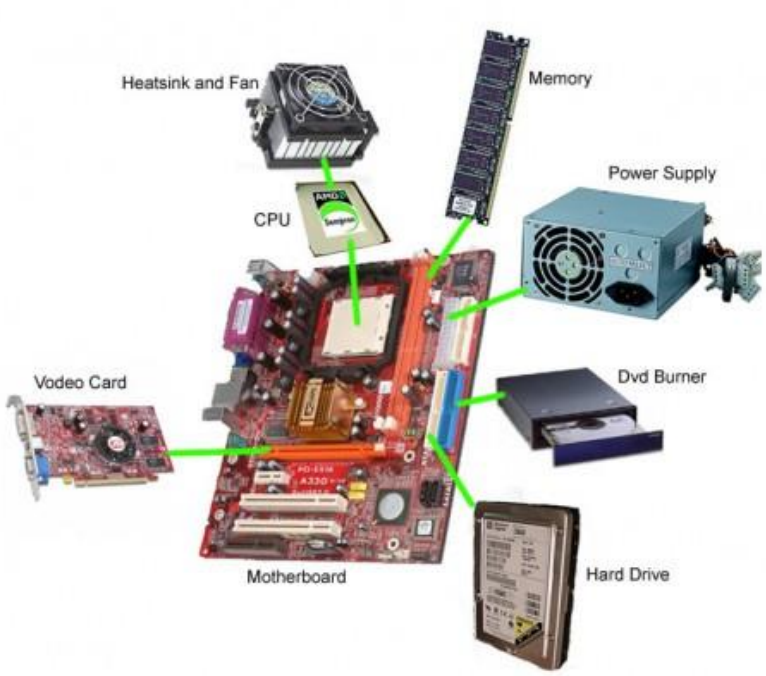
Kaypro CPM computer

In 1983, the company introduced the Kaypro IV, with increased-capacity double-sided/double-density floppy drives. The same year saw the release of the Kaypro 10, one of the first computers to feature hard disk storage – in this instance a whopping 10MB hard drive.

By 1985 CP/M machines had been roundly beaten by DOS-based PCs from IBM, Compaq, and others. Kaypro tried to jump on the IBM-compatible bandwagon with the Kaypro PC and Kaypro 286i models, but they were too little too late.

COMPUTER RELATED NOTES

Computer Hardware



All personal computers have the same basic physical elements (hardware) which work together to carry out the commands of the software. The most important ones are shown in this picture:

- Motherboard – a plastic board with a pattern of printed wires connecting the various places to plug in the other components
- Central Processing Unit (CPU) – microprocessor that does the work (math, logic, traffic cop, memory manager, and external device control)
- Heatsink/Fan – keeps the CPU from overheating
- RAM memory stick – working memory for software, data, calculated results, and other things are stored temporarily. Info erased when the power is turned off.
- Power Supply – converts AC to DC for the computer to have electrical power
- DVD Drive – laser disk which might hold software, data, video, pictures, and other info
- Hard Drive – file cabinet memory where information is while it is waiting to be accessed including the operating system, program software, files, and other information. Info stays when the power is turned off
- Video – turns commands from computer into signals for a monitor. Video cards have their own dedicated memory

At the most basic level, the CPU and RAM memory are just a huge number of on/off switches representing 1 or 0. Each switch is called a bit so a 64 bit computer has it's switches grouped together as 64 bit numbers. It takes one transistor to represent one bit so a 64 bit RAM chip would store one number in a set of 64 adjacent transistors.

The 64 bits are transferred between the components using a high-speed 64-bit wire "highway" which is designed to transfer one number at a time. There is a "clock" that tells the circuits when it is OK to transfer, calculate, and perform other operations. Every time the clock "ticks", all of components operate in sync. A modern PC can operate with a clock speed of over 4 Billion ticks per second.

The largest whole number (no fraction) that can be stored in a 64 bit computer memory location is 9,223,372,036,854,775,807.

Before transistors were invented, the vacuum tubes found in radios and TVs were used to as electronic switches. Compared to transistors, tubes are large, slow, took lots of power, very costly, and failed after only a few hours of operation.

Because of the advances in Integrated Circuits and other technologies like lasers, each of these components have steadily increased in speed, capacity, features, and at the same time the cost have either stayed the same or decreased.

Today Integrated Circuits have many different functions (calculation, logic, memory, Cell tower, WiFi and Bluetooth communications, and other things) all "printed" on a single, very small square of silicon. All of the transistors, wires and other pieces to do all these functions are shrunk down to an almost unimaginably small size.

Currently the IC wires connecting things inside the IC are about 5 nanometers thick. A nanometer is one-billionth of a meter. It's difficult to imagine just how small that is, so here are some examples:

- A sheet of paper is about 100,000 nanometers thick
- A strand of human DNA is 2.5 nanometers in diameter
- There are 25,400,000 nanometers in one inch
- A human hair is approximately 80,000- 100,000 nanometers wide

- A single gold atom is about a third of a nanometer in diameter

For more information on the history of computer hardware (https://en.wikipedia.org/wiki/History_of_computing_hardware)

Moore and His Law

All computers at the most basic level operate with transistors which act as little on/off switches. This are called bits and computer programs, numbers, pictures, videos, etc. are nothing more than a collection of lots and lots of these on/off switches.

The primary reason for the rapid development and spread of computers is a result of what is known as "Moore's Law" (https://en.wikipedia.org/wiki/Moore%27s_law). Gordon Moore observed in 1965 that the number of transistors in a dense integrated circuit (IC) doubles about every two years. This meant that the performance and capacity would continue to increase while the cost for a given amount of transistors would decrease dramatically.

This was an unbelievable prediction when it was made over 50 years ago, but it has. Even with the technical obstacles over the years, Moore's law seems like it will continue for the immediate future.

In 1970 the Intel 4004 computer contained 3,000 transistors on a single chip. In 2020 the AMD EPYC Rome computer chip contained 39.54 billion transistors and is used in Internet servers.

One of the challenges for humans to understand modern computers is that we don't have an intuitive understanding of very large and very small numbers. We can understand hundreds and thousands but it is hard to visualize a million or billion or trillion of anything.

Every aspect of computer technology today is light years better than the systems that were available in 1970. Every time the hardware chip technology got smaller, faster, cheaper, etc. the software and the systems could do more things and computers became easier to use and more varied in their applications.

The History of the Integrated Circuit (https://www.youtube.com/watch?v=SYSJefKc7L4) gives a brief overview on Moore and the birth of the IC industry.

Computer Programming

Today most people have some concept of telling computers what to do by "programming the computer". Basically, programming is about problem solving and then describing the solution in a way that the computer can understand and carry out the instructions.

The Programming Process generally follows these steps:

1. Defining the problem.
2. Planning the solution.
3. Coding the program (translating the solution into computer language).
4. Testing the program.
5. Documenting the program.

Since problems can be large or small or anything in between, solutions might be simple or complex and the number of commands (lines of code) would also vary accordingly. Almost every beginning programming course starts with a "Hello World" program. Just one line that sends that message to the screen or printer.

The really wonderful thing about software is that once it is written to solve a problem the program can be used for any type of problem that requires that solution. For example, a program written to calculate the area of a rectangle will work for situation where the width and height of the rectangle are known.

Programs can also make decisions based on inputs and calculations. For example, a program could monitor the temperature in a commercial candy cooker and turn on/off the heater as needed to keep the temperature at the desire level.

Another benefit of software is that it can be used to write programs which are general tools that can be adapted to a wide range of uses. For example, a spreadsheet helps users create budgets, estimates, bills, etc. A word processor can be used for academic papers, letters to mom, resumes, etc. The user has flexibility to use the tool in a variety of ways without having to have a programmer make a change for each situation.

Most commercial programs are written by teams of developers and programmers working on different parts of the software. Sort of like a team of people (carpenters, painters, plumbers, etc.) use their special skill to build a house, a team of software developers with specialized skill and experience would be used to build a large software program.

Computer/technology learning tips

Learning computers (or any technology) can be a challenging process for most students. As a teacher I always had my "how to learn computers" speech on the first day of class.

I told them that to be successful in the course students had to do the follow:

- Pay very close attention to everything the teacher says in lecture. Details and concepts really matter.
- Take notes that focus on the general ideas and terms. Mark things that are confusing and try to learn them before the next class.
- Learn the terminology of the subject (programming and computer applications). Technical fields give words very precise meanings so getting terms confused or not understanding them is a real hindrance to learning.
- Do not read the textbook like a novel. The textbook is a reference book and an alternate way of explaining a concept covered in class. Look at the examples. Try to enter the example in the lab.
- DO THE LAB assignments! Actually, writing programs or using a word processor is the best and only way to learn the skill of programming or word processing.
- When learning a computer program, write the steps you did to make the machine do some function or action. For example, CTRL-P or CMD-P brings up the Print dialog window. Hold down the CTRL/CMD like a shift key and hit P.
- Ask questions in lecture or lab. Do not waste time going over things that are confusing. Get help immediately from a teacher, fellow student, lab tech, your spouse, your 11 year old cousin, ...
- Do not expect to cram for a test. Learn a little bit every day. Practice a little bit every day (30 min/day is much better than 3 hours straight on the weekend).

As a student you goal is to learn small pieces and then how these pieces fit together. Like building a house brick by brick. It seems like slow progress but if you keep at it long enough you will end up with a really nice building.

Computer Operating System

An operating system is the most important software that runs on a computer. It manages the computer's memory and processes, as well as all of its software and hardware. It also allows you to communicate with the computer without knowing how to speak the computer's language. Without an operating system, a computer is useless.

Examples of Operating Systems are Microsoft Windows, macOS, CPM, Android (smartphones) and iOS (iPhone & iPad).

When a computer starts the first thing it does is to load and start the Operating System which will load other programs and show the user the “desktop” when it is ready work.

The OS acts as a traffic cop directing information to the screen, disk, printer, Internet, etc. and getting input from the mouse, pen, keyboard or other input device.

User programs such as word processors or spreadsheets have to be designed to work with specific OS systems and will not work on any other type of system. If software must run on multiple OS systems, then programmers must modify their software to make a compatible version for each target OS systems.

Computer Languages

Both FORTRAN and COBOL are considered “high level” languages since the commands are easier for humans to understand and write but these commands have to be “translated to computer machine language” before they can be executed by the actual computer. One line or command in FORTRAN would have to be translated in to many lines of machine language so the computer to run the program.

The IBM 1130 and the HP 2100 had FORTRAN compilers that read the FORTRAN statements and then created a computer readable version of those statements. When that was done then the computer executed the “FORTRAN code”.

All programming languages have similar features or statements such as PRINT (send something to the screen/printer) or READ (get something from the keyboard). Different languages use different words for these common features and so PRINT in one language might be OUTPUT or WRITE in another.

Also, all programming languages have a specific format or syntax for each command in the language. These are called keywords (think of them as verbs) must be spelled correctly and must only have valid parameters (nouns) which conform to the syntax of that language.

Learning a different computer language is usually fairly easy since all languages must have the same core actions, so you just need to find out what the equivalent of the PRINT command keyword is in the new language. Once you know that then you need to understand the syntax and options that are available for that command.

Here is a list of computer languages that I either used as a programmer or taught at ACC or both:

FORTRAN
HP Assembler
Sonar Processing Language (I helped write SPL)
BASIC (several versions on different PCs)
PASCAL
C & C++
Java
Forth
HTML
CSS
Javascript
DBase
FileMaker
SQL (several versions)

Excel Macro Language	XML
ColdFusion	AJAX
JQuery	HP/APPLE/MSDOS/CPM commands

For more information on FORTRAN see https://en.wikipedia.org/wiki/Fortran. Also note that FORTRAN is still in use and one survey showed it number 20 in the top 20 languages in 2021! (https://www.zdnet.com/article/this-old-programming-language-is-suddenly-getting-more-popular-again/)

For more information on COBOL see https://en.wikipedia.org/wiki/COBOL.

The Internet, Email and World Wide Web

The Internet is the name for the electronic highway which connects one computer to another through various communications methods. The Internet "came of age" when in 1983 several computer networks were linked together with a new communications standard called TCP/IP.

The early days of Internet was all text and no graphics. It reminded me a lot of the Operating Systems like MSDOS or CPM where users had to learn a strange set of commands to get the system to function. I knew that "normal people" would have difficulty learning the commands and would think that it was not worth their time.

The Internet is just computers talking to computers so as the computers got cheaper and faster then the Internet got cheaper and faster. Internet applications like Email, Chat, Newsgroups, and File transfers between systems were reason enough for some people to jump on the Internet bandwagon.

The next "big thing" was the creation of the World Wide Web in the late 1980s. It featured a language called HTML which allowed developers to control the look of a webpage by specifying things like fonts, pictures, and tables. In addition, webpages can have "links" which are text, buttons, or graphics which have the address of another webpage "underneath" them. When a user "clicks" on one of these links the system loads the new page from the designated location anywhere on the WWW.

The ability to link from one document to another without having the user type anything really opened up the web to many new applications such as online ordering, related document lookup, and complex webpages with extensive menus for navigating large websites.

The WWW expanded so quickly that finding relevant information became a serious problem. Several "search engines" were developed to scan the web sites and index the keywords for each web page. A user could then go to the search engine website and enter some keywords and get a list of webpages which matched.

These days we recognize that the Internet/WWW is both a wonderful tool and a threat to privacy and safety. In the recent COVID restricted environment, we have conveniences and advantages of online purchasing and working from home. At the same time, there is evidence that these benefits also have negative aspects also.

Real Time Data Acquisition and Control

Systems which have to interact with the environment (pipelines, garage doors, temperature gauges) are known as Data Acquisition and Control (DAC) systems. Special sensors outside the computer sample conditions such as speed, temperature, valve position, and other physical data. The signals from the sensors are convert to digital numbers that are fed into a computer.

Data arriving in the computer must be processed such as storing it on disk or displaying it on a screen or using it to calculate the oil flow rate and decide if a valve should be closed. These computers are dedicated to the task

and environment they were constructed to deal with. They may have special operating systems and normally do not do things like word processing or spreadsheets.

Most of the computers used in homes or offices are not DAC systems and their actions are not time sensitive. Generally, these computers only get input from the keyboard and mouse which are extremely slow compared to the speed of Data Acquisition input.

Many of these DAC systems not only receive data and store/display it but they use the data to carry out actions based on calculations based on the data. This control allows these computers to manage complex processes like oil refining and electric utilities much more accurately and efficiently than a human operator.

If this sampling and controlling has to happen quickly as things are changing quickly, then the systems are known as Real Time DAC systems. These systems must "juggle" different tasks in a time critical way (even small fractions of a second can be really important if a car is traveling at 60 MPH). The data must be updated quickly so that the control actions are able to correctly make changes needed to keep the vehicle or industrial plant operating smoothly.

Some Real Time systems are also safety critical systems such as the Apollo Guidance computer or a self-driving car. The systems I worked on were Real Time DAC systems but were not safety critical so if they failed there was no danger of human lives being lost.

Real Time systems are written as a group of modules which have a specific function (get the current speed or determine if an object is in the road). All of these modules have to share the same computer and so the Operating System has to be specially designed to "ping pong" between the different tasks as needed.

This type of software is difficult and complicated to write but even more difficult to test and debug. When the computer is taking data, it behaves like a juggler as events happened continuously and the overall operation of the computer is constantly changing.

When the software fails (or hits a bug), most of the time it just stops and finding what triggered the problem can be exceedingly difficult and hard to reproduce. In machines like the HP 2100, the limited memory and speed added to the difficulty of writing and testing the system.

As challenging as my systems were, systems like the Apollo Guidance Computer that landed people on the moon are much more difficult to develop because it had to perform in a novel environment and failure was not an option.

The YouTube below is a presentation about the Apollo 11 landing and the computer that controlled the spaceship in spite of some issues in the final minutes of the descent to the moon surface. The presentation is by a programmer who is employed to write real time code for Internet routers and has spent a lot of time researching the Apollo guidance computer.

The video is lengthy (1 hr 20 min) and is pretty technical both about the computer and the Apollo mission, but it is an excellent description of the moon landing, the challenges of creating the guidance computer, and the difficulty of writing the programs for that mission. The challenges of Apollo are the same ones that are being faced by programmers writing software for self-driving car and airplane autopilot computers.

Light Years Ahead | The 1969 Apollo Guidance Computer
(https://www.youtube.com/watch?v=B1J2RMorJXM)