BNL--52036

DE87 010641

BNL 52036
UC-13

FUTURE POSSIBILITIES
DATA, HARDWARE, SOFTWARE AND PEOPLE

Rear Admiral Grace Murray Hopper, USNR

Lecture Presented
on
September 19, 1985

BROOKHAVEN NATIONAL LABORATORY
ASSOCIATED UNIVERSITIES, INC.
UPTON, LONG ISLAND, NEW YORK 11973

UNDER CONTRACT NO. DE-AC02-76CH00016 WITH THE
UNITED STATES DEPARTMENT OF ENERGY

Our speaker for this afternoon is Rear Admiral Grace Murray Hopper, USNR and, at her request, I am cutting my introduction short to one sentence. Rear Admiral Hopper was the third Programmer on the first large-scale digital computer in the United States, and she has been coping with them ever since. The title of Rear Admiral Hopper's lecture this afternoon is Future Possibilities--Data, Hardware, Software and People.

There is a particular reason why I like this introduction to my talk. It is because it enables me to remind you that the first large-scale digital computer in the United States was a Navy computer, operated by a Navy crew during World War II. If you are wondering why I have kept my cap on, I have a good reason for this, too. I talk to large numbers of civilian audiences, and they are not always sure what I am. This cap is my identifier. You know that every record you put in a computer must have an identifier which has to be understood, both by the person who originally puts on the data, and also by the person who later looks at the record. Recently, I have been having trouble with my identifier: in fact, someone thought I was a security guard! I am wearing my cap so that you will know that I am proud to belong to the United States Navy.

I want to begin my talk with an analogy. Years ago on Sunday afternoons my family would go and sit on Riverside Drive along the Hudson River to watch the beautiful horses and carriages go by. Then the Drive was a dirt road: in a whole afternoon, we might see only one car pass by. The cars were enormously expensively, and individually built. There were no gas stations, so if you went on a trip, you bought five-gallon cans of gasoline, and took them them with you. If you broke down in the middle of nowhere, you telegraphed the manufacturer who sent out a man with a part to work on the car on the spot. Then along came Henry Ford with two novel concepts: standard interchangable parts, and an assembly line. He started to build inexpensive Model T's and soon many people owned cars, so changing the whole world of transportation. Paved roads, and gas stations stocking the interchangable parts appeared. People moved to the suburbs to live and shop, and they drove to work. But in all this we made one major mistake. We forgot to look at transportation as a whole. Because of this omission, we let the railroads decay, so that today, when we need them again, the roadbeds are falling apart and the flat cars have disappeared. The Army cannot move the tanks from inland to ports to ship them overseas; there are too few hopper cars to move both the coal and the grain crop. We have done a very poor job of managing transportation as a whole.

We are now at the start of the computer industry, an industry that will be the largest in the United States. The Model T's of the computer industry are here, and people are beginning to use them. I am very much afraid we will make the same mistakes over again. We will buy beautiful pieces of hardware and software and totally neglect the underlying force, the total flow of information through any organization, activity, or company. We should now be looking at the information flow, and then selecting the equipment that will best implement that flow. To do this, the very first thing we need to decide is which information is the most valuable and put our best equipment there.

For close to 18 years I have asked people how they value their information, and have gotten the finest assortment ever of blank stares. Indeed, some people even question that there is a difference in the value of information. To illustrate, consider a computer that receives information from marketing and controls the production of a chemical, operating the equipment by opening valves, channeling the raw materials through pipes, and finally telling inventory how much has been made. In addition, salaries are calculated and paid by the computer. Now suppose that two pieces of data simultaneously enter the information flow. One message comes from a valve in the plant, saying that if it is not opened the plant will blow up. There is a second in which to act, a hundred lives and a hundred million dollar chemical plant are at stake. At the same instant, a second message arrives that Joe had two hours of overtime. Which is the more valuable piece of information? And what are the criteria needed to make this choice? I have suggested three. The time we have to act on the information, the number of lives at stake, and the number of dollars at stake. So far, we have totally failed to look at the value of the data we are processing, yet it is absolutely essential that we begin to establish criteria for evaluating data, so that we can properly design the systems of the future.

I have used the word "system" a couple of times, because we tend to forget that we are dealing with a system. The raw material, data, is fed into a process as in any manufacturing process. The process consists of hardware, software, communications and people. Hopefully, the output product is information; this process is under some form of control. There is a feedback loop to the control to improve the quality of the information. Although we have talked endlessly about hardware, software, communications, and training of people, practically never have we looked at the raw material or at the output product. Also, we have failed to notice that information by itself is completely inert. It must be processed by a human being, who analyzes the information, thinks about it, compares and projects it, and turns it into knowledge or intelligence, on which we can base decisions. Since there is also a process involved, albeit a human processor, we need a control, and a feedback loop to improve the quality of the intelligence. To date, we have not worried very much about this person, but have swamped him with data, and given him no assistance in handling it. Artificial intelligence is going to be of the greatest importance here. We will require the expert routines. For example, consider the captain of a ship, with 200 computers on board, all sending information. He poses the question: how many gallons of oil do I have in the bunker? Now it is not sufficient just to tell him how many gallons of oil he has. The complete reply is, "you have x gallons of oil, you are sailing northwest into a northwest wind, and using y gallons an hour so that you can continue for z hours." This is where we need these expert routines, and computer users will need quite different training. Technical training alone, is not enough. In order to properly handle the present day tremendous load of information, computer users must know some history, some economics, and even some philosophy. The whole question of the value of information is an interesting one. Practically all large corporations insure their data bases against loss or damage or against their inability to gain access to them. Some day, on the corporate balance sheet, there will be an entry which reads, "Information"; for in most cases, the information is more valuable than the hardware which processes it. We have not recognized that fact. Also, we have not looked at the possible cost of incorrect information, assuming instead that everything that comes from a computer is correct! I made very good use of that failing once. I submitted a budget that was beautifully typed and set out. It was turned down. I took the same data and put it through the computer, and resubmitted the budget by computer. It went through with flying colors.

This whole problem concerned Major Rendleman, so he evaluated the possible cost of incorrect information in a Data Processing System (Rendleman, 1977). He found a section of the Privacy Law which states that government and military employees have a right to sue a federal agency if they are denied a raise or promotion because of incorrect information in a personnel file (the only time, I think, that an individual is directly given the right to sue an agency). The exact wording is, "Whenever any agency fails to maintain any record concerning any individual with such accuracy, relevance, timeliness, and completeness as is necessary to assure fairness in any determination relating to the qualifications, character, rights, opportunities of, or benefits to an individual, that may be made on the basis of such a record, and consequently a determination is made which is adverse to the individual, the individual may bring a civil action against the agency, and the District Court of the United States shall have jurisdiction in the matters under the Provisions of this sub-section." Rendleman calculated the cost of having such incorrect information on record. He supposed that there was a file with information on 8,000 people that was 95% correct. Therefore 5%, or 400, of the records contained incorrect information.

He discussed this with personnel officers and psychologists who suggested that in 90% of the cases the incorrect information would cause a negative decision to be taken on that person's prospects or promotion. So, in 360 cases, a negative decision was taken on incorrect information. Few people are aware of this law: those who are aware, might not have nerve enough to sue. By the laws of probability, we can assume that the chances of their taking action are 50-50, therefore 180 people will sue the Government, and they would win their suit. Suppose that the damage is placed at $2,000, together with $100 for Court costs and $650 for lawyers fees; then each case would amount to $2,750. For the total file, the possible loss is $495,000. We have never looked at the cost of incorrect information in our data-processing system, and there has been no follow-through of Rendleman's paper.

I thought you might be interested to know that the first computer bug is still in existence. During the summer of 1945 we were building Mark II for ordnance computations. We were working in a World War I temporary building, with no air conditioning and broken screens. Mark II suddenly stopped. The operator finally located the failing relay, where, beaten to death by the relay contacts was a moth. With a pair of tweezers, he very carefully removed the moth, put it in the logbook covered it with scotch tape, and below it he wrote "First actual bug found." The bug is still in the logbook in the museum at the Naval Surface Weapons Center, Dahlgren, Virginia. If you ever drive by Dahlgren, you can stop and look it.

In 1946 the war was over, and each one of us had to decide what we were going to do next. Although the Navy was the last to accept women, we were always part of the service, we were Reserves, never an auxiliary. At the end of the war we were given the opportunity to request transfer to the Regular Navy. Naturally, I applied for transfer but was turned down because I was too old (the cut-off age was 38, and I was 40). I elected to remain in the Reserves. In those days, we had three jobs to do--summer training duty, weekly meetings, and take correspondence courses according to our designators. I was designated as Ordnance, so I learned about big guns and gun turretts and so forth, until I ran out of ordnance courses. The only other courses for which I could gain credit were were the War College courses. Absolutely terrified, I sent for the first War College course and they sent me the first problem: I was to fuel a Task Force at sea in minimum time. The only information given was how fast the different classes of ships could pump and receive oil. So I lined up an oiler and a carrier and started pumping. Clearly, that was not going to get the task done in minimum time. I looked more closely and found I could simultaneously pump oil to a carrier and then to a destroyer, and both could be refueled because the rates at which they could receive oil were different. Earlier I had taken a course in problem-solving, and knew that I must always extend every solution. So I started pumping oil from the destroyer to a corvette. But that same course also taught me that I must generalize every solution. Therefore, on the other side of the oiler, I lined up a cruiser, a destroyer and a corvette--and ended up with half the Task Force, all hitched up, sailing down the middle of the ocean. My answer was returned with the comment, "An interesting solution!"

My next problem was how to scout the Caribbean in minimum time with a squadron of submarines. I called upon my friendly computer to help me and used a Random Walk Program for each of the submarines. I covered that map in minimum time, but had those submarines cutting across each other, making "U" turns and little circles here and there, -an interesting solution! Then along came the third problem, the one I want you to remember. I was to make a plan to capture an island; then I was to review my plan in the light of all possible enemy actions. Then I was to review the cost of not carrying out the plan. Again and again, I find that cost of not doing something is left out of our calculations. Therefore, we assessed the cost in men and material, of taking an island, then we looked at the consequences of bypassing that island. It is amazing how many times the cost of not doing something is greater than the cost of doing it. This factor must be an important consideration in every plan. I find people making their plans for computers based on the equipment they have in-house, their present requirements, and failing to review their plans in the light of their future needs, and of the equipment that then will be available.

Probably the most dangerous phrase you can ever use in the world of computers is "but we've always done it that way." That is a forbidden phrase in my office. To counteract it, I keep a clock which operates counterclockwise. The first day that people see it, they cannot tell the time. By the next day, they discover that what used to be five of, now is five after. They can tell the time again! It is not until the third day that they realized that there was never any reason why clocks should run clockwise. I will promise you something. If, during the next twelve months, anyone of you says, "but we've always done it that way" I will instantly materialize beside you and haunt you for 24 hours to see whether I can get you to take a second look.

Another major change we must make is to learn to listen to our juniors. Recently I gave a presentation to the Electronic Data Processing Committee of the Joint Chiefs of Staff, all of them Admirals and Generals. I reminded those gentlemen that they had tremendously thick reports to review and understand, upon which their major decisions would be made. They did not have time to keep up with the rapidly advancing technology; therefore, they were going to have to learn to listen to their juniors. I watched it happen in the Navy, with the greatest of glee. One young Lieutenant (Junior Grade) was ordered to a very small ship, that was deemed too small to have a computer for administration. So the LTJG took his own computer aboard into which he very quickly entered all the files. He did an absolutely marvelous job. When he was transferred, the Captain had to buy his computer, because the ship would not run without it. There was also the case of the young Commander who was told to take his Squadron out to an aircraft carrier. He found that the maintenance records of his planes had to be left ashore in the local Naval Air Rework Facility. But he wanted those records with him. So he bought an Apple Computer, "liberated" his maintenance records and entered them into his Apple. He used the computer to update the maintenance records, a venture that was highly successful. Finally he told the Department of Defence Computer Institute about it. Somebody said, "Are you supposed to do that?" He replied "I didn't ask." That story brings me to the most important piece of advice I could give anyone. "If it's a good idea and that, in my case, means if the idea is for the good of the ships at sea, go ahead and do it. It is much easier to apologize than it is to get permission." You must go ahead and do things. You cannot wait because there is too great an overload of micromanagement. Incidentally, four years later, Commanders of squadrons are supplied with microcomputers to keep track of the maintenance of their planes. But this change took four years to become effective, and that is too long. We cannot afford such delays: we must move into the future. If you are a little bit worried about taking such chances, let me loan you a motto, "A ship in port is safe, but that is not what ships are built for." I want you all to be good ships and sail out to sea and accomplish new things.

I can give you another excellent example. A young Electronics Technician First Class out in the Pacific fleet built a computer aboard ship. His public relations officer was very impressed, so he took a picture of the sailor and his computer and put it in the Navy Times. The article was seen by Rear Admiral Peter Cullins who wrote a congratulatory letter to the sailor. Knowing how long it would take a Navy letter to get from Washington to a ship at sea, he sent it directly by U.S. Mail. Well, the sailor decided if the Admiral could write to him directly, he could also write to the Admiral directly!. He told the Admiral in complete detail exactly what was wrong with the computers in the Pacific fleet and what he though ought to be done about it (it was an excellent survey). As a result, the Admiral ordered him to NARDAC, Norfolk and assigned to him three electronic technicians, three data processors and a small amount of money and told him to build the computer. In four months, they put together a beautiful computer from off-shelf components, and established themselves as the Micro-Computer Evaluation Group under the Naval Regional Data Center, Norfolk.

Part of their job, they decided, should be to put on a seminar to tell the Navy about microcomputers. You may not know what it takes to put on a seminar in the Navy: one must go all the way to the top to get permission, come back down again, and that can take up to a year. They ignored all of that, and with their own funds, hired a local hall. They called all the manufacturers of microcomputers, and said, "For $200, you can exhibit your equipment." They thought they might get 100 people, but had to cut off registration at 500, for lack of space. A year later, they decided to educate the West Coast, obtained space in a Naval Post-Graduate School and held the seminar again. This time, 800 people attended. A year later, 1250 people registered in Virginia Beach. This spring there were 2000 people, not only from the Navy, but from all the Services and other agencies of the Government. A year ago, the third time the seminar was held, it became legitimate, with the Admiral introducing the proceedings.

Despite the success of the seminars only a small number of people had been reached and it was felt that everybody in the Navy should know about microcomputers. It was decided to start a magazine. If it is rough getting permission to put on a seminar, try starting a magazine! One has to go all the way to the top of the Navy, and the Department of Defense, and the Office of Personnel Management, and back down again. Again, they ignored all these steps, but did one thing legally, getting permission from NABISCO to entitle their magazine *Chips Ahoy*. This magazine is probably about the best one on microcomputers. New hardware and the new software are reviewed each issue and there is an exchange library of programs. Finally, in January 1986, three years after the first issue the Secretary of the Navy determined that this publication is necessary in the transaction of business required by law in the Department of the Navy. Funds have been approved by the Navy Publications and Printing Policy Committee. But the magazine would not have existed if they hadn't <u>gone</u> <u>ahead</u> <u>and</u> <u>done</u> <u>it</u>.

About this time, the Navy and Air Force signed their first contract with Zenith Computers for 10,000 units and there are now over 50,000 of them. It became perfectly clear that there would have to be courses on how to use them. So the group decided to plan a curriculum. This is their curriculum brochure, and they've got just about everything anybody would want in micro-computers. The courses are open to anybody in the Government. They have introduction to Micro-Computer, MSDOS/ZDOS, MSDOS, Micro-Communications, Word Star, Multi-Mate, Speech Text, Super-Calc, Graph-Talk, Condor-III, Lotus-123, d-Base II User, d-Base-II Programmer, an updated version has d-Base III, C Language, Unix for Managers and Unix for Programmers. Everything you might need is there, and the courses run from one to four days. If there are enough people, they will come to you and present it (if you pay). Otherwise, they are held in Norfolk. The NARDAC is industrially funded so that they had to earn their way, and sell their products. Their sales brochure is printed in black and white, a disappointment to them, because there was not enough money to print in color, but the style is such that it could have been written on Madison Avenue. "Step into the future with NARDAC-Norfolk: Micro-Computer Services, with your future in mind." Everything is there to start you off; micro-explorations, consultation, technical assistance, software development, training, new items and conferences. Other questions are coming up at the same time, one being the question of the next higher-level languages such as Focus and Ramis, Muapper and Nomad. NAVDAC made a slight error in studying them, and gave one of the fourth-level languages to each of NARDACS. Now each NARDAC thinks that theirs is the best high-level language. Somehow, we must reconcile all those NARDACS. But meantime, they published the pamphlet, "Effective Use of Non-Procedural Languages," collecting all the information and advising people how to use it.

Meantime, we began to have trouble with our Seniors. The captain of one ship wrote to the Admiral saying "Do something. They're all over my ship." It sounded as if he had been overrun by cockroaches and rats but what had happened was that he been invaded by microcomputers. So Chief Slater wrote a paper, Everything You Always Wanted To Know About Micro-Computers, But Didn't Know Who To Ask--An Informal, But Detailed Look Into Current Micro-Computer Technology, Standard Pitfalls And Recommendations. He wrote it in plain English so that non-specialists and our Seniors can read it. We often fail to write about what we are doing in such a way that our Seniors can understand. Many times that is why we get "No" for an answer. We must make a much greater effort to produce material that is readily understood by everyone. Slater's paper was circulated all over the Navy and the Association of Computing Machinery and other organizations because people could read its plain English. There is a very interesting end to my story. A year ago, it was my privilege to swear in Chief Slater as an Ensign in the United States Navy. Although he was offered double an Ensign's pay elsewhere, he elected to stay because the Navy had given him the opportunity to do the things he wanted. And he had always wanted to be a Naval officer. Rear Admiral Peter Cullins was present at his commissioning, the Admiral who had written the original letter.

Something very interesting and important was at work there, something that often we forget. The second thing I learned at midshipmen's school was that my primary job would be to provide leadership, and that leadership was a two-way street, loyalty up and loyalty down. This means having respect for your superior, keeping him informed, and making suggestions. For the leader, loyalty is in taking care of your crew. There is something else we should notice in this story; one young sailor can turn the whole Navy around, if he goes ahead and does things. Today, Ensign Slater is at Alameda, in charge of the new Information Center. About six weeks ago he called me feeling very low, and saying that one cannot get as much done when one is an officer as when one is an enlisted man. I replied "You know what to do. Get yourself a good Chief." I think I heard him laugh all the way from Alameda to Washington, for he knew exactly what I meant. We forget how very much one person can do, even in very large organizations like the United States Navy. We must look to our youngsters to forge ahead and do things.

Some things bother me about the future. When I first met Mark I, on the 2nd of July, 1944, it was 51 feet long, 8 feet high, 8 feet deep, standing in a magnificent glass case inside which one could walk around. Mark I had 72 words of storage, each word consisting of 23 decimal digits and an algebraic sign, and it was capable of three additions every single second (actually one every 300 milliseconds). That sounds utterly pitiful today, but we always have to remember that Mark I was the first machine finished to assist the power of man's brain, instead of the strength of his arms. We didn't stop there. A lot of computers were build during the war. The Navy built Zephyr, Hurricane, Typhoon and Whirlwind: the latter, started at MIT, was the first real-time computer. It was one of those tragic occasions when the Navy ran out of money before it was finished, so the Air Force completed it. Whirlwind became the prototype for the Sage computers, the big Air Defense Computers, those huge concrete blockhouses just filled with millions of dollars-worth of computers. Do you know what the Air Force recently did with the Sage Computers? They took them apart, little piece by little piece, because the gold in them was worth more than they originally paid for the computers. (I have not yet found out what happened to the gold.)

It was not until World War II was over, that you could buy a commercial electronic computer. The first, called UNIVAC I, was completed in 1951, and the first four models went to the Census, the Army, the Navy and the Air Force. UNIVAC I could do an addition in 282 microseconds, (282 millionths of a second), so we were now a thousand times faster. But we did not stop there. In 1964, out came the first of the CDC 6400's, that did an addition in 300 nanoseconds or 300 billionths of a second, another thousand time faster. If you're a nut about the future, as I am, you will want to write the next line. Will we need an XYZ System that adds in 300 picoseconds--300 trillionths of second? Another thousand times faster? The answer is yes, we need that system now, but it cannot be built of present equipment. First of all, why do we need it? I can think of two major applications. World population is increasing, and food supplies must be increased. The biggest form of assistance in increasing food production would be to have better long-term weather forecasts. Today, there is no computer which will run a full-scale model of the atmosphere and the oceans. We can only run over limited areas for limited periods of time. Another problem is the question of water. I first encountered it two years ago. My sister lives in Northern New Jersey. They had a water shortage, and each person was limited to 50 gallons of water a day. Water began to run short in the wells in Norfork, Virginia. So the authorities decided to drill a couple more wells in the corner of the Naval base. But the two new wells actually were in Suffolk County. Suffolk County sued Norfolk County for stealing their ground water.

In Florida, so much water was drawn out of the underlying aquifer that huge sinkholes were created and houses and cars fell into them. In Tucson, so much water was drawn from the aquifer that the whole city is sinking: periodically cracks open up under buildings. In Colorado, the eastern half of the state is dry, the western half has water. Naturally, the people in the east think it is a good idea to obtain water from the Western half by driving a tunnel through the Rockies. The people in the West disagree strongly. Currently, there are battles in courts in Colorado, California, Nevada, New Mexico, and West Texas concerning who can draw how much water out of which river and which aquifer. Water supply must be managed. There is an even more threatening example. The eight Great Lakes states and two provinces of Canada have formed a consortium to manage the Great Lakes water, and they plan to sell water to people in the Southwest. States will be pitched against state for their water supply. Can you imagine the engineering job that will have to be undertaken? The pipelines, the reservoirs, and then the computer system, that will insure that every resident of the United States has an adequate supply of pure water? The agencies involved in ensuring our water supply are not glamorous, the Forest Service, the Reclamation Service, the Corps of Engineers and the Geological Survey. Unfortunately they do not make newspaper headlines, so that people do not realize that we have a major problem that is much closer than we think.

We are going to need computer power, and we will have some trouble getting the power we require. To explain that, I shall have to tell you something about myself. I am an extraordinarily annoying employee because I won't do anything until I fully understand what is required. I ask many questions to get a clearer picture. I was told that the first computer added in milliseconds (thousandths of a second). I could see a second on a clock, but not a thousandth part of it, so I said, "Show me a millisecond." But nobody would show me a millisecond, and I was stuck. Some years later, somebody looked at one of my programs, and said, "Hey, you wasted five microseconds." I said, "So what. What's a microsecond." He told me it was a millionth of a second. In the first place, I didn't know what a million of anything was (the biggest check I had ever seen was less than a thousand dollars). Soon afterwards people in the Engineering Building were talking about nanoseconds (billionths of a second). If one does not know what a billion is, how on earth does one know what a billionth is?

Finally, one morning, in total desperation, I called the Engineering Building, and I said, "Please cut off a nanosecond, and send it over to me." And I have brought you some nanoseconds today. When I asked for a nanosecond I wanted a piece of wire, which would represent the maximum distance that electricity or light could travel in a billionth of a second. Now, it would not travel through wire, but out in space at the velocity of light. So, if you start with the velocity of light and use your friendly computer, you will discover that a nanosecond is 11.78 inches long--the maximum limiting distance that light or electricity can travel in a billionth of a second. I was very happy with my nanosecond, and I looked at it from all angles, and thought about it. Then, at the end of a week, I called back and I said, "I need something I can compare this to. Could I please have a microsecond." I cannot give you each a microsecond, because I have only one: it measures 984 feet. I think we should hang one over every programmer's desk, or maybe around their necks, so they know exactly what they are throwing away, when they throw away a microsecond.

If you have a handy nanosecond, it gives you a feel for what's happening. The best way to make picoseconds is to take a big pepper mill, and grid picoseconds all over the table: one thousand equals a nanosecond. There lies the root of my problem. I no longer have sufficient distance to get two quantities from memory, add them both together and put the answer back, since I am close to the velocity of light. Einstein showed that when matter attains the velocity of light, it turns into energy. What can we do when we are up against the constant of the universe? Maybe we can turn to history. In the early days of this country, when people wished to move heavy objects they used oxen. If one ox could not do the job they did not try to grow a bigger ox, but used two oxen. Their approach tells us something. When we need greater computer power, the answer is not to get a bigger computer, but to get another computer, as common sense would have told us. We must build systems of computers and operate them in parallel. In data processing there was never any reason for putting Inventory and Payroll in the same computer: we did it because we could only afford one computer. That is no longer true.

We must start to think in terms of systems of computers. The biggest such system that I have seen so far is at NASA-Goddard, consisting of 128 by 128 processors, or 16,384 processors all in one system. Each processor talks to its four neighbors. The system is used to analyze the LANDSAT data, for example, to find a change of color of the desert, denoting that there might be oil below. It can follow the movement of a swarm of locusts across a crop. The direction of the future lies in building systems of computers. We must learn to become ingenious, and start again to ask why. I find many people working very hard, using a variable-length record. When I ask them why they are using variable-length records they always look at me blankly. Of course, the reason we did so initially was because of tape time, the input/output time. The rationale was to compact information on the tape, so that there would be less tape to read. Variable-length items moved tape faster. Now we use discs that are fixed length anyhow. If we went back to fixed length-fixed format records, and you wanted to find the payroll records of men over a certain age, you could go quickly straight to those records, as a matrix, without having to pull every individual record. We have forgotten that we used variable-length records to save tape, not to solve a problem.

Another one of my favorite questions is, why do we have an operating system? People respond that they need it to handle the input/output. No, you do not. You could put a micro- in front of a main-frame and have it contain all the protocols. One could answer long-distance, one deal with local lines, another with lines within the system, and you could have a back end that will take data and send it to any other computer. "Oh," you might say "I've got to have virtual storage." No, you do not. You buy the right size computer for that particular segment of the problem. We forget that today's micro-computers are more powerful than mainframes were 15 years ago. Then, a mainframe with 64K was a big computer. Now, even for a fairly large data processing application, one Micro is adequate, so we can build systems of computers.

We must concern ourselves with the protection of data. The issue rests upon the fact that we have tremendous amounts of data to handle, that is a major problem in every installation. The story is grim and sad. In 1972 Bell Labs first proposed using a backend computer to manage the data for a large mainframe. In 1974 they published a paper, Back-End Computer For Data-Base Management, discussing their results. Nobody paid any attention. In 1977, the Army Research Group at Georgia Tech used a back-end computer to manage the data for a large mainframe, and finally proved it goes faster and costs less. Yet it was not until 1982, ten years after the idea was first proposed, that we put our first data-base machine in the Navy. As we move to the future, clearly we must have data-base machines. Now, a data-base machine is not a computer: it does not multiply nor add. It is a little black box full of chips and it is a relational data-base. Furthermore, if there is a data-base management system on a mainframe, only that particular mainframe can gain access to it. If there is a database machine, more than one computer can have access to the data. And that leads us to the fact that we can build not only systems of computers, but systems of data-bases. When we recognize the concept of multiple data bases protecting certain information we can put classified data on one system, and open data on another. I think the data-base machine is the first of the specialized pieces of hardware of the future. I can envision an input device, a PBX with a computer, to take care of the protocols, a backend computer. I can see a graphics machine producing all sorts of beautiful graphics. Probably one would need an intelligent switch as some of the telephone systems have, to enable these computers to communicate with each other. But I shall need something even better to protect the data in an emergency. If an A-bomb was detonated about 250 miles up above the center of the country, it would emit an electromagnetic pulse that would take out all the telephone lines, all the power lines, and all the computers. At present, our protection consists only of lead surrounds and concrete walls. The real answer is the optical computer.

The first attempts in this field were made at MIT, where the optical match of the transistor was developed. We are beginning to get optical cables. They not only carry more, they cannot be listened to and they cannot be tapped; any attempt to do so causes the signal to disappear. There are cables connecting Boston, New York, Philadelphia-Washington, Chicago-Washington, etc. and AT&T is now installing optical cable under the Atlantic. Optical computers and cables will prevent damage from the electromagnetic pulse.

There is an exciting world up ahead, and I suggest that you go ahead and do things and keep asking why. But there is a slight danger, of which you must be aware. At any given moment, there is always a line representing what your boss will believe. If you step over it, you will not get your budget. Go as close to that line as you can. Meantime continue to educate your boss, so the line moves further out and next time you can go a little further into the future. I will show you how easy it to step over that line. We created mathematical compilers, then I found there were a very large number of people who hate symbols and did not like mathematics. They liked words. Therefore I decided to write my program in English statements, and build a complier that would translate words into machine code. I was promptly told that I could not do that, because computers do not understand words. I allowed that I never expected any computer anywhere to understand anything. All I planned was to compare bit patterns. In my boss's eyes that was a no-go idea, so I finally decided that the only way to get it done was to go ahead and do it. We built a pilot model of a compiler which accepts English statement. Then we wrote another request for a budget and on the end of it, put a statement inviting management to come and see a demonstration of the computer in action, we would feed in an English language program and get out a machine-coded program. Our compiler was an ancestor of COBOL. But the more we looked at it, the less satisfactory it seemed, since we had asked for the biggest budget ever. We decided to do something more. We wrote a program that went into the compiler, located all the English words and told us where they were. We replaced them with French words. Well, if you do something once, it's an accident. Twice is a coincidence. But if you can do it three times, you have uncovered a natural law. So we changed the words again. The third program was the impressive one, in German.

It was absolutely obvious to management that a respectable American computer in Philadelphia, could not possibly understand French or German. I had to spend the next four months saying that we would not think of doing such a thing, and were only going to program it in English. We had stepped over that line of what management could believe. Of course, COBOL exists, and in all the major languages in the world. Japanese is probably the most startling you will see. The verbs and connecting words of COBOL are known around the world. I made good use of that once when I was left behind in a Japanese installation where no one spoke English, and wanted to get back to my hotel. I finally remembered COBOL, found a COBOL programmer with whom I could communicate, and found my hotel.

Of course, that power of communication did not happen overnight. But pretty soon IBM built the Commercial Translator, and Honeywell came up with FACT. It looked as if there would be many data-processing languages. Now, those who were concerned wanted only one. But we could not work together, because we worked for different companies, and if we agreed on anything, this would contravene the anti-trust laws. The only way we could work together was under the sponsorship of a University or of the Government. We went first to Dr. Saul Gorn, University of Pennsylvania, who convened a meeting. The group became too big for the U. of P. to continue, so we had to go to the Federal Government, specifically to Charles Phillips, the Comptroller of the Department of Defense. He arranged a meeting in the Pentagon with every computer manufacturer represented together with all the biggest users of computers. It was agreed to have one data-processing language, preferably in English. The organization known as CODASYL was established, and within it the Programming Language Committee whose purpose was to combine all the know Languages and come up with COBOL, or Common Business Oriented Language. Less than a year later, there were rumors that COBOL was dying, or if not, dead. This upset Howard Bromberg, the RCA man. Once, as he was worrying about its possible demise, he passed a place where gravestones were sold. With sudden inspiration he bought one, a beautiful piece of white marble with a reclining lamb on the top. He had the word COBOL cut in the front of it, then he shipped it (express collect) to Mr. Phillips in the Pentagon. That was the turning point in the development of the COBOL language. Support was restored, and we defined the language which is so generally used. But if it had not been for that tombstone, we might not have accomplished our task. This spring, it was out great pleasure to give the tombstone to the Computer Museum in Boston.

Of course, we have a tendency every so often to go too far. One of our favorite remarks, for instance, is "Wouldn't it be nice if we could program computers in ordinary English"? A. D. Hill of the U.K. Medical Research Council disagrees: rather, he would prefer to write his instructions in programming language. I will give you one of his examples. "Consider the following from a shampoo bottle: For best results, wet hair with warm water. Gently work in the first application. Rinse thoroughly and repeat." Repant from where, surely, the rule must be that, in the absence of other information, repeat from the first instruction. But this means we have to wet the hair we have just rinsed. Let us use a little common sense and not bother with that. But, the next instruction refers to the 'first application' and cannot do that again, so perhaps logic tells us to miss that one out too. So the only thing left to repeat is, rinse thoroughly and repeat, and now we are in a closed loop, and must continue rinsing our hair aborted. Hill says that it is much clearer to give the instructions in programming language. Thus, the shampoo bottle might be labelled: begin wet hair with warm water; for j = 1, 2 do: begin gently working in application (j); rinse thoroughly: end end. We might keep that in the back of our minds and not try to go too far.

Finally in 1966, and I recieved a personal letter from the Chief of Naval Personnel, the first paragraph of which said, "You have completed 23 years in the reserves, which is more than 20." I knew that. The second one said, "You have attained age 60." I knew that, too. And the third paragraph said "Enclosed forms. Please apply for retirement." The 31st of December was the saddest day of my life, when I was officially placed on the Naval Reserve Retired List with the rank of Commander. Thanks to our highly automated Pension System, I got my first pension check on April 1. Two weeks later, I got a call from the Pentagon telling me that they wanted to talk to me. I was asked to return to active duty with the job of standardizing the high-level languages and getting the whole Navy to use them. The first half of that job was finite, the second was infinite, but I was very glad to make a start on it. I reported on August 1, 1967, for six months' temporary active duty: so far, it is the longest six months I have ever spent. I was given one Lieutenant, one civilian, two D.P.-3's and a Secretary. I had an office on the fifth deck in the Pentagon (the attic); later, we were promoted to the third basement.

We had our orders, but no equipment or money. Well, since I was initiating a new Navy activity, the first thing I did was to buy a coffee urn. The second thing was teach my new crew, in full, the things my Chief had taught me during World War II. Which is, if you really need something you "liberate" it from the Air Force, or failing that, from the Army. They have everything, and they cannot count. We did marvelous job of furnishing the office. Only once we nearly got in trouble, when the captain asked where we had obtained our coffee table. I remembered what my Chief told me to do, I stood perfectly straight and said "It wasn't bolted down." That became one of our mottoes: "If it isn't bolted down, bring it home."

Our first problem was computer time: we had none, and no money either so we worked on the slack. Now slack is something that MBAs rarely understand. It is the little bit of "give" in anything that operates. If you make two pieces of metal so that they roll perfectly on each other, they will freeze and never move again. So you allow a space between them, where you put the grease and everything runs beautifully. That is slack. No good Chief Operator will schedule a single computer to work 24 hours a day, 7 days a week: five minutes are lost here, and ten minutes there. So my crew made friends with every Chief Operator in the Pentagon, who would call us if they did not need their slack time. Now, of course that meant that our programs had to run on any computer. So we wrote a program in low-level COBOL which could do just that.

Two and a half years later we had a set of programs that would tell us whether or not a COBOL Compiler fully met and implemented the criteria of the American National Standards Institute. I was invited to give a presentation to the Secretary of the Navy. I was rehearsed for two weeks. I went through one dry-run after another. Finally on the day as I was walking down the Executive Corridor beside the Captain, he said, "This is the first time a woman ever gave a presentation in this room." About ten feet further along he said, "The first time anybody below the rank of Captain gave a presentation in this room." Oh, was I in good shape by the time I got there! I diverged from my speech, and told the Secretary that it was fortunate that we did not have a budget, or I still would be filling out papers to get our first hour of computer time. At the end I was asked if there was anything that could be done to help. I waded right in, and I asked for two more DP's, two more programmers. and $20,000. The whole room collapsed into a roar of laughter. "Don't you realize "said the Captain," nobody ever asked for less than $20 million in that room before."

We got $20,000 and two more DP's, only to run into an extraordinarily serious problem. We found there was a very large number of civilians, and even some officers, who would not listen to a young man or woman wearing a sailor suit, even though they were trained experts. It became so bad that the Captain finally decided to take our team out of uniform. I have never regretted anything, so much as the day I told those youngsters to take off their uniforms, so that people would listen to them. Therefore, I had cards printed with a Navy seal and I made them all Managers. This had a wonderful effect on the civilians, as they are scared to death of managers. I find, in general, we underestimate our young people, and I think the fault lies largely with the media. They publicize only the 20% of youngsters that are no good, and never tell us about the 80% that are the brightest, and the healthiest this country has ever had.

These top-notch youngsters are everywhere and they are looking for positive leadership. I believe we lost the concepts of leadership after World War II, and went overboard on management. Not only in the Pentagon, but everywhere the Systems Analysts and MBAs were brought in. We forgot about leadership in the old sense. Leadership is a two-way street, loyalty up, and loyalty down. Respect for one's superiors; care for one's crew. In my case, I decided every one of the youngsters should be able to get up and give a report. So when I was invited to give a presentation, I would arrive with my whole crew strung out after me. Then, instead of my giving the report, I would introduce them, one by one, to talk about their part of the work. I watched those youngsters grow two inches, when an Admiral said "Well done." How often do we remember to say 'thank you' and to give praise when praise is due? Yet we bawl people out all the time. Praise is part of leadership, and that is what youngsters are looking for. The Marines never forgot it. Their philosophy is "When the going gets rough, you cannot manage a man into combat, you must lead him." We need leadership not only in the Armed Forces, but in Government and in industry. I was very pleased a few weeks ago, when the Secretary of the Navy sent out a new directive: he told the Washington Post that he was going to stop managing human resources and lead people. Our young people are the brightest and best we have ever had and we owe it to them to remind them that they are the future. The leadership that they pass on will greatly affect the future of this country.

Back to my story: eventually, our routines were accepted, and the order went out that all compliers brought into the inventory Department of Defense should be tested. But this left out the rest of the Federal Government. The National Bureau of Standards came to us for a copy of our program and soon the heading "U.S. Navy Compiler Test" was removed and they were putting out an NBS Compiler Test. I was angry, so when they wanted a new set of the programs, I refused. I was promptly told that, I must cooperate with the National Bureau of Standards. Two days later I handed them a new tape, promising that if they again attempted to change "U.S. Navy" into NBS, it would blow their operating system off their computer.

Eventually we held a peace conference, and the National Bureau of Standards delegated to the Navy the job of testing all compilers for the entire Federal Government.

This continued until about five years ago, until the Appropriation Committee of the House of Representatives was absolutely horrified to discover that the Navy was performing a Federal function. They plucked the Compiler Testing Service out of the Navy and put under the General Services Administration, taking our people with them. They now put out an excellent report six times a year listing all the certified Compilers. The activity has recently been transferred to NBS. I am tremendously proud of what those youngsters accomplished. Two of them received the Navy Achievement Award for work on the test routines. For myself, I probably spent my busiest, most challenging, exciting and interesting eighteen years with these eager youngsters. I have loved every minute of it. I have also received most of the awards that can come to anyone in the computer industry. Each time I have thanked my hosts, and then said something that I would like to repeat to you.

I have already received the highest award I will ever receive, no matter how long I live, no matter how many more jobs I may have. That has been the privilege and the responsibility of serving "with true faith and allegiance" in the United States Navy.

REFERENCES

Hill, I.D. (1972). Wouldn't it be nice if we could write computer programs in ordinary English – or would it? The Computer Bulletin, 16, 76-83.

Slater, L.W. (1982). Everything you ever wanted to know about microcomputers (but didn't know WHO to ask). Navy Regional Data Automation Center Publ., Norfolk, Virginia, 19 pp.

Rendleman, J.E. (1977). The other side of privacy. Armed Forces Comptroller. August 1977, 3 pp.

* U.S. GOVERNMENT PRINTING OFFICE: 1987-711-035/40014