-Begin Transcript-

Welcome back to No Such Podcast.

Today, we bring you the second half of a bonus episode featuring a lecture by computer science pioneer Rear Admiral Grace Hopper.

If you haven't heard part one, we recommend you back and listen.

Here's the second part of "Future Possibilities: Data, Hardware, Software, and People," delivered at NSA in 1982 by then-Captain Grace Hopper.

[APPLAUSE] So we'll build systems of computers. Truth of the matter is it s already happening. I have one chip here.

0:37

The chip is made by Hughes the system s being built by Goodyear aerospace for NASA --

0:43

This particular chip has 8 processors on it. They re lined up four and four, you may want to

0:51

look at it afterwards. I ll have a here, there are 8 processors on that chip -- The system consists of a hundred and twenty eight

0:58

hundred by a hundred and twenty eight processors. [TIME: 00:01:01] Sixteen thousand three hundred and eighty four processors all in one system.

1:05

They call it the multiple multiple parallel processor. Now, of course, you might think that s going to fill a room like Mark One did, not at all.

1:14

All the processors are in one cabinet. Second cabinet holds all input, output and controls. Third cabinet is a PDP 1134,

1:23

because obviously you have to have a computer If you re going to manage sixteen thousand computers. Each pixel from a LandSat

1:30

photograph will go to one processor. Pixel can include position, color, intensity,

1:39

everything else that you know about that pixel, and can communicate with the surrounding pixels. This is to be used to hunt for oil and minerals and stuff, like that there.

1:48

The other chip I have here is the first I ve managed to get my hands on of the very large-scale integration. I have no doubt that there are more of them

1:56

and some of our classified things and weapons systems, but this one is out in the open --

2:01

[TIME: 00:02:01] Three of these chips form the micro mainframe three chips to form a mainframe computer. when you look at it you ll find that the micro processor

2:11

is just one of the processors on this. It has built in multi- programming all the other fancy things -- The micro mainframe --

2:20

We re beginning to be able to build systems of computers and the sooner we start doing it the

2:26

better we can respond to the need for rapid response and Rapid development of things. --

2:32

Now we re not going to get there overnight -- We can t suddenly, throughout all of our big dinosaurs and replace them, with systems of computers.

2:41

It will have to be a step by step procedure -- I d like to show you something about how

2:46

we can get at this sort of concept -- This is my horrible example. It s a DoD system --

2:57

Not Army,Navy, Air Force, Marines or Coast Guard. DoD, that s different --

3:04

[TIME: 00:03:03] They went out and bought themselves a 256K computer and of course they were going to put great big problems on it.

3:13

Along came the manufacture and said you can t run that thing without a --

3:18

An operating system so they loaded in an operating system, of which 35K is

3:23

resident at all times, the rest is out on disk, you call it and when you need it. Well they I didn t want to take now naps or rest, so they said, we d better multi program it. So in

3:32

went a multi programming system, of which 28K is resident at all times. Some of you may recognize

3:40

which computer this is but don t mention it out loud. It s rather well-known one --

3:45

then they want to hang terminals on it they could put in, ask questions and put in programs, so they had to have timesharing 35K -- Plus 2K for each of six terminals

3:56

for total of 47K for time sharing -- then they want, then they want communications,

4:02

they want to talk telephone line, so that was 10 K and, not too bad, and then they had to have security -- [TIME: 00:04:08]

4:09

So far we are doing very, very badly on the score of security, we re doing a very poor

4:14

job. Not only in government but an industry at protecting valuable information and protecting

4:20

against fraud and theft. I think mostly people forget how many reasons or are for -- For security, first, of course,

4:27

as national security, but we ll leave that to you and and the department of defense, this company security company confidential information that has to be protected.

4:37

You realize there have already been cases where companies have listened to each other s microwave when they were bidding competitively on projects? That s not only not very nice.

4:46

It s illegal -- Then there s the question of personal security. Personal information you realize there have already been cases where people access personal

4:55

information and use it to blackmail people? That is also illegal, and then this fraud

5:01

and theft and regrettably, it s increasing -- I lecture in the white collar and computer crime

5:06

course down at the FBI Academy. [TIME: 00:05:09] I asked them if they had a message they wanted me to give people and I ve been

5:12

spreading it all around the country. They said please tell people if their system is broken, let us know, we ve got to know what s happening.

5:21

They showed me the case. A man in a bank, an officer of a bank,

5:26

stole a large amount of money using the computer. He was discovered and he was fired

5:31

But the bank didn t want the stock holders and depositors to know they d lost all that money, so they covered it up. So he threatened to go to

5:39

the newspapers unless they gave him a letter of recommendation and they did.

5:45

He went to a second bank. He stole a large amount of money. It was discovered. He was fired and he blackmailed them for a letter of recommendation It wasn t until he got to the third bank

5:55

that it was reported to the FBI as it should have been in the very beginning. The failure to report was very, very costly. They also showed me another case. We haven t

6:06

been telling our youngsters that it is not funny to use other people s computer time and to erase

6:11

other people s data. A youngster out in -- uh -- Kansas City broke the network of a software

6:20

house that was providing services to other companies. He got tired of playing with it, so he sent a message through the network to the managers at the network, and he said, I have all

6:29

of your access codes and he listed em. [TIME: 00:06:31] I have all of your personal ID codes and he listed them, and who they belong to. Then he

6:36

said I ll tell you how I did it. And I ll promise never to do it again, if you ll give me a red chevette equipped with a radio telephone and please- fill it with gasoline --

6:47

The FBI got him -- Then there was the case, was in Washington, New York papers, Canada Cement Lafarge, found that someone was not only using

6:56

their computer, but wiping out their data -- So they called in the Royal Canadian Mounted Police in (Bellev?) Canada, and the calls were traced to the -- uh --

7:04

border. They called in the FBI and Mother Bell, and the calls were traced to the Dalton School in New York City. The offenders turned out to be four 12-year old boys,

7:14

now known as the Dalton Gang -- (laughter) So, um, I said to the FBI, well, what are you

7:20

going to do with all these youngsters, and they muttered about juvenile delinquents. I said,

7:25

oh, don t do that; make them work for us. They might be able to make the worldwide military

7:31

command and control system work. -- [TIME: 00:07:37]

7:40

Certainly throughout industry, I have found major difficulties in security. I was tickled to pieces for those knowing an insurance company in Portland Maine --

7:51

They looked at their data and they said: Hey we ve got two databases here, not one big one.

7:56

All of this information deals with our policy holders. This information deals with how we run our company.

8:03

We need two database machines -- which have different classification so to speak.

8:09

People are beginning to look at the data and how you re going to protect it.

8:14

But so far, while this system has been broken about every two weeks or so, so security is still growing- 17k for security -- And then we re all lemmings in this

8:23

business. If anybody has something we all have to have it, so of course they had to have a database management system -- In went a data base manager and system, 23K --

8:32

For total of a 160K of overhead, pure overhead -- [TIME: 00:08:37]

8:38

That minor earthquake they had up in New England did not come from the Norumbega fault. When my sainted Scott s grandfather heard about this, he rolled over his grave --

8:47

160K for overhead. That left them with 96K to put their programs in, but they couldn t write a 96K

8:55

program because they re multi-programming, so they had to have at least three partitions in there.

9:00

Programmers are limited to 32K per program and ain t that ducky? You buy a 256K computer,

9:06

you can write a 32K program. Have you looked at your overhead lately? Let s get rid of it! Let s start building a system of computers --

9:16

Let s put it in one microcomputer and have it handle all internal consoles and, and security;

9:22

another one and have it handle all external consoles and security.

9:28

Now I ve begun to protect my system. This is my operating system. If I can access your operating

9:34

system, I ll break your system. If you want to how to do it, I ll tell you, too -- afterwards. Now I

9:39

put micros out front -- [TIME: 00:09:42] A terminal calls in. The micro asks for the ID of the person operating the terminal. You give it.

9:48

The micro breaks the connection and calls back the terminal on which that ID is legal. Now, why go through a callback procedure? Look at those kids calling in;

9:56

they got the telephone number of Canada Cement (Lafars?); they called in. The microcomputer asks

10:03

for an ID. You give an ID. Now, it s you with the borrowed, stolen, copied, or bought someone s ID,

10:09

and all those have already occurred. You give it. Micro breaks the connection. It ll never call back the illegal terminal. And short of piggybacking or an extra wiring

10:19

within the building, you don t get over that. That s another advantage in that either the micro or the mainframe can change the number of the term -- the terminal number.

10:29

So you may know your access number, but you do not know the number it s sending to your terminal because it may have been changed 2 minutes ago. And it s a fairly good way.

10:37

I wish more of the banks we re using it, the call back procedure, because it s much easier to keep people out of system to begin with than it is to get em out after they ve gotten in.

10:47

[TIME: 00:10:47] Now, let s put a third micro in and have it handle the database -- database manager. That concept has a long and interesting history and it s a

10:58

good example of the way we do not accept -- new ideas. It was first proposed by Bell Labs

11:04

in 1972, and they showed that a back-end computer went faster and cost less to

11:09

manage the data. Nobody paid any attention. In 74 they published a paper, "Back-end Computer

11:17

for Database Management." I can t believe people read it. Nobody did anything about it. Usually, we listen to Bell Labs. 77, the Army working with Kansas State

11:27

University in (Cullinane?) put a back-end computer for database management, showed it went faster and

11:32

cost less, but even they re not doing it. Three years ago, at the National Computer Conference at Anaheim, they showed the first of the database machines, a machine built to

11:42

do database management. Not a computer, doesn t compute -- except (account one?) once in awhile. A

11:48

database manager -- people didn t pay attention. [TIME: 00:11:52] Finally, this year, 10 years after we first latched onto the idea, the Navy s putting a

12:00

database machine in at Point Mugu. And I have a paper here by Frank Miller-Barber, "Database Machines: It s about Time." I can t understand why it s taken us so long to

12:10

realize that a better, specialized machine to handle database was always going to go faster

12:16

than a general-purpose machine. Plus the fact that the minute you go to database machines, you can have more than one database -- And in the Navy installation we re going to have

12:25

one database which is highly classified, and another one which is open -- And it means it more that than one computer can access the database --

12:33

So the concept of database machine is very much at the heart of building the systems

12:38

of computers of the future. And the concept of multiple databases. And that was I why so

12:44

happy with that New England insurance company that recognize that right in front of them it was clearly two separate databases,. One for the insured and one for their own company.

12:54

[TIME: 00:12:54] And if we look at our data, we ll find in most cases, that it does breakdown pretty clearly. After all, it was never any reason why we

13:00

put inventory and payroll on the same computer. The only reason we did, was because we only had

13:06

one computer. Now, we ve got to overcome that concept and realize those can be on separate

13:12

computers. And, of course, the minute we get to systems of computer, we don t go down anymore, which is another important point. Now I can put three more computers in here: four,

13:20

five and six to run those three problems that I wanted to run. They ll run and parallel, instead

13:25

of sequentially, under multi-programming. And today they can be 64K micro instead of 32K. Next

13:32

week, of course, they ll be 256K, or something -- I can put another micro in, so they can all talk

13:38

to each other. It cost me less than 1/3 what the big system did -- So the sooner I move to systems of computers,

13:46

the faster I m going to go -- The better security I ll have --

13:52

And the less it will cost me -- [TIME: 00:13:54] All good reasons for moving toward systems of computers. But it won t happen overnight.

14:00

Only recently have we persuaded our Admiral, to send out the order to all of the Nordics --

14:05

Identify the first system you re going to pull off and put on a separate computers --

14:11

Maybe by the time our big mainframes are about to pass out We will begin- have begun to get our things into a system of computers. It means looking ahead.

14:21

It means planning ahead. And it s going to take us the next 5 to 10 years to get there. And it s awfully hard to get somebody to do it on their watch.

14:29

When the payoff will not come this year. But we ve got to begin to look ahead and plan for the systems of computers of the future,. And I hope at least one

14:39

of you will get a set of those nice -- -- gadgets they use in chemistry and start

14:45

constructing a system of computers. And you can even have different kinds of bonds between him,

14:51

one of which you know is classified, the others not, and so on and so forth. [TIME: 00:14:55] I think you ll have lots of fun starting to

14:57

design the systems of computers of the future -- Well, so far, I ve talk only about data and hardware -- We ve got to do something about software.

15:07

One of the big cries today is that software cost too much, and you can t maintain it -- Combination of a proof by Dr Haney and my crew -- We developed the technique which has paid off

15:18

in a rather big way, particularly from the point of view of maintaining a system. Now

15:23

Dr Haney proved something rather interesting. You may want to repeat it to your programmers, it s very good for them. Programmers tend to believe something that s mathematical. So if you can get

15:33

a mathematical proof, they are much more apt to believe it, than all the orders in creation --

15:39

Dr Haney said, suppose I have a piece of a program and I make a change in it.

15:45

Do I have enough programmers here to know what happens when you do that? You probably have to make another change. The darn things ripple.

15:52

All right, he said, let P be the probability that if I make a change, it causes another change --

15:58

[TIME: 00:15:57] And then he said, I d like to know, on the average how many changes do I have to make as a result of that first change.

16:03

Well, that s the expected value statisticians have told us how to compute it -- E is the first change plus the second change times the probability that I have to make it p,

16:13

but sometimes this second change causes a third change p-squared, sometimes a third causes a fourth p-cubed, it s the infinite series -- The probability is always one or less so the

16:24

mathematicians told me, have told me that in that case the sum of that series is one over one minus p -- So if I have some real good applications

16:33

programmers, the chances if they make a change is going to be 1 in 10 that they ll have to make another change. This becomes 1 over 1 minus a tenth,

16:41

1 over 9 tenths -- 1.11111, which looks like something I can live with it -- But I ve practically never have a system

16:49

which consists of only one module. So now, let s have another module --

16:57

I know turn I ll behave like a normal mathematician, this, ith module. [TIME: 00:17:01] This is the jth module --

17:03

And now p(i,j) is the probability that if you make a change in the ith module it will cause a change in the jth module. I change something in the selector

17:11

that upsets the I/O -- Now I d like to know --

17:17

What is the -- How many changes do I expect to make as a result of that first change. Now incidentally we ve run into that more

17:25

than once, I ve known cases where they made one change -- it caused 30 changes, that caused about 500 changes, and at that point they decided they

17:32

better throw it away and start over again -- In this case, Dr Haney s proof shows that

17:38

in this case the expected value turns out to be 1 over 1 minus np.

17:44

Where p is the probability that a change in one module, causes a change in another one and n is the number of modules -- Now that n --

17:53

Suppose I have a hundred modules in this system. It s an operating system or a database management system or a great big application. And now I have systems programmers, who are much

18:03

better, of course, than applications programmers. [TIME: 00:18:06] To know the probability that they make a change that s going to cause

18:09

another change is one in a hundred. How many changes will I expect to make as a result of that first change? This should be a minus sign, I m sorry. One over

18:18

1 minus 100 times a hundred this one -- which is one over zero, which is Infinity

18:25

and the system will not stabilize -- We ve been awful close to that point. You can probably all remember cases where we change something in the I/O and to a week later

18:34

the whole system blew up -- And you didn t know where it started from -- Is there an answer to it?

18:40

The answer is yes, there is. that is to make all of the P(i,j)s zero.

18:46

What s that mean? It means you write the system in independent modules. Modules have one entrance point and one exit point.

18:54

And no module ever accesses the interior of any other module --

18:59

Never touches it. And the way you exchange data between the modules is through a series of interfaces. This module put something down,

19:06

another module picks it up -- [TIME: 00:19:08] I asked my crew to build me a COBOL compiler for the 8K, ruggedized Nova. Of course, everybody

19:16

knows you couldn t do that, but they did -- I asked them to build an entirely in independent modules and they developed a technique which I found extremely useful, and so did they.

19:26

They had a micro computer that kept all the modules on the micro computer as a file.

19:32

So anytime anybody wanted to look at one of the independent modules, he could call it up and look at it -- They also kept a file of the interfaces --

19:41

By the number of the interface, the contents of the interface and who was in charge of the interface followed by a list of the other modules and who was in charge of it. If anybody wanted to

19:52

change and interface, you can immediately find out who he had to check with before he could change that interface. And they didn t have that awful case of a change something over

20:01

here and it blows up over here next week. They had those, the modules themselves and the interfaces

20:07

completely under control. [TIME: 00:20:08] They were in the micro-computer and any given moment any one of them could look at them.

20:14

And -- One sec-, one second class petty officer, two third class and one seaman in four months time built a

20:22

COBOL compiler, that s normally two years -- They did a magnificent job and part of it --

20:30

I think was the fact that they had the file of the modules and the interfaces to work with. Also,

20:36

when they were transferred and a new group came in the documentation was complete for them to add

20:41

to it, to change it, because any module could be pulled out without affecting any other module. All you had to check was those interfaces -- And there was a complete

20:49

record of those interfaces -- So, maybe there s something we can think about. Toward easier building of systems of software and of managing not only the software

21:00

development itself, but also the maintenance. Well it finally got to be 1966 and I got a letter

21:07

from the chief of naval personnel -- [TIME: 00:21:09] And the first paragraph said: you have completed 23 years in the reserves,

21:14

which is more than twenty -- and I knew that. The second paragraph was also aimed right at me.

21:22

It said you have attained age 60 -- and I knew that too. The third paragraph

21:27

said here are the blanks please apply for retirement. So I did. And I was officially

21:33

placed on the naval Reserve retired list with the rank of commander on the 31st of December 1966 -- the saddest day of my life. I left out one thing. Going back to

21:44

that course I had from the -- War college. The second review I was to make was to make the review of the cost of not doing something.

21:53

Now we did that for all the islands. We looked at the cost of men and materiel of taking an island,

21:58

then we looked at the consequences of bypassing the island. And some we took like Saipan and Tinian -- Others we bypassed like (Truk?)

22:07

That study was always made. What is the cost of not doing this? [TIME: 00:22:11] I ve long advocated the use of

22:13

the standard, high-level languages. And I d go to one of our data processing installations and say

22:18

why aren t you using standard Cobol instead of all those bells and whistles the vendor gives you?

22:23

Oh we get a few extra bits and bites or something like that or microseconds.

22:30

I said if you looked at the cost when you go to the next computer -- Cost to conversion -- Well, we ll think about that later.

22:39

I finally decided that I had to find somebody that would really scare people. So I looked around Washington to see which agency really did scare people and I

22:48

decided it was the General Accounting Office. That s because they come and look at everything you re doing and then go tell Congress about it. So I went over to GAO and said: do you realize

22:56

how much it s costing us not to use the standard, high-level languages? And they said "No." And they made a year-long study. Yeah, so they came

23:04

up with was rather interesting -- They said that between that time and 1985 about 8,500 of the general management computers in the total federal inventory

23:14

would be replaced. [TIME: 00:23:16] And, furthermore, that the cost of conversion was running 450 million dollars a year.

23:23

450 million dollars a year down the drain, simply because we had failed to implement

23:29

the standards. Money we could have had for all the things we wanted to do -- and far outweighed anything we saved by using the special bells and whistles that any vendor gave us.

23:40

That was a case where most people totally failed to look at the cost of not doing something.

23:45

They d tell me, oh well they couldn t possibly enforce the standards this year. They d have to get new forms, they d have beat all the programmers over the head. They d do

23:53

the standards bit next year -- and they wouldn t look at the cost of not using the standards.

23:59

So that review goes right up along with the all possible enemy actions,

24:04

the cost of not carrying out the action. Well, finally they retired me on 31-December

24:12

66. Thanks to our highly automated pension system I got my first pension check on the

24:17

first of April. [TIME: 00:24:19] Two weeks later, I got a call from the Pentagon: "Come down to Washington,

24:23

we want to talk to you -- So I came arunning, as I always do when the Navy sends for me -- I had to wait in Mr. (Reem s?) outer office.

24:32

There were two captains there -- Naturally, since I was only a commander, I

24:37

spoke to them very pleasantly and respectfully -- However, inadvertently, I managed to tell those

24:42

two men, my aren t the captains young nowadays -- One of them turned out to be my new commanding

24:48

officer, so I started on the right foot -- Mr. (Reim?) took one look at me and said "the Navy payroll has been written, 823 times, this has got to stop." So naturally I said "Yes, Sir."

24:59

It ended up he asked me to return to active duty with the job of standardizing the high-level languages and getting the whole Navy to use them. I realized the first half of that job was finite,

25:08

the second was infinite, but I d be very glad to make a start on it. And so I reported on the first of August 1967 -- on six month s temporary active duty. And

25:20

so far it is the longest six months I ve ever spent in my entire life. [TIME: 00:25:24] As for Navy personnel, they gave me two Navy

25:27

men, DP-3 s -- a civilian, and a secretary -- An office on the 5th deck of the Pentagon -- that

25:34

s the attic -- slant roof, escalators go to the fourth deck and you hike to the fifth. They gave us each a desk and a chair and a pad and a pencil.

25:43

Standardize, the high-level languages start with COBOL -- No cards, no tapes, no computer, and no budget. Since I was initiating a new Navy activity,

25:53

the first thing I did was go out and buy a coffee urn. The second thing I did was teach my new crew the things the Chief

26:00

had taught me during World War Two. And I d like to assure you that our new Navy men and women are just as good as any World War Two men ever was.

26:09

It only took them two weeks to completely furnish the office --

26:15

The only day we nearly got in trouble, they turned up with a coffee table like they have

26:21

down in the secretary s office, and the captain took one look at it and said: where did you get that? I remembered what to do, I just stood perfectly straight and said: Captain,

26:31

it wasn t bolted down. [TIME: 00:26:33] So, um, that became one of our mottos -- if it isn t bolted down, bring it home.

26:41

In fact, we did so well that the junior officers at (Nav-consact?) eventually gave us our own flag. It s a beautiful nylon bolt flag with grommets

26:49

and everything -- on a pole beside my desk -- and it s a skull & crossbones.

26:55

And to the best of my knowledge we were the only office ever in the entire Pentagon to openly fly

27:01

the Jolly Roger and operate under its aegis. Of course I also taught my crew that when we go

27:07

out to get something always liberate from the Air Force first, because they have everything and don t know how much they got. Second, if you can t get it that way,

27:16

try the Army because they have almost everything and they can t count. And -- It s no use trying

27:24

to liberate anything from the Seabees of the Marines because they liberated it to begin with.

27:30

We survived -- one way or another, on no budget -- [TIME: 00:27:34] We had no money to buy computer time.

27:37

We lived on the slack -- in the Pentagon computers.

27:43

No good chief operator will ever schedule a computer 24 hours a day, 7 days a week. You leave 5 minutes here, 10 there, 15 there for the things that happen when you operate

27:51

systems of computers. If they didn t need it, slack time. My crew made friends with ever chief operator at the Pentagon. I never knew

27:58

how they got in some places, and I didn t ask. They d call us and say: can you use five minutes?

28:04

Can you use 10 minutes? That meant our programs had to run on anybody s computer. Because

28:09

everybody knew you couldn t do that. Except my crew didn t know you couldn t do that. One Third-Class Petty Officer wrote a program, in low-level COBOL,

28:19

that asked you which computer you re on. You can say anything from Apple to Amdahl, 370, 3500, whatever you ve got your hands on. And as it ends the remainder of the program

28:29

it enters the control cards and special names, and you re on that computer. And you can write programs that ll run on anybody s computer. And the entire set of test sets of test routines for

28:40

both COBOL and Fortran compilers have been run on everybody s computers.

28:45

[TIME: 00:28:45] There s no excuse for not writing programs that are portable. Well, that solved our problem of computer time.

28:52

And we began to develop the test routines -- first for COBOL, later for FORTRAN.

28:57

Two and a half years later, I was invited to give a presentation for the Secretary of the Navy. It was going to be the Secretary of the Navy, Assistant Secretary,

29:05

Chief of Naval Operations, Chief of Naval Materiel, all of the Vice-Admirals. Oh, they rehearsed me for two weeks ahead of time.

29:13

I went through one dry run after another, until I was letter perfect.

29:18

We finally got to the great morning, and I was walking down the corridor, beside the executive corridor beside the captain. He looks down at me and says "first time a woman

29:26

ever gave a presentation in that room" -- and so I d feel more at ease, more comfortable.

29:31

He let me get about 10 feet further and he says "first time anybody below the rank of captain

29:36

ever gave a presentation in that room." He had me in good shape when I got there. So the first thing I did was break from the speech and tell the Secretary it s just as well we hadn

29:44

t had a budget -- because if we had had one I d still be filling out the papers to get our first hour of computer time. [TIME: 00:29:51]

29:51

But I managed to get through to the end and he was a charming gentleman. He thanked me, he congratulated related me -- And then he said: is there anything we can

29:58

do to help you? Now I hadn t been briefed on how to respond to that question. So naturally, I waded right in -- and said yes,

30:06

I wanted two more DP s, two more programmers, and twenty thousand dollars to survey the users

30:12

of COBOL to find out what they needed next. He said he d see what he could do about it, and that whole room collapsed in one raw laughter. Boy did I get out the door.

30:21

I looked at the Captain and said: well, what did I do this time? He said: don t you realize? Nobody ever asked for less than twenty million in that room before.

30:34

So we got two more DP s and twenty thousand dollars, only to run into a major problem with our DP s in the Pentagon. We found there was a large number of civilians and

30:45

even some officers who were not about to listen to a young man or woman who wore a sailor suit.

30:52

[TIME: 00:30:51] It got so bad that the captain finally said we will have to take them out of uniform. I don t think I ever regretted anything more

30:59

Back in the day I had to tell those youngsters "take off your uniforms, so the people will listen to you." I tried to make up for it. I personally

31:08

paid for having cards printed for each one of them with the Navy seal on it, and I made them ALL managers. Mr. George (N. Bet?), Manager,

31:16

Test and Evaluation. Why I had a marvelous effect on the civilians, ever respectful of managers.

31:22

It even had a good effect on some of the offices, particularly Army.

31:28

I find we have a very bad habit underestimating our young people. I think we totally failed or recognize how much more they know than we knew at the same age.

31:40

I can make the comparison -- They ve had radio and television all their lives long for both information and misinformation. I didn t have a radio set until I was a senior in

31:50

high school, I built a crystal set. [TIME: 00:31:51] I didn t have a vacuum tubes set until a senior in college, that was the year

31:55

the superheterodyned circuit first came out -- I knew man would never walk on the moon. They know

32:02

he has. They know all about jet airplanes. They can t remember their first flight in an airplane,

32:08

never taken on a jet to visit their grandparents when they were babies. I didn t fly on an airplane until I was a sophomore in college. I spent 10 dollars, and

32:18

that was a heck of a lot of money in 1925. I want to put an open cockpit biplane,

32:25

built out of wood and linen and wire, and it went up about a hundred and fifty feet

32:30

and floated along about eighty miles an hour. I d be scared to death to go near it today. They know all about jets. I was reminded of this not long ago, because I was walking

32:39

out to take Allegheny city big commuter flight from Washington to Philadelphia- I guess it s ransom airlines now. And there was a young man beside me.

32:48

He was looking up at that plane, finally he turned to me and he said is that thing safe, and I said yeah, Why not? [TIME: 00:32:54]

32:54

He said I ve never flown in a prop plane before. We have a whole generation that has never flown in prop planes. We ve totally forgotten how much

33:03

more they know than we knew at the same age. On the other hand, they are no more mature than

33:09

we were at the same age. And they are looking for something which they cannot always put in words.

33:15

And I ve seen it across the country, I ve talked to schools and colleges and to our young people. What they re looking for is positive leadership. I mean leadership in the

33:26

old Navy sense. It s a two-way street, it s loyalty up and loyalty down, it s respect for

33:32

your superior and keep him informed of what you re up to and take care of your crew.

33:37

We ve forgotten that. We think leadership only comes from some guy up there at the top. It s everybody s job. It s everybody s job to take care of their crew.

33:47

For example, I decided that youngsters coming through my group, that every one of them should be able to get on his feet and give a report, and not once say you know.

33:56

[TIME: 00:33:56] So I ll put a little square box on the desk with a slot in it and if they said "you know" during a report, they had to put a quarter in. And we didn t take the quarters, but it tied up the capital and you d

34:07

be surprised how fast they learned not to say "you know". I also decided that everyone of them must

34:12

be able to write a report and correct English statements and spell all the words correctly. I bought five Dictionaries. It also meant a lot of extra time for me with the red ink.

34:23

And nights and weekends and then the explaining the changes. Yet before they left, every single

34:28

one of those 50 or so youngsters was able to give orally and in writing a correct report in

34:34

good English and spell all the words correctly. I think we forget that the four and five year old

34:40

s are learning arithmetic. A little professor. The six year old s are getting speak and

34:46

spell. You better lookout, there s going to be a generation coming, that will know how to spell.

34:52

The seven-year-olds, of course, are learning BASIC, running the computers. I know one man that bought a computer and took it home, his son is teaching him BASIC. His son is seven.

35:02

[TIME: 00:35:01] Of course I know another guy that took a computer home, now he has to apply to his three children for computer time.

35:09

They re tremendously bright and they are out there, the brightest youngsters we have ever had. Now, they are not coming from the two coasts and the big cities,

35:18

they are coming from farm country, where they still believe you have to work to earn something and work to learn something. North Dakota, South Dakota, Idaho, Wyoming, Utah, Nebraska, Kansas,

35:29

Kentucky, Tennessee, Northern Alabama, Northern Georgia, Southern Indiana, Southern Illinois, West Texas. They are coming from out there in the farm country where they still have good schools,

35:39

and they still believe you have to work to learn something they re out there. The brightest we have ever had. And yet somewhere in the last 30 years,

35:46

we lost that word leadership. We went overboard for management, partly under the influence of Mr.

35:52

McNamara, partly under the business schools. We concentrated on this quarters bottom line,

35:59

this years fitness reports. We forgot to look ahead for the next five years for any enterprise,

36:06

and we lost that concept of leadership. [TIME: 00:36:08] Loyalty up and loyalty down. It s the one thing those youngsters are looking for.

36:14

You can t do it all by management. If I had a Marine standing here beside me, what he was say would be: When the going gets rough, you cannot manage

36:22

a man into combat. You must lead him. And I think he would further add:

36:27

you manage things. You lead people. We need people, we need to bring that back very

36:33

badly, not only in the armed forces but in all of government, throughout business and industry.

36:38

It is the one thing that those youngsters are really looking for- good, positive leadership.

36:45

Well, eventually, the tester teams were successful, the order went out which that all compile is brought into the inventory of the department of defense should be tested.

36:53

You Navy shall set up and compile testing service which was done and we started testing all fortran and COBOL compilers for the department of defense.

37:02

That left the rest of the federal government to worry about. National Bureau of Standards came and got a copy of our programs who was glad to share them, until two weeks later --

37:11

[TIME: 00:37:11] Our programs had been printing out at the top of every page of every report U.S. Navy compiler tests. I found them

37:18

used in Washington and they were printing out NBS compiler tests and boy was I mad.

37:24

That came around wanted a new updated set of the programs, and I said I wouldn t give them one. I was promptly told politically i must

37:30

cooperate with the national bureaus standards -- So I thought it over for two whole days and I

37:37

finally gave them a new set of the programs and, as I did, I promised them if they again attempted to change us Navy and NBS. It would blow their operating systems

37:46

off their computer -- Will eventually, we held a peace conference and exchanged our prisoners of war and the national --

37:55

And the national average stand delegated to the Navy, the job of testing, all COBOL and fortran compilers for the entire federal governments that continued to about two and a half years ago.

38:06

At that time, the appropriations committee of the house of representatives made a horrifying discovery they discovered to their horror that the Navy was performing a federal function --

38:18

[TIME: 00:38:17] In their completely horrified state, they pick the (component?) testing service out of the Navy and put it

38:23

over in the general services administration -- Now I might be real worried about em over there

38:29

except I d trained them all they re all ex-Navy. I think they re doing alright.

38:34

Since they got over there, they ve gotten all new offices, all new furniture and two new computers -- Every three months they

38:42

put out a certified, compiler list. Which list all compilers which have been tested and certified to meet to the Yangtze specifications all the way from Apple to amdahl

38:52

and everything in between, and there is a COBOL for Apple and it has passed the standards tests

38:59

for both COBOL and Fortran. They re all listed And you can even get the details to the test, most of them cost around eight or

39:06

nine dollars for the complete details of the test of any given compilers. So -- At last,

39:13

we re beginning to be able to implement and the standard high-level languages -- [TIME: 00:39:17] I m tremendously proud

39:19

of what those youngsters have accomplished. Two of em received the Navy achievement award for leading the development of those test routines. One received the Navy achievement award

39:28

for leading the development of that -- the COBOL compiler for the 8k ruggedized Nova. But

39:34

particularly proud of one member of the group. She finished or courses, She was a second class petty officer, so we shipped her off to OCS. She s now on

39:42

a cruise missile installation in Germany -- a lieutenant and she has just been selected to attend the naval post graduate school. For myself, I probably spent the busiest, most

39:53

exciting, most challenging and most interesting fifteen years I ve ever spent in my entire life

39:58

and I ve loved every single minute of it. I ve also received most of the honors that

40:03

are given to anyone in the computer industry -- Each time I received one I thanked them and then I

40:09

told them something I d like to repeat to you -- I have already received the highest award.

40:15

I will ever receive no matter how long I live, no matter how many more jobs that I may have --

40:22

[TIME: 00:40:21] And that has been the privilege and the responsibility of serving very proudly in the United States Navy.

Thanks for listening to this episode of No Such Podcast featuring Captain Grace Hopper's lecture, "Future Possibilities: Data, Hardware, Software, and People," delivered at NSA in 1982.

And if you haven't listened to the first season of our podcast, check it out.

We'll see you again soon

-END-