JG: My name is Jack Gilmore and I gave the second lecture of this series [DIGITAL HISTORY] in June. The first one was given by Bob Everett, covering the Whirlwind computer. I covered the early graphics work on the TX-0 while it was at Lincoln Lab and then some graphics work on the PDP-1 at Itek. Before we begin I'd like to acknowledge a few people in the audience. First of all we have one of the fathers, if not the father, of time sharing, John McCarthy. We also have Marvin Minsky, father of artificial intelligence, and I believe Ted Johnson, who did a tremendous amount of work getting Digital off and running in sales. We are celebrating today the thirtieth year of the PDP-1, and this is quite significant. I'd like to welcome members of the press and the media.

I have two slides that I'd like to pop up and then I'll turn the meeting over to these youngsters. At the last lecture series we talked about the TX-0 and the the early graphics work that was done there, and the advent of the light pen by Ben Gurley, and the early scope writer graphics work. Then we pointed out that this was really the beginning of three other branches, one growing straight ahead to the more conventional but still very exciting graphics work that Ivan Sutherland and others developed on the TX-2.

One other point. There's an individual who isn't with us who played a very major role in the development of the work on the TX-0 and on the PDP-1, and that was Ben Gurley. Ben died a tragic death in '63, but I felt it was appropriate to remember him as well tonight.

[MOTIONING TO PANEL] You're looking at the "right stuff" of the early computer world, so dig in and have a good time.

SL: I wonder if that makes me Tom Wolfe! My name is Steven Levy. I'm the author of Hackers which had a few nice words, probably not enough, to say about PDP-1. We have a terrific panel, people who could tell you all about the computer which was, as I found out when I researched the book, an incredibly significant advance and one which is under appreciated. Rather then my telling you what I found out from them, let me turn it over to them, one by one, to tell you about it and then we'll open things up where you can ask questions and they can contradict each other and maybe blow some holes in apocryphal stories or whatever.

Our first panelist is Jack Dennis who, at the time that PDP-1 was delivered to MIT thirty years ago, had just joined the faculty as an assistant professor. He's a professor emeritus, and he is going to talk a bit about

the TX-0 and PDP-1.

JD: The TX-0 arrived at the MIT research laboratory of electronics in 1959. MIT's Whirlwind computer, which was the first powerful stored program machine, was decommissioned in the late 50's. The main computer facility then became the IBM 704 installation in building 26 in the MIT computation center. The 704 was basically a punched card operation. You brought your deck of punch cards to the door, passed it in to the operators and a couple of hours or the next day you'd come back and see what your results were. That kind of operation was very discouraging to many people, particularly to John McCarthy here, who wanted to get some really good AI programs operating and couldn't see how they could be developed well unless you had more a more suitable way of interacting with the machine. That was the motivation behind the development of the time sharing idea and a little bit in anticipation of that, the TX-0 was run at MIT as an kind of open shop system, where programmers or users could come in, have the machine entirely to themselves for a period time for which they would sign up. At the beginning what we had to work with from the Lincoln Laboratory was the hardware, as Jack pointed out 8K of 18-bit memory, and he mentioned that at Lincoln Laboratory the TX-0 had had 64K of memory. 64K of memory requires a 16-bit address. With 18-bit words how much

does that leave over for an operation code? It leaves 2 bits. So when TX-0 arrived at MIT, it had a 2 bit operation code. Talk about reduced instructions set computers!! This was a reduced instructions set computer! We only had 8K of memory when the machine was delivered to MIT so we decided to take some advantage of the three extra bits that were left over, so we implemented a 5-bit instruction set - slightly less reduced but also fairly effective for our uses. The way the machine was used at MIT was as a laboratory tool, a tool for people in research laboratory electronics to use to connect to their experiments, to process data from instrument recordings on magnetic tape, and so on. It was very much a hands-on operation. To support that work, some very interesting software developments were made. We received from Lincoln Laboratory a tape called UT 3. Utility Tape 3. This was an interactive debugger but you could only talk to it in terms of octal addresses, and if you were trying to develop a large assembly language machine code, it wasn't too convenient to simply work with octal addresses, so professor Thomas Stockman and myself developed a program called Flit. Flexowriter Interrogation Tape. We put that together and this was a symbolic debugger, it actually could use the symbolic addresses from the symbol table generated by the assembler, so another innovative feature it had was it allowed break point tracing of programs at execution

time. As far as we can tell, this was the first interactive debugger with those two facilities both symbolic addresses and a break point tracing of programs.

The other piece of software which we developed for the TX-0 was a macroassembly language program. This was an evolution from a tape which we also got from Lincoln Laboratory. The paper tape we got from Lincoln Laboratory in this case was for a very primitive assembly language program, which came as a binary tape with no listing to it. So the first thing I had to do with it was decompile it, to find out what it really was doing. That was kind of tricky because some of the tricks that one could do on the TX-0 where there's primitive instruction code were very subtle. [For instance,] after you had completed a successful comparison, you knew that the number 1 was left in the accumulator of the machine. A neat trick was to use that one to index something else that you needed to index. There are a number of tricks of that sort which one could use, and this program depended upon it. I developed a macro assembly language program, a similar program from that, which was called macro. Eventually that got translated into a similar program for the PDP-1.

When the PDP-1 arrived as a generous gift to the MIT electrical engineering department from DEC, John McCarthy was around with his time sharing so we decided that we

would develop a time sharing system for the PDP-1. That went into demonstration at the end of 1963, and operation with users during 1964. This was an unusual form of time sharing, because we wanted it to support the kind of use that the TX-0 and the PDP-1 were being put to within the research laboratory of electronics, which was to work with various laboratory groups in hands-on computation within the laboratory. This time sharing system was not designed for remote use, but designed for shared use of the machine within the computing facility itself. There are a couple of interesting things about this time sharing system. One was that it supported interaction of time shared programs with user IO devices; that's something which even modern systems are fairly weak on. The other thing was that we put in a trap facility into the PDP-1, such that the debugger could be fully protected from users programs, so we could run our DDT debugger developed for the PDP-1 on the basis of the work on Flit earlier. We developed that so that no matter what your user program did, it couldn't upset the debugger in any way. The facilities which we built into the PDP-1 for that purpose, it seems to me, were taken over later by DEC in the PDP-11/45. I'd be interested in knowing a little more about the history of that development.

I might point out some interesting ideas in this time

sharing system. One was the swapping drum. DEC put together, with the advice of Ed Fredkin, a drum [that] in one revolution could exchange the entire content of a field on the drum, put that in the memory of the computer and at the same time read the information in the program on to another field on the drum. We had DEC build two of these drum systems, one for Ed's machine at Bolt Beranek and Newman and one for the machine at MIT. At MIT, we started out with only 4,000 words of memory in our PDP-1, and how were we going to make a time sharing system with only 4,000 words of memory? What we did, is wrote a mere 500 word executive program which sat in the top piece of this 4,000 words, and did all the work associated with buffering teletype characters and doing the primitive scheduling of the machine. Later, we managed to get enough money together to buy three more banks of 4,000 words, so we could allow programs to be either 4,000, 8,000, 12,000, or 16,000 words in length.

A lot of the work in the connection with the TX-0 and the PDP-1 was done by groups of undergraduate and graduate students. The undergraduate and other hangers-on became known as the "hackers." Part of that started because I have a history, as an undergraduate at MIT, as being involved with the Tech Model Railroad Club. The Tech Model Railroad Club was an interesting place because it had this huge system built with telephone relays that

could manage the operation of the trains around the layout of the Club's 8 gauge layout. I thought the people over there might be interested in seeing what the computer facilities in the electrical engineering department at the research laboratory in electronics [were doing], so I invited a whole bunch of them to come over and see. When they saw what was there, they got really excited about it, and started making very good use of these computer facilities. A lot of that is what you will hear now from my colleagues on the panel. Thank you.

[APPLAUSE]

SL: Our next panelist is currently a professor of physics at Boston University, but thirty years ago he was working at BB&N and laid claim to the title of the best computer programmer in the world. This is Ed Fredkin.

EF: There was in the Hynes Auditorium, I believe, something called an Eastern Joint Computer Conference that I went to. I'd heard that there would be a machine there from Digital. Digital made modules in those days, so I went, and there was the prototype PDP-1. It was like a dream come true. It ran as fast as machines that cost millions of dollars, but it was priced at $120,000; actually $80,000 with one thousand words of 18-bit

memory. That's like 2K bytes; that was a lot of memory in those days. I was watching it, and I ended up right then and there doing my first bit of debugging on the machine, because about every three or four minutes the machine would suddenly stop, and the people there would look perplexed and start it up again, and say something like "Well, it's a new machine." So I started snooping around, and I discovered behind this machine was a large copper strap with a wire going to the machine. I followed this strap over to a booth, several booths down, that had a tape drive with these huge motors that went "punk" like this, and every time they would push the button to start the tape drive, the PDP-1 would stop. So asking a few questions, I discovered that these were the only two booths that requested a ground wire, so they supplied them each with a ground wire that just went from one booth to the other. Unhooking that fixed the problem. There I ran into Ben Gurley, whom I'd met before -- he is the designer of this machine, it was really a brilliant concept. I want you to understand how Digital came to the conclusion of what it was that was to be designed. This machine was just right. What were the specifications, I asked? What were you told? What was your charter? I knew that he was hired by Digital, at some point and was told to design a computer. He said he had the following conversation with Ken Olsen and Harlan Anderson who ran the company. "We would like to hire you to design us a

computer." Ben said, "What kind of a computer?" They said "Out of inventory." This was a company that financed itself with very little money, and there was this stock room, and he was to look through it and see what was there. The truth is, he had to design about half the module types that were used in the computer, but that was the general idea. It was really a brilliant conception.

I decided that we had to have one of those. I worked at BBN and I argued and politicked there and we bought the very first PDP-1 and had an enormous amount of fun with it. I wanted to mention that John McCarthy and Marvin Minsky hung around there, and this had a dramatic effect, I think, on the use of the PDP-1 and computers in general. All kinds of things were done. Marvin and I were reminiscing that ever since the PDP-1, we've been wanting a machine that could do what the PDP-1 did, which is write some simple code and say, "Here's an XY, put it up on the screen for me." All displays have gotten so sophisticated, you can't do that unless you're some kind gee whiz programmer. Someone should make that possible again, maybe give Marvin a machine like that and something new will come out of it. As was mentioned by Jack, the PDP-1 was one of the machines that pioneered time-sharing. The other, of course, was the 709 of the compatible time-sharing system. John McCarthy was at BBN

and one day he explained to me the idea of time sharing. He had the right idea. We worked at trying to make that happen and of course Digital helped us. We invented an interrupt system, and the drum system, and all those things. They all got built. Those were the days where you could say, "Hey, I need this new instruction. It's really terrific." They would go and wire it up and the machine would have a new instruction.

I'd like to tell you an anecdote which you don't know and which means that the PDP-1 is, in some sense, still alive. I started a little company and I located it in the Mill in Maynard [Massachusetts]. I had a PDP-1, naturally. One day I brought a friend of mine, John Cocke from IBM, to look at it. I showed him the machine and explained to him the architecture. Ben Gurley was there; he had left Digital and worked at my company. We had a great time. John Cocke [and Ben were] computer architects and we went through the machine in great detail, all the intricacies of its design. It was all really good stuff. Later on that week, I got a call from John and he wanted to tell me something funny. He said, "I was visiting my father. I described this Mill where your company was, and where Digital was, and my father said, 'Tell me more about that mill. It sounds familiar.'" So John kept describing it and then finally his father said, "I know that mill. The American Woolen

Company used to be there. I used to own it." So, John Cocke's father used to own the Mill that Digital got started in. Here's the interesting thing. Many years later, once late at night, I got a call from John Cocke. (IBM recently had a John Cocke celebration on a week-end. They invited people from all over the country and spent three days doing nothing but heaping honors on John Cocke. What I discovered is, that there are all these people, other than myself, who got these late night calls that lasted two hours or so while John Cocke told all his interesting ideas. I thought I was the only one until then.) John said, "Do you remember that machine you showed me in The mill in Maynard?" I said yes. He said [he had an idea]. He had worked up a design that was based on the PDP-1. It had a very simple instruction set like the PDP-1. PDP-1 had a multiply step and a divide step instead of an instruction. He'd come up with a modification of that, that did two bits at a time instead of one, and handled the sines properly instead of not handling, and little things like that. He had in mind that it would have a 16-bit word instead of 18. One other change he had in mind was the timing cycle: instead of being 5 microseconds it would be 5 nanoseconds. He had this all worked out in his mind. He had shown that this machine could emulate every model of the System 360 in faster than real time, so his idea was that they should build machines like that, and part of his idea was

to have a very clever compiler. That was different, but the architecture was the PDP-1. Of course that machine is alive and well; today, it's called the RS6000. It went through the 801 and a lot of evolutions, but that project got started right then and there, and it absolutely started with the PDP-1 as its idea. To me, the PDP-1 was one of the great experiences of all time, and I was always perplexed that they didn't sell thousands of them. I think they sold 120 or so. One day, I remember John [McCarthy] and I had this idea that this was such a great thing, we should convince Digital somehow, we'll go out and convince other people to buy these machines. The world took to the idea slowly, but to those of us who appreciated it, this was a miracle completely out of left field, there was nothing else like it anywhere in the world. I think the world owes quite a debt to Digital for this PDP-1, because it was the germ that started a tremendous number of things. It was the first really fun interactive computer. It was fantastic to sit at the terminal, have the scope there and make it make music and fancy pictures and started things that ended up being word processors, and real time control programs, and all kinds of things that hadn't been done before because the machines you needed cost millions of dollars. That was the future that burst on the scene and captured the imagination of a lot of people. It's a great story. Thank you.

SL: Our next panelist is currently an electronic consultant in California, but 30 years ago he was a research assistant with John McCarthy. He's going to tell us how he unleashed the curse of the video game upon the world with the PDP-1. Here's Steve Russell, known as Slug Russell.

SR: I did see the PDP-1 at the Eastern Joint Computer Conference, and thought it was interesting but it was awfully crowded. I didn't have a chance to play with it at the time. When Digital donated the PDP-1 to MIT I was down the hall, and came in to see it. I belonged to the Model Railroad Club. We talked about it a little. I thought it was a great thing, because it was the first "appliance" computer I had ever met, and probably one of the first "appliance" computers. [By that I mean] it had a switch, you could turn it on, you got a satisfying clunk and it started working. When you were done, if there was no one else using it, you turned it off and left. The other computers of my experience, if they got turned off, it was a major trauma, and five vacuum tubes burned out and field service had to come and dance around it for a while to make it work again. It was also a very satisfying machine, because you could type a single character at it and it would type a little message back you. Granted it was a little cryptic, but it gave you a

great feeling of power, much better than flipping switches. It had a cathode ray tube, so that you could draw pictures. I got a real itch to use it and started thinking about what I could with it. Marvin Minsky had written a little demonstration program that made interesting kaleidoscope-like patterns on the machine, and it was interactive. You could put in different starting numbers, and it would give you different patterns. It wasn't terribly exciting, because most of the time, when you put in new patterns, it didn't work very well. The play value wasn't really too good. I wanted to use it for something. We talked it around at the Model Railroad Club and the Hingham Institute, a dive on Hingham Street where some of us lived. We eventually decided that you could probably simulate space ships. We thought about that for a while, and eventually I got shamed into writing the code to maneuver a couple of space ships around the machine in two dimensions. I fixed it so two people could play. Then there was a great deal of kibbitzing about the speed at which it ran, and how there really ought to be gravity [for it] to be properly realistic. So Dan Edwards wrote some gravity routines and put them in. Then Pete Samson complained that the star map wasn't properly realistic, and he put in a star map. Shag [Graetz] grumbled about the explosion when he went into hyperspace. It seemed necessary to escape into hyperspace every now and then,

it helped make the game more playable. So Shag put in hyperspace and things went on from there. The whole process when we took a couple of months all told, but it was great fun, and it created a real problem because it made it a little hard to use them machine because you had all these Spacewar! players to kick off. It became necessary to publish a policy that Spacewar! was absolutely the lowest priority on the machine. Fortunately, debugging new versions of Spacewar! was higher priority. If you think playing video games is fun, writing video games is even more fun! That's how Spacewar! came to be. It appears that it was the first two-person interactive game played on a CRT. The people who made the most money off it were the patent lawyers who are still discussing it. I think a great deal of credit goes to John McCarthy and Marvin Minsky and Jack Dennis, who encouraged everybody around to try things and were not at all critical, and radiated a lot of enthusiam, which helped us try all sorts of things that eventually turned out to be interesting. I think the most significant thing was what people get in personal computers today: you sit down, you have something to do, you poke at the computer, it does something and you find out about it right away, no muss, no fuss, no bother, no wait. The PDP-1 really started programmers thinking about how to do that, and how to do it well. It took them about fifteen years to figure it out. I think it's

much better now.

SL: Our next speaker, picking up the thread from that, is currently a free-lance writer but back then he was also hanging around MIT as a Hingham Institute Fellow, thinking, of course, about Spacewar! and hyperspace. This is Shag Graetz. [J.M. Graetz]

SG: I'm glad Slug mentioned the lawyers. Earlier this week I was down in New York and New Jersey doing some software detective work, indirectly for the lawyers for a lawsuit that I call Spacewar East. The people at Magnavox who have a patent that they are using to insist they have the authority to license anyone who writes any kind of interactive game, video game, computer game, whatever, periodically get sued by other companies who think, this isn't quite right. In the first place, Magnavox's patent only came out in 1967 and of course the first thing that they adduce as evidence to the contrary is Spacewar! This has happened about three times over the last fifteen years. At various times, Slug, Marvin Minsky, John McKenzie, and others, get called and come and help out to see what we can do. It's still going on. The PDP-1 lives both in body and spirit, I guess. All I want to do is take a slightly different thread or maybe more than one thread, because it's not just people who were actively involved in electronics with Digital or

with MIT that got involved with this. That's only one of the ancestors of Spacewar! I probably have the least legitimate credentials of all the people here to be called a hacker. I did go to MIT. The first thread starts there because one of my first and still one of my closest friends was Wayne Wiitanen, who was a mathematics major and one of the few people [during the mid-fifties] when, at that time at MIT, the only digital computer that was doing anything, was Whirlwind. But like the TX-0 a year later, it was also available to people who wanted to do research. Wayne was one of those who got involved with Whirlwind. He was, as an undergraduate, a very early programmer. We became good friends most throughly activities in the Outing Club and our interest in science fiction. I flunked out and then I flunked out again. At the same time, Wayne left because if he had stayed he would have flunked out. One thing led to another and we went through our necessary six months in the Army. We back out, and we lived in a cooperative house for a little while, and then found our first, genuine, grown-up apartment, which was on Hingham Street in Cambridgeport, right by the river. We called it the Hingham Institute because it made us sound important. It was just a play thing, really. At that time, I was looking for something that was better to do than be a research assistant in a chemistry lab because that's all I really knew. I was out of work for a while, and it turned out that at

Harvard, where Wayne was working as a programmer at the Litaur Statistical Lab., they needed a machine operator, (comma) junior, which is the bottom of the totem pole. I got that job. It paid about 50 percent more than I could get as a trained chemist. That started that thread going. By that time we had become friends with Slug and we were all avid Outing Clubbers. We went hiking every chance we got. The other thing we did, every chance we got, was buy and read, or go down to the movie theatre and see, some of the out and out worst science fiction ever written and/or filmed. The novels of Edward E. Smith, all about the skylark of space and the great lensmen - 1930's pulp trash. It is tremendously exciting. If you pick one up now, even these days, it's very difficult to put it down if you let yourself go for just a few paragraphs, but it is bad! The same is true for all these science fictions movies that came out of Japan at the time from Toho Studios. We gave it the name Grade Z science fiction, which is certainly what it was. The best thing about it was the model work. They built these beautiful, intricate models of Tokyo, San Francisco, other places and all these marvelous rubber monsters, the ones you would be most familiar with would be Godzilla and Rodan, but there was a whole family of them. They also had space ship epics with model space ships, weren't quite as successful somehow. Probably because we knew more about all that kind of engineering

stuff than we did about monsters and Tokyo. But these two concepts got fused in our minds, and the first thing we thought of was, "Well, the obvious thing to do is they should make movies out of the Skylark of Space." But of course that would have required the kind of model work that didn't become feasible really until, probably until 2001. "Star Wars" is probably what the Skylark of Space would have been. It virtually had the same plot, really. But, at the same time, we were working, first at Harvard. Then I left Harvard Statistical Lab and went to work for another old friend, someone I had met when I was still an undergraduate at MIT, Professor Jack Dennis, who, at that time, was in RLE [Research Lab in Electronics] and was master of the TX-0. He hired me in the summer of 1961 to write a diagnostic program. This did two things: it gave me some employment for the summer, and it also set me on the path I was going to lead as a software writer. I wrote a diagnostic for the Potter tape unit, that was just then installed on the TX-0. When that was over, I went to work for a man that we affectionately called Zeus. His name is Douglas Ross. At that time, he was Director of the Electronic Systems Lab, which was in a nearby building. In the fall of that year, the PDP-1 arrived at MIT. We had been anticipating this for a number of months, and Wayne and Slug and I would get togther at Hingham Street and think about, what we could do to show the machine off. It always revolved around

this business of space ships. Between the Toho Studios and E.E. Smith, it was always on my mind - space ships moving around on a scope. It didn't take very long for us to come up with what turned out to be the basic rules of Spacewar! - two space ships opposing each other, firing something, rays or torpedoes, moving all around space and trying to blast the other out of the ether. We thought it was a good idea, but at the time we didn't have any immediate way to implement it. Wayne got called back up into the Army after the Berlin crisis. But as Slug pointed out, some of the hackers at the Artificial Intelligence Lab had got involved in the discussions. They were really quite eager to get going on it. Here was this machine, it had no software, [it was] just there - dying to be used. Slug kept making excuse after excuse, what was he going to do? Couldn't write anything. Alan Kotok went over to DEC, found a couple of mathematic subroutines. He came back with them, said, "Here they are. Write." So starting in January of 1962 Steve sat down, wrote the main control routine and like Tom Sawyer with the whitewash brush, got everybody else involved with writing pieces of the program. The first thing we had were the space ships and the torpedoes and the acceleration. Then Slug threw in a few random stars. It was really the display of DDT sitting in upper memory. [LAUGHTER]. Understand, we did all this is 4K, no more than 9 kilobytes. Peter Samson didn't like that, so he

wrote the Expensive Planetarium, which displayed the stars of the central belt around the equator in their respective magnitudes so that brighter ones looked brighter. Dan Edwards put in that star in the middle that had gravity, not particularly Newtonian, but everybody fell into the star if they didn't move. He also wrote a routine that compiled the space ship outline so that there wouldn't be any flicker on the screen. We were starting to exceed the refresh rate. I was working on hyperspace. Hyperspace was where we inhaled the Minskytron. You'll pardon me, that's what we always called it. Professor Minsky always called it the "tripost display" but we called it the Minskytron, because that's what it looked like. And I found a way to make that display, as the ship went into hyperspace. So out of these threads - MIT flunk-out, hacker-type mentalities, science fiction, monster movies, and something that was a computing equivalent of a Heath kit -- came together in 1962, and Spacewar! came out of that. It eventually showed up, I think, on every PDP that Digital ever produced, and is still kicking around all over the place. It must be on the VAX somewhere. That's how it all happened. The rest, if you want to blame us for it, go ahead. We don't mind.

[APPLAUSE].

SL: Next up is Dave Gross. Currently he's a consulting software engineer. Thirty years ago he was a student at MIT, who managed not to be around when the PDP-1 was delivered, but he returned soon afterwards to find our birthday machine sitting in the kluge room. Dave?

DG: I was sitting there trying to figure out what I was going to say and realizing that if this was a meeting of Hackers Anonymous, I would be up here saying, "I am a hacker." [LAUGHTER] That's what I was back then. As long as Shag's admitted it, I was another one who had to escape from MIT before they flunked me out. During that escape, the PDP-1 arrived. Before I left, a friend of mine, Bob Saunders, showed me this little glossy brochure from some obscure company whose name I can't recall now, but I think it was Digital. He showed me this brochure that had two computers that they were going to advertise for sale: one of them was the PDP-1. I said, "What's that?" He said, "That's a commercial version of the TX-0." The other one was the PDP-3, a 36-bit machine that Digital never actually built. As a matter of fact, we didn't even have a prototype of it. The PDP-1 was quite an advance over the TX-0, although it was obviously a close relative. The TX-0 required some training if you wanted to turn it on. There wasn't one "on" button and yes, you did get this satisfying clunk, but then you had to wait for the power supplies to warm up, because

they were powered by a vacuum tube system. To do this properly, you had to watch this little timer, and you sat back and you twiddled your thumbs for a couple of minutes while it warmed up. Then there was another button to push that started the clock generator, and if you didn't do it right, or maybe the button bounced, you might have gotten two. ..the clock generator was a chain of delay lines. If you didn't start it up right, you were in danger of overheating the circuits, I believe, because the two pulses in the chain, cycling the transistors more than they were supposed to. You weren't supposed to turn on that machine unless you were authorized, and I wasn't authorized. That really pained me. I really suffered. I knew I knew how to turn on that machine, but heaven help me if I ever did it, so I didn't. The PDP-1 was quite an advance. It had those magnetic resonance saturating transformers, the power supply that lasted all the way up to the PDP10 generation. Those were the power supplies with the super heavy transformers where if you opened the back doors of the later machines that required more power, the machine tended to tip over. The PDP-1 was where those power supplies started. Another advance in technology in the PDP-1 was that it had pulse amplifiers. The TX-0 had a delay line to generate the pulse chains, and then I believe the pulses would distribute the logic through plain transformers, which meant if you had to alter the circuitry somewhere and put

more load on the pulse, you'd be changing not only amplitude of that pulse, because you're loading the one transformer, but the change would reflect back through the transformer into the main delay line, and alter the amplitudes of pulses other than the one you had. The TX-0 had this oscilloscope over in the clock generator pack bay, with little circles on the screen marking where the tips of the pulses ought to be, and if they weren't there, the machine was in serious trouble. And heaven help you if anybody changed the gain on that scope, because it would really cause a disaster. The PDP-1, on the other hand, was a relatively reliable machine and it was really great for us hackers. My first stint in the computer room was as a witness to Bob Saunders writing this macro-assembler. It was a really fine product, probably as good an assembly language as you'll even see on a PDP-11 class machine. When I got back to the PDP-1, the work of converting that assembler from the TX-0 to the PDP-1, had already been completed. But I understand that Digital was first offering another assembler written for the one thousand word PDP-1, and it had to run there with no auxiliary storage. The original PDP-1 had a bidirectional paper tape reader. I've never seen a bidirectional paper tape reader. But it was a very fine mechanism. It had two pinch rollers and two brakes, one on each side of the reading head. The idea of this assembler was it was going to punch an intermediate tape.

I believe what you were supposed to do - you took the source tape and put it in the reader and it would read pass 1 normally, but then it was supposed to read the tape in reverse for pass 2.

EF: There was no room in memory to put anything other than the binary code you were assembling, so it read pass 1, and it turned out that if you read the tape backwards, every definition arrived just in time for its use. [LAUGHTER] By some miracle this worked perfectly with no auxiliary storage needed for definitions.

DG: Needless to say, we were not in love with this assembler.

EF: I thought it was fine. [LAUGHTER]

DG: The crew consisted of Samson, Saunders, and Kotok and a few others. They were given a challenge to do the work in one weekend. It was all night sessions. That wound up being the successful assembler for the PDP-1, too. It was a pretty neat job.

SR: It should be noted that the level of human engineering has still left some of us with the mnemonic for what to do with the switches when you get done, which was start to continue, continue to start. [LAUGHTER]

There were the start and continue buttons on the console, and unfortunately, the meaning of pushing them was reversed from what you wanted, which was what you had to remember. We've learned a lot, we've improved the human engineering of computers a lot. If you wonder what's been going on between the PDP-1 and the Macintosh, a great deal of it is learning the hard way - that start to continue to continue to start, and backward paper tape - isn't easy to remember and easy to explain, and getting rid of this nonsense piece by piece.

DG: Both the TX-0 and the PDP-1 were paper tape machines, but I believe Jack hired this grad student to do a magnetic tape controller for the TX-0. The grad student's [was] Gordon Bell, and he actually succeeded in devising a mag tape. Kotok and I were playing bridge one night in the TX-0 room, -- the TX-0 room was a great place to play bridge in the hot days of the summer because it was about the only room at MIT that was air conditioned for the computer, so it attracted us for more reasons than just the computer hacking -- and we looked at that magnetic tape and thought there ought to be some way to save that rewinding that paper tape, and rerunning it back into the assembler for a pass 2, there ought to be some way to record the paper tape on the magnetic tape during pass 1, and play it back on pass 2, and wouldn't it be neat if we could also put the resulting loadable

image on that magnetic tape, too. Between Kotok and myself, we came up with a patch for the assembler that would actually write source code onto the tape during pass 1 and leave lots of blank spaces between the blocks so that when it did pass 2 it could also write the resulting object code and load it from magnetic tape. That was a kluge that wasn't supposed to be able to be done, but we managed to get it working and it was a pretty neat system as such things went. Another hacker who was there to witness, was Peter Samson's music program. The TX-0 had a monaural hi-fit unit under the console and it was attached to probably bit 14 of the accumulator, or some random bit that was determined by experiment to make the best sounding noises when you ran the average program. In fact, you could tell when your program went into an infinite loop because the noise would go [MIMICS WHINING NOISE]. Your program was supposed to make [DIFFERENT NOISE HERE]. So it was a great tool. You could sit back and play bridge and not worry about the computer dying on you without your noticing. Peter Samson thought it would be pretty neat to use it to make music. He invented this game where he could compile a tune and play the melody on that speaker. When the PDP-1 arrived, he realized that that was a faster machine, with slightly better architecture, and he could actually code this up to play three voices at the same time. I remember when I got the first demonstration

after I returned to MIT, he had coded up a Bach trio sonata, which is very good for a music system that could play three voices. It starts off as a solo for the first voice, and then it gradually brings in the second and the third voice until you're playing some fugue and it goes on complicated variations. It was neat. He played it. [SINGS] Then came the second voice. I said, "Wow, two voices." Then the third voice. Then Pete did the hack of hacks, I think. He had very carefully timed the paper tape reading loop, so that it was a multiple of the sound incrementing loop. When the music reached the end of the buffer, it read the paper tape while continuing to play the music in tune, which is a pretty neat thing to do on a machine that had a five microsecond memory cycle. It would read the tape, and that had any number of people floored that it could do that, with just minimal buzzing in the background. When we finally got the successor machine to the PDP-1, which was supposed to be the PDP-6, and that machine was faster still, had the bigger computerware and Pete Samson said, "Ah ha, I can now do a six voice machine." Indeed, we designed something called the MK-6, which stood for the Music Kluge Six, which could connect your PDP-6 to a speaker and indeed he did have a six part music played on that machine.

Another thing developed on the PDP-1 that I think is significant is a product of a fellow named Dan Murphy

here in the audience, and Mr. TECO. I remember Dan debugging TECO, and thinking what a crazy thing to do. The right way to edit a paper tape is to use those Flexowriters. I'm going to admit it. I'm an old stick in the mud, and I still like the coding in machine language where a machine only had a two bit op code; I think that's what real programmers ought to do.

[APPLAUSE]

SL: Before we throw it open, I think I would be a shame not to hear, if they'd be willing, from some of our guests out here in the audience. I think particularly Drs. Minsky, McCarthy and Richard Greenblatt. Would you like to say a couple words? Marvin Minsky.

[END OF TAPE]

MM: You were talking about flunking out [of MIT] and most of the people who made the biggest innovations in this field did in fact flunk out. In fact, none of them flunked out they all disappeared right? I've never heard of anyone [at MIT] flunking out -- it happens, but the point is that it wasn't until 1970 that the professors could be said to know more then the students in computer science. They had a kind of artificial view, and I think this may be a real phenomenon. It certainly is why so many of these discoveries came from below; the image that Jack Dennis was head of RLE is a perfectly reasonable view from the bottom.

[LAUGHTER]

He's all the way up there at at assistant professor rank. Everything everyone said brought dozens of images [to mind], like going over to Hefron's and getting the right switches to play Spacewar! Eli's is still there but it's no use anymore because it only sells parts from computers. [LAUGHTER] There was a great junkyard which had every possible kind of surplus but it had surplus you could use, because if you're doing anything with computers you obviously don't want computer parts. One of the ironies there. There was a nice moment of banning Spacewar! just for a little while, of course. Many years later, some town, Braintree, or somewhere, banned arcade

games and I remember thinking "Oh, they thought of that, too." [LAUGHTER]

One of the great mysteries to me of this whole period was the disappearance of the graphic display. DEC started it pretty much. I was just mentioning to McCarthy it's really hard to believe the 704 was before the PDP-1 and I have this pretty well confused in my mind, but of course, the 704 was a bigger, more powerful computer but you couldn't program it much, and I don't think anybody's even mentioned Steve Russell's role in the programming of the first versions of LISP. He played an immensely important role in that. But that was mostly on the 'other computer'. I remember when we got the PDP-6 one day, about '64. It turned up, and I said, "Well now we've got our own real computer. How are we going to do our AI research on it with no LISP?" At the end of the next weekend there was. So a great deal of the success that I'm credited with, and John is in the AI lab, came from the fact that all this research was BP [Before Programming] and the same people who did these hacker exploits also did an immense amount of serious scientific work, mostly by figuring out what was going to be needed next year in AI. So when I said "Let's have a LISP," and they said "Well, it'll take at least three or four days." [LAUGHTER] There was this decay of graphics; the PDP-1 had this thing and you could, say, plot XY. The only

other machine that I've been able to plot XY on is some little Casio pocket computer which has a plot XY and it's Basic, but it's almost impossible to plot XY in any real computer. So in order to do graphics research, you have to get an awful lot of equipment set up and it'll go to an awful lot of trouble. Later I built one myself. I built a little computer for working with for children to be able to run LOGO programs. Unfortunately, it cost more then anybody thought, and schools couldn't afford it but it had a vector plotting scheme and that was great fun. Shortly after that I saw one in a little computer game named VECTREX which I guess has bit the dust. As soon as raster displays came out, graphics went down hill and took many years to recover. There are lots of other wonderful stories but there are too many...it's getting too late.

[APPLAUSE]

SL: Mr. McCarthy, would you like to say a couple of words here? John McCarthy.

McCARTHY: I don't have very much to add, to what has already been said. I would like to compare a little bit the psychology of that time with the psychology of computer use today. Maybe I am guessing here, because I don't really know too much about either one, but consider

that the PDP-1 as people said started with a 1K memory and went up to a 4K memory and then somewhat more, and now people are talking that one megabyte is rather small; Harvard is letting me use a machine with eight megabytes just as a terminal and so forth. Now, I have not actually learned to write programs 8,000 times as fast as I did many years ago, in fact I probably write somewhat slower then I did a number of years ago. While people are somewhat faster, they aren't certainly 8,000 times as fast as the people many years ago. The result is, of course, that people have to program at, so to speak, more like an executive level, where they don't really know what all these pieces that they are ordering around are. Like the boss of a company who doesn't fully understand what his henchmen are doing, or thinking, or going but he gives them some orders and hopes that the right thing will happen. Now it seems to me that a programmer is a kind of executive, putting together these parts. What is always amazing me, is people are saying "Well, three megabytes is not really enough. You need more," and of course the answer is if they really knew what was going on, and were in a position to change it, even three megabytes for many of these things would be plenty. I had something else I was going to say but I forgot what it was so I think I'll stop.

[APPLAUSE]

SL: Richard Greenblatt, one of the canonical hackers from that period, will say a couple of things then I will allow time for a couple of questions.

GREENBLAT: I was really a late comer to the PDP-1 scene. I arrived as a undergraduate, and never was an official user, but [here's] what I did. The sign up list went up on Friday morning at eight o'clock, and within an hour the entire week for the following week was signed up 24 hours a day. Users coming in and signing up. I would come along and look at the signup list and see who had signed up, and try to figure out who was usually late for their computer time, and I would then wait [until] the appointed hour and somebody would be a few minutes late showing up for their computer time and I would jump on the machine, and play around, and do my thing and then maybe, maybe, they wouldn't show up for their time at all or they would come in fifteen minutes late. That was computer access in those days. Later I was involved in a number of the things that have been alluded to.

[APPLAUSE]

MCCARTHY: The Sale system at Stanford will -- which is a was originallly a PDP-6 and went through various versions of PDP-10 and is a KL 10 and is of course thoroughly

obsolete but nevertheless still working, at least the new service works and so forth, will if it lasts, be turned off on June 8, of next year which will be its 25th anniversary. I've been thinking about how it should celebrate its demise, and it should send people a message, I think. If anyone wants to be sent the final message of the of the Sale computer just before it's turned off, then you should communicate. Now having sort of thought of that from an administrative point of view it's not instantly clear to me as to where you should send this E mail. Well, my poor secretary, I'll make her do it. [LAUGHTER] If you want to be sent E mail when Sale is turned off, send E mail to MPS at CS.Stanford.EDU and you'll get the final message out of the PDP-10, provided it doesn't crash irrevocably in the meantime because if it crashes in the meantime it's sort of been agreed that no heroic measures will be undertaken. It has asked for that. It has this living will.

SL: Would Ted Johnson like to come up here and tell us about selling the machine as DEC's VP of sales for twenty something years.

TJ: I hope I don't contaminate this hacker's delight by throwing in a commercial dimension, but I think I can add a few facts here and a different perspective. As Ed Fredkin pointed out, before the PDP-1, Digital was

entirely a modules company. You're looking at the sales force at the time that the PDP-1 was invented. I was the only person in the field. We had no field, only a person in sales officially, that is. Everybody was really selling, but I was the only sales engineer, and we had no field officers. I was in Maynard. One exception to the module business is that we also sold memory testers. Even though I was the only sales person out selling modules, they took me out of the field for about two weeks to design the first memory tester with the logic when after talking with RCA the company decided to build their own system. I didn't realize they were laying the groundwork for getting in the computer business, at all. Our total focus was the module business, and I really didn't know all the cards that Ken was planning to play, although I had seen the original business plan. I probably should have known better, but we were really focused. In April of '59, I happened to be in the office with Ken and Harlan Anderson when a request for a quotation came in. It was from the US Naval Ordinance Test Station in Pasadena, and there was a request for a 36-bit machine, five mega cycles, fit the module line perfectly, and Ken turned to me and he said, "This is just what I wanted to build. Go sell a computer." What's a computer? I didn't really know. After thinking about it, and talking to Ben Gurley who joined in June, the decision was made that if we were going into the computer

business, the 36-bit machine would take awhile not only to build but also to close the sale with the Navy, why don't we start small and build an 18-bit machine? By the way, we should probably build one in the middle someplace. I've never known whether that was supposed to be 24- or 27- bits but that's the reason why there was a gap. There was the PDP-1, the PDP -2, and the PDP-3. I believe the PDP-3 actually was built, by the way. We had a line of 10 megacycle modules and many, many years later I was showing that machine, over in Waltham, someplace in a basement by some drug company.

EF: They ordered one and DEC declined to build it. Actually two were ordered. AFCRL ordered one. They were delivered two PDP-1's, with some explanation that that was the.....[LAUGHTER] The other one was a company, an architectural firm, who really wanted it so they said, "Do you have any pieces of paper with any kind of design?" They took the design, bought the modules and built a machine, and made it work.

TJ: Ben Gurley joined in June and began designing and laying out the PDP-1. We introduced a working machine in November, so that was a tremendous time to market, and from our perspective selling modules, it was just a testimony to what you could really do with using our standard modules. The one competitor that we had the

first year that I recall was the CDC 160-A. I have a slightly interesting story there, too. I think most of you might know that Digital and CDC started off in the same month, same year - September of 1957 - both going off in the scientific engineering area but totally different strategies. My module selling was very difficult; it was difficult to find a customer for modules in those days, and I remember calling up CDC at one point and visiting them in Minnesota and trying to sell them on converting and using our modules to build their computer. Of course, they were much too far along and it was a silly idea, but I was naive and an eager salesperson. I did see in the side when they were building the 1604, and the 160A was the prototype, the test machine, that they built, and it was sort of a side line for them to go out and try to sell this machine. I think they were the first ones to make a deal, an OEM contract, for small machines. From then on the brass largely focused on the real selling of PDP-1's for a couple of years, and the sales force -- by now I was on the west coast -- continued to largely focus on selling modules. We did sell 49 PDP-1's. Sixteen of those were sold to ITT, who used them for message switching communications applications. Nick Mazzarese was brought in to be the account manager for ITT. That was really the start of the computer OEM business, and of course, later on in 1965 when we were into the product line

organization, Nick was put in charge of all the small computers. I think that's just some added perspective. Thank you.

[APPLAUSE]

EF: I'd just like to make a comment. When DEC got started, one ground rule they had was no software. That was for other people: You want this machine, you guys write the software. About selling the computers I remember that one day we had a PDP-1 at BBN and Harlan Anderson, who was a partner with Ken Olsen running the company, called me and asked me to go with him on a sales call for the PDP-1. I agreed. We went off to Ohio State University, and talked up the PDP-1, and it was clear that they weren't going to give an order right then and there. As we were coming back on the plane, Harlan Anderson grumbled to me, "That's the last time we'll ever do that." I said, "That's the last time you'll ever do what?" He said, "Call on someone to try and sell them a computer. If someone wants one of these machines they got to come here and buy it..."

[LAUGHTER]

SL: Before I throw it open for some questions, I actually have a question. There was one story which I determined

at least to be at least partially apocryphal. It concerned a little wire that some of the hackers hooked up between the TX-0 and the PDP-1 to play a prank on the professors - Minsky and McCarthy. [It involved] hooking up this computer, giving it the software to play chess, and one was in one room and the other was in the other room and they were actually playing against each other, thinking they were playing against the computer until one of them, I think as the story goes, McCarthy noticed that the moves were being really put on one letter at a time as a person would type it. He walked in the other room and discovered it. I think I eliminated Marvin Minsky from the story and then when I asked him about it..... Do you remember that at all?

MM: The part about me is correct. The conjecture was that I was playing against some human but it just seems to me I wasn't actually told against whom I was playing so I just assumed that it was one of the hackers. Is that true or not?

SL: I actually attributed the story to Samson, the way he told it, after I eliminated Minsky. He insisted the idea was that you were playing against the computer not another human.....

MM: Yes it was certainly billed that way.

SL: Do we have any questions from out here for any of our panelists?

AUDIENCE: Steve, tell me why weren't the torpedoes in Spacewar! never affected by gravity?

SR: Because it took too much time to calculate gravity for the torpedoes. An example of an important principle of PDP-1 programming, which was the dominating principle of PDP-8 programming, was that you compromise the problem until it fits on the machine you've got.

AUDIENCE: DEC sells a game book with all the original games that were on all the original PDPs and I think Spacewar! is published in that book .....

SR: That's the Dave Ahl book isn't it? Basic Computer Games?

AUDIENCE: One of the original books and I'm assuming that Spacewar! originally was not written _____.

SR: I used the most sophisticated language available at the time I started, which was Macro.

SG: Spacewar! was written in a language that you will

find in this particular book. There are two separate listings of Spacewar! The original final version and a version that was developed at MIT over the ensuing summer months and that came out in September. If you want to see what the kind of programming that we did, you're welcome to have a look at this afterwards. I'll open it to a few pages, and you'll get some idea of what went through our minds.

SR: I would like to point out with some unjustified pride that I was quizzed by the legal consultants a year or so ago about what Spacewar! did and I had commented it just barely well enough so that I could look at what was puzzling them and say, "Oh, yes that does," and explain it correctly.

[LAUGHTER]

DAN MURPHY: Do any of you know the origin of TECO?

SR: It's all his fault! The guy with the yellow shirt. He did it. He's the last guy to design the whole thing.

DM: Alright. I plead guilty. TECO was cooked up on the very same PDP-1 that we've all been talking about, on the second floor of Building 26.[DIGITAL MILL IN MAYNARD]....

SL: TECO, for the benefit of those who might not know was,

DM: It was a Text Editing program, one of the early ones that you could use interactively, and it had a couple of things that were advances over what seem to be kicking around at the time. When I first arrived at the PDP-1 there was a program out called the Expensive Typewriter, which you could in fact use to change your program using the computer and _____ you _____ the tape. It had two problems. One, you had to use it on the computer and secondly, you could only change the whole line of your program at the time, so if one letter was wrong you had to retype the whole thing. My motivation for writing TECO was to change both of those aspects, to let you be able to change one character at a time, but also to save what changes you wanted to make, off line, using the flexowriters in the next room. So when you got your precious little shot of machine time, you could go there, take your program, make the changes, and have a new tape punched out all in a very short order. It turns out, of course, that TECO was almost never used in that mode. We very quickly added a switch where you could enter the command while you were on the computer, and of course that's what everybody then did with it for all the years later that it was used. I want to tell one other story about TECO. Ever since your book came out [Hackers:

Heroes of the Industrial Revolution] I just wanted to point that some of the hackers in MIT in that era, not all of them, came from the Model Railroad Club. There were a few of us who came from the MIT radio station. I think this is significant because one's background tends to have some influence on the reasons that one conducts a certain task. With TECO I had a version of it that did just the very basic kinds of editing things, replacing characters and finding stuff and so forth and the next level of improvements to it came about this way. At the radio station we had a guy who was an interesting talent, and this story's a little embarrassing, but here's what it was. He was able to read copy like news reports and so forth in a simulated Chinese accent by virtue of merely exchanging all the L's and R's of the text and he could do this in real time. Like saying, Digital, "Digitar histoly recture series." So sitting around [a Chinese restaurant] one night we decided it would be really nice if a computer could change all the L's and R's and keep checks so that anybody could read in this pseudo-Chinese manner, so you go through and you look for the L's and you change it to something else. From that came the loop capabilities, the conditional capabilities, and several other things at TECO that turned out to make it rather useful.

SL: So we all owe search and replace in our

wordprocessors to that hack? Is that it?

MM: In fact I did both by programming in the sixties in TECO, and I still have this little bit of code which I think is the shortest description of the universal Turing machine ever written, about four little lines of TECO. But the last time I tried it it didn't work. I wonder if you have a up to date version of TECO running on anything.

DM: You may be interested to know, I had nothing to do with it, but some of the people in the VMS group had in fact produced a native version of TECO for VAX/VMS in the past few years. I don't know if it had enough [INAUDIBLE]

SR: For those of you who don't know TECO, you should understand that those who do, have some form of love/hate relationships with it. It is very powerful, and we've all used it to do something that we couldn't do any other way. But it's also powerfully mysterious when you start trying to do things that are complicated, and we've all been done-in by it more times then we can remember.

MM: It's good because it makes APL look so simple.

[LAUGHTER]

EF: I don't know what the ultimate TECO hack is, but I consider the macro written by Gosper that found the first glider gun by searching through every kind of configuration for the game of life. This was discovered by a TECO macro.

DG: For many years at DEC, TECO was an important CAD tool too; we couldn't have done the PDP-10 without it.

SR: And vital to Digital's engineering data processing because Dick Best's list was maintained in TECO.

AUDIENCE: Many years ago at DECUS, I think they were celebrating the tenth anniversary, there was a big trivia contest. One of the questions was 'what does TECO stand for?' What's the real answer?

DM: I do get asked that question occasionally and in fact it was 'Tape' because the only thing you could do on PDP-1 was take the tape or take the _____ and punch out a new one and so in fact the T for tape wound up in several places. DDT - DEC debugging tape.

SR: I guess we should also pay homage to another family of PDP-1 programs. We mentioned Expensive Typewriter, it was also Expensive Desk Calculator, and Expensive

Planetarium, and at Stanford when we had a PDP-1 we also briefly had Expensive Tape Recorder and Expensive Mirror. The idea was these things did the functions they said they did but they did them using a $120,000 worth of equipment which made them spectacularly more expensive then using the real item.

AUDIENCE: Was did Expensive Mirror do?

SR: It was a program that read the TV camera with excruciating slowness and displayed it on the CRT. [LAUGHTER] SMOP: meaning, Small Manner of Programming.

SL: There was one more question out there okay?

AUDIENCE: I was just going to mention that there was another interesting piece of equipment on the other PDP-1 1 at MIT at the nuclear science lab. That was about a 58K machine and DEC hadn't figured out all the problems with shipping things. What they did for adding more memory was bolt in an expanded cabinet in the back. Then you wired across the cabinet frames. When it got to twenty feet long it was 58K of core, it couldn't fit in the elevator so they actually had to hoist it up and enter through a window in order to deliver it. [INAUDIBLE]. It was a piece of equipment on it the IBM typewriter that.....

MAN: The Soroban Compu-typer.

MAN: We had an IBM ball typewriter. It's kind of a interesting characteristic. The program we had we needed to call the operator to input things - we were doing film scanning for high energy physics. The operators would be out of the room, so what they would do to call them when when they needed input. They would type out a message saying please input and wait a bit. If nothing happened it would switch the ribbon from black to red and then back again. That was the command you'd be given [INAUDIBLE] If nothing happened after a few seconds of doing this, and the operator didn't arrive, it would then shift the whole keyboard. The whole carriage would go grumf, grumf. You'd walk down the hall and here's this typewriter going like that...it was our signal for the operator.

SR: That reminds me of one of the things that we didn't mention about PDP-1 which I was noticing as I went over to the other building, is that they all came with a piece of do it yourself I/O in it. One of the things that you could do merely by purchasing a few extra modules was cause whatever you had in the room to get connected to a computer, and almost everybody used that for getting Spacewar! controls in, but it was also done for a lot of

lab equipment. Nobody else had ever even thought of telling you how to get your switches into the computer or get your signals out. It's just you plugged it into the official printer and that was all you needed to do.

EF: When the first production PDP-1 was delivered to BBN it had this kind of IO capability, and we had this scheduled event, and Ken Olsen was there and all kinds of important people. We had a ribbon cutting ceremony planned, except the computer actually did something that pulled that paper cutter down and it chopped its own ribbon once.

The greatest thing about the PDP-1 was the fact that when you had an idea -- this was the first time in the history of the world and this idea involved either programming or hardware ideas -- you could implement them in a few days and all kinds of very important things that used to be projects that took forever just a bunch of guys would go and make something happen. That illustrated that it could be done, and they'd do it by staying up a few nights in a row and have it done in a few days. It was just fantastic.

MCCARTHY: I want to mention something else that perhaps I should have mentioned before, and that was the PDP-1 was a machine that had a five microsecond cycle, yet it

it was possible to build time sharing systems on it that served a fair number of users better than many systems do today, in terms of promptness. Swapping drum, of course, was the key to it, in that in one drum revolution which was thirty milliseconds without latency - another piece of its elegance --it could swap one 4K user out and another 4K user in. So when we had it at Stanford, it could really give reasonable service on this very slow machine to twelve users. I remember when we started the LOTS, the Low Overhead TimeSharing at Stanford, which was in '75 and we used DEC-20, I had made calculations back in the early days which said that PDP-1 ought to be able to handle quite a lot of users if all they're doing is editing because here's how long it takes for an interrupt and here's how long it takes to put the character in the buffer. Then the question arose, well, Why is it that one DEC-20 is not enough to handle these students? Why is it so slow? I put a guy to working on it and he found out that if you were using the ancient editor then as it was written at Stanford a long time ago, then a typical editor user used one 250th of the DEC-20, which was pretty bad from the point of view of what ought to have been possible, but was plenty good from the point of view of our requirements. However, if he was using Emax, he was using 1/70th of the machine, which made him an average user, that is we were trying to serve 70 people on the machine. One of the other consequences of this

enormous increase in memory is that nobody knows today where the computer time is going.

SR: It goes into running all that code that fits in all the extra memory.

AUDIENCE: We've got these machines with all this memory so how many virtual PDP-1's are there? You guys love using PDP-1's they're color scopes, there are black and white scopes. Are there lots of emulators running around for this great machine?

[LAUGHTER]

SR: I don't think anyone claimed it ultimately great for now. It seemed great at the time. It was a paper tape machine and at Stanford the first time sharing system we built there we faithfully simulated paper tape on the drum. We then tried to use it for some undergraduate courses, and discovered that paper tape was pretty cumbersome to explain. It was a forty-six step process to get your program assembled when you wrote it all down and when you faithfully simulated it, the fact that you didn't have physical tape didn't make it a bit easier to understand at all. It really is nice to have a proper file system.

SL: On that note let's thank our panel and our guests for a terrific afternoon. And thank the PDP-1 for kicking off a year of interactive computing.

[END OF SIDE 2]