

AMERICAN FEDERATION OF INFORMATION PROCESSING SOCIETIES

HISTORY OF COMPUTING PROJECT

Chairman: Dr. W. F. Luebbert

DARTMOUTH TIMESHARE SYSTEM

TRANSCRIPTS OF 1974 National Computer Conference  
Pioneer Day Session

Transcript provided by Session Organizers:

Technical: Professor Tom Kurtz  
Dartmouth College

Administrative: Ms. Evelyn Bonney  
AFIPS  
(now Dartmouth College  
News Svc)



## Pioneer Day at NCC

"I think to start off I would ask each of you up here to identify him or herself as to what you were then and what you are now. Do you want to start, Nancy?"

N.B. "Thanks. I'm up here sort of misrepresenting, I never was a Dartmouth student. I worked as a part-time operator/consultant and probably more appropriately also /housemother. I'm still there. You could eliminate the first parameter of the job description?"

S.H. "My name is Steve Hobbs. What I currently am is a grad student at Carnegie-Mellon University. Back then we all went by then names of programs. Back then people would come up to me and say, 'Hey, DDT, how come such and such doesn't work?'. For example, I am sitting next to one of several people called BASIC. I also worked on the LISP system back then, and a variety of other various things."

N.W. "I'm Neil Wiedenhofer. I was a graduate student at Dartmouth back in those days. I worked mostly on the BASIC compiler. Now I'm a systems programmer for United Computing Systems, a time-sharing outfit."

J.M. "My name is John McGeachie, and my current job is to worry about things like getting people paid and making sure the accounting system runs. In those days I was a student and worked on part of the time-sharing system. In difference to what Steve Hobbs said, as I recall it, and I was there about a year before he was, we used to know our programs by the names of the people who wrote them."

D.M. "My name is Dave McGill. I am currently a teacher of transcendental meditation and systems analyst for Planning Research Corporation, and back in '64 to '68 I worked variously on the BASIC project, on 635, and on the time-sharing project on 635."

R.M. "My name is Ron Martin, currently with the Cyphernetics Corporation as Manager of telecommunications development. I've been there for five years. Back in '63 through '67 I was a student at Dartmouth and I worked together with Mike Busch in developing the first Datanet 30 executive. After that I worked on the MOLD system which was a developmental time-sharing system of historical interest only, and also on the first 635 time-sharing system. In both of the latter cases, on the communications executive."

K.O. "I'm Kevin O'Corman. These days I run programming in BASIC and of BASIC for a small computer outfit in New York. And in those days I guess I was ALGOL."

M.B. "I'm Mike Busch. I am presently in management at Computer Sciences Corporation. I've been involved extensively with CSC's INFONET(?) time-sharing network and more recently involved in facilities management for state and local government. Back then I was involved in writing the first master executive for the Datanet 30 and perhaps John McGeachie's principle adversary."

J.K. "My name is John Kemeny. In those days I worked with Tom Kurtz and was in charge of the time-sharing system over all. I wrote the first BASIC compiler. I won't mention my present job, but it's clearly not nearly as good as the one I had in those days."

D.L. "I'm Dick Lacy. Now I am a mathematician with the Defense Department doing programming I guess also. At the time I sort of inherited the 235 executive after John graduated while we were in transition between the 235 and in the 600 systems. I got out of Dartmouth in '67, so I didn't really see the 600 doing all that it does now. In some ways I would like to go back and really get some hands on experience with it."

T.K. "I'm Tom Kurtz. Looking around the panel here I think that Nancy and I are the only ones who are still doing the same things we did nine or ten years ago."

S.M. "My name is Sidney Marshall. I am currently a scientist at the Xerox Corporation. And I started out on the LGP 30 ..?., I think I was the first high school student to start stealing time from the college. I was not in on the original conception of the 235 system. I wrote TSAP and the MOLD system, and a good part of the exec for the current system, which I suspect has now been chipped away by succeeding programmers."

Dobbs "My name is Greg Dobbs. I am currently a research associate at MIT. Some of the other people that couldn't be here today like myself are not mathematicians. I'm in chemistry. It wasn't too uncommon at Kiewit for people from all types, pre-meds, and whatever to do systems programming. I inherited the editing systems for the 200 and in conjunction with Pete Neilson shepherded them through the MOLD system on to the 600."

J.W. "My name is Jerry Weiner. I'm a new product manager at Pitney-Bowes. I'm the first one in industry that ever believed what Kemeny and Kurtz said could be done. I was the GE project manager. My function was flying in spare parts when they burned them out. Although the Datanet 30 was standard off the shelf, it was standard off the shelf numbered double zero and one of ten unique machines. We rebuilt it without telling other people, things of that type."

G.D. "My name is George Detlefsen. I work at the General Electric R & D Center in Schenectady. My participation on the 265 was to build the background FORTRAN system which enabled the people with the bigger jobs to work in background."

"I'd like to start things off by asking a question. Does anybody here really remember what happened on May 1, 1964?"

"I think that you people and pioneers are missing the one crucial element of Dartmouth Time-Sharing System and that's the user. I tried myself participating in Thomas Kurtz's class in 1964, in the spring of 1964, and we started out in that class using BASIC on cards on a GE 225 in the basement of College Hall. And then we moved from there to three terminals in the basement of College Hall and Mr. Busch and Mr. McGeachie...?..."

J.K. "Let me at least try to answer that I can swear that this is true. But I think the goal we set ourselves was to have a program in BASIC work simultaneously on two terminals. I think that's what happened early in the morning of May 1 and it probably was adding two plus two and wasn't ..?..."

M.B. "If my recollection was right, I don't remember the specific incident, but it just had to be prime numbers by the sieve method."

J.K. "At any rate, there was a relatively simple program in BASIC that both the BASIC worked and time-sharing worked in that we got the right answer, the program to run on two terminals at the same time. Actually you may think that's a very small achievement, but once we got to that point things went terribly fast after that. From that going to a larger number of terminals and having most of BASIC work really took much less time than that major breakthrough."

J.M. "Let me comment on how fast things went. We went from Phase 0 to Phase 4 in nine months."

J.K. "Just like the Federal Government. I think that since Tom is chairing this panel, I would like to make one remark about what really happened. A key event occurred really in the academic year '62-'63. The serious planning started in the spring of '63 and a year and a half later there was a multi-terminal system in operation, and those you see here had a very major role to play with it, but how it really started I think goes back earlier than that. Namely, the fact that Tom Kurtz had the ridiculous dream. He had the idea that it would be possible to come up with a computing system that every student at Dartmouth would have a chance how to learn computing. Furthermore, he had a dream to make a computing system available the way an open stack library is available. The earlier panel has talked about that, but that is a terribly radical idea as recently as eleven years ago. But at most academic institutions it is still an unheard of idea. I think as we talk about some of the details, one should not forget the fact that Tom really had those two absolutely key ideas that revolutionized educational uses of computer. .. The only other thing I remember about those days was that it was very good that both he and I were quite ignorant about computing, because we did not know that what we were about to attempt to do was impossible."

T.K. "John, didn't you invent a particular way of solving problems or disputes when there were disputes or disagreements between the student programmers? Didn't you say once that they could solve the large problems but you reserve the right to solve the small problems? I remember something like that. Am I the only one who remembers that?"

D.L. "Dartmouth was very selective in hiring computer operators at the time. I remember handing Professor Kemeny decks of cards to put through the card reader and then everybody ran over to the printer to see if BASIC was doing what BASIC was supposed to be doing. This was before the time-sharing system was running but he was debugging and we were I guess helping him."

M.B. "Does anybody remember the one card wonders?"

J.K. "Yes. It was quite crucial, of course, because before you got time-sharing going you have to get into the system basically with cards, and we had a number of single card things. It was a competition on how much you could make one single card do. And Mike, do you remember some of them? I remember the great competition things everybody said you couldn't possibly get all on one card."

S.M. "Actually, that was the one card wonder, one card wonder, maker, maker. Straighten that out."

J.K. "Well, since we're reminiscing, let me mention one thing about the early days of BASIC. We only had one big argument on BASIC. The idea sounded very attractive but coming up with a brand new computer language when FORTRAN was clearly going to be the only language ever used in computer, many people argued was a crazy thing to do and that we might be leading our students down the wrong path in inventing a new language when FORTRAN was clearly going to be the only language ever used in computers, many people argued was a crazy thing to do and that we might be leading our students down the wrong path in inventing a new language where when they graduated from Dartmouth it would not be usable on any other system except our own. I think ten years later it's safe to say we won that particular gamble. I'm trying to estimate how close we are to a billion jobs being run in BASIC. But just from the statistics we saw, about fifty million of them must have been run by now at Dartmouth College and given all the other places they use we probably can claim that more than a billion jobs have been run in BASIC all over the world."

T.K. "Well, I claim at least ten thousand myself."

J.W. "GE did a survey of all its major programming managers and the conclusion was that BASIC had absolutely nothing to offer that wasn't done better in FORTRAN. And I said you missed the point completely. It's a simple system for the users."

T.K. "Well, getting down to some more amusing things, I seem to remember things like wastebasket fires in the basement of a most un-fireproof building as being one of the hazards. Does anybody remember wastebasket fires? I guess those people are not here."

N.B. "Yeah, I think I remember. I remember also one of our students that one time, only once, I saw actually put a cigarette into an ashtray. Unfortunately, the next thing he did was set a notebook on top of it."

S.H. "It's interesting to see how much student assistants' techniques have changed over the years. I remember when I first wrote my first system, it was time to debug it. I'd sit by the console typewriter and look at the lights. When something that looked like my program swapped into memory I'd hit the carriage return that'd finish off the Dump command and dump all the memory on the printers to see if I could figure out what was going on. If that wasn't good enough, I went to the D-30 and flipped the switch that prevented that jump the boot instruction from being loaded into memory. When I came into memory, I'd hit manual and single step my program to see what it was doing. Much to the -- very upsetting to the people upstairs in the terminal room, because while I was single stepping their program time-sharing had definitely stopped. Nowadays they use fancy debuggers with symbol tables supplied by assemblers and multiple debugging. You can single step your program while everyone else runs. It's a real change."

G.D. "Along this line debugging the format statement processor for this FORTRAN system, I wrote a routine which did a trace by entering interrupt mode, printing out the contents of all the registers, setting an interrupt and exiting interrupt mode, and this feeded out paper for several hours before I found the error. But that was one of the nice things you could do about ..?.."

R.M. "As I remember the first error you found was that you've got to turn off the interrupts on the line printer."

J.K. "I really think that the memory that sticks in my mind is not things like fires in wastepaper baskets, but seeing Busch and McGeachie at the two machines. One of the very bit problems was, I think it was, the first attempt of trying to have two different computers working in parallel and cooperating with each other. And the two of them took this terribly personally. It wasn't John's machine and Mike's machine; it was John and Mike who were not responding. And they would stand at opposite ends of the room and yell at each other at the top of their voices, and how they became lifelong friends in spite of that is something I will never understand. I wish you would tell us a little bit about the experience of debugging that two part executive."

J.M. "I have a story about that; one of our long lasting quarrels. I got so sick and tired, I said well I'm going to fix it and so I put a couple of messages into the 235 executive. One message said "Busch did it." The other message said "I did it." I figured the message that said "I did it" would never come out. So I went away and the next day this particular thing happened, and I was very embarrassed to see the message "I did it." appear on the console."

R.M. "In the early days of development before the 235 exec was working, the Datanet 30 exec was checked out with a little dummy routine that just sat in the 235 and responded with the right responses without doing any of the computing or processing. This was the basis for many of the arguments that ensued, because we were absolutely sure that Datanet 30 couldn't possibly be wrong, because it always responded right with the simulator. Of course, completely ignoring the fact that the simulator might be wrong. I believe that was the fuel that convinced Mike that he was right, and John had his own reasons for thinking that he was."

J.M. "I think our approaches to a system reliability was sort of interesting too. I recall one time in the early days we were developing the system, it didn't matter very much when you made changes because the new versions were not really that much better than the old and they all had bugs. Well, after the system had been running for about six months, people started to count on this, and it became a little harder to put in new versions of the executive. Mike and I developed this technique of appearing about two in the morning, putting in our new binary decks, taking the old decks and throwing them away, thus making sure that the next day there was no way to back up. Unfortunately, we did this once when President Kemeny had a demonstration scheduled."

T.K. "Nobody is going to tell what happened."

J.M. "I don't know. All I remember is getting up very early in the morning and driving into Hanover frantically from where we were staying that summer and trying to fix things. I have no recollection of whether it ever worked again or not."

R.M. "I have a little story I would like to tell. There was always an immense feeling of power when you worked in the basement of College Hall. You knew those people were upstairs pounding away at the terminals and you held their very life in your hands, essentially. Along the way I had developed a little program which is well remembered, I'm sure, by anybody who was there. It was a program which generated random sentences. The subject matter of the sentences was completely obscene. The interesting thing was that it wasn't enough that we were amused by the program, but at one point we even went and printed out reams of line printer paper to test out the first time-sharing compatible batch BASIC. We used to drive down to the Polka Dot or the Four Aces, and we would sit there going through these reams of paper and looking for something that was more than just random, that was considerably creative. I remember the look on some of the waitresses faces while we were doing that. We got kind of tired of that and found a new technique. We would sit at one of the control terminals of the system, change a few internal system pointers, and pawn our program off on an unsuspecting user. The only thing was we didn't want to waste the time to look at who it was, so we never knew who the victims were. What would happen from the users point of view is that they would be typing in their program to solve a differential equation or whatever and finally they would be done, and they would say RUN. Somewhere along the line I had slipped them this other



program in the place of the one they were creating and instead of the output they expected they would get these strings of random sentences. This story has kind of an interesting kicker to it. Several years later when we had a dedication ceremony for Kiewit, the new computation center, I remember that Dr. Kreider was sitting up on the stage reminiscing at that point about some of his early experiences as a user, and I was in the audience with a couple of other people that remembered the incident, and I remember he made some remark about how he sat down at his terminal and typed RUN and how all this stuff came out of his terminal. And he never knew who did it, and I don't know that until just now that he ever did know who did it. The cat's out of the bag."

Hobbs "I think that people are cutting the reliability of the system short. I got there in '65 so I'm not sure if I'm a real pioneer. When I got there, one year later, in September of '65 it was completely reliable. As far as I could see, they brought up experimental time-sharing on Sunday and no one could tell the difference that it was experimental. It seemed to work just as well as the regular days. During one experimental session, just to let people know it was really experimental, we went to the character translation tables that take internal BCD and translate it to the teletype and switched the entries for the digits 2 and 3. Made the BASIC program for I=10 to 100 go 10,11,13,12,14,15,16,17,18,19,30,31,33,32... This is no problem to a systems programmer, but at least let the users know it was experimental time-sharing. Cause it really was when I got there extremely reliable. I was amazed. I had had some experience with CTSS at MIT and in reliability it was much better. I was surprised that such a small system with relatively poor hardware was that reliable."

R.M. "I remember one thing about that, if you didn't already have the program already typed in, you had the problem of being creative and typing your lines to either avoid line numbers containing digits 2 or 3 or to use them properly. And properly is not an easy concept in this case. Because clearly if the END statement wound up being sorted before the end of the program it wasn't going to run, and other such similar problems ... you could have for statements after your next statements and so forth."

S.M. "I'd like to mention the first person to patch that program changed the translation table from 2 to 3 and then discovered that he had no 2's to put into the system to put the other patch in."

R.M. "That was a serious problem, but we did it by using a third digit and an intermediate step."

T.K. "And then there was the case where a slight modification in the program was made so that on March 15 all output came out in Roman Numerals."

J.M. "Tom, I think I have a sequel to your story. One of our very ablest student programmers who graduated a couple of years ago finished the new version of BASIC and he finished it sort of about the time he graduated. Nine months after this gentleman left, March 15, everything started appearing in Roman Numerals."

"When did you leave Nancy?"

N.B. "I never saw time-sharing until I'd been there about four months. But at that point we started.. I was the only operator and I was there during the day and then all our student programmers were assigned hours during the evening when the system was up where they were the official operator. I think that from looking at this crew you can imagine how ten years ago really how often they showed up when they were supposed to be there. Luckily the system was sufficiently reliable and there were pretty quickly enough knowledgeable people on campus that if you happened to be using the system and it stopped, there was one very common symptom and everygody knew what it was. So you went down to the basement of College Hall, unlocked the door with a nail file or your I.D. card, whatever you happened to have handy, walked over to the disc, pushed one button, everything started up, and you went home. And the system sort of ran that way, completely unattended over the weekends, and pretty much generally unattended."

T.K. "I remember one time I got a call at night, it was actually quite late at night and for some reason the system which would normally run, and run, and run, it went down on one of those rare occasions and they needed somebody to come in and push that little red button on the disc. But the person who called didn't live in Hanover. As a matter of fact, he lived in Pittsburgh, PA and he called me because I was the only one he knew the phone number of and said the system went down. And I said, 'Jerry, do you know what time it is? It's 3:00 a.m. in the morning. I'll be damned if I'm going to get up and go down and push that button.'"

N.B. "Speaking of phone calls and things, one of probably the earliest programs in the program library was FTBALL. A lot of us here know who wrote it. I remember one night we had been apparently having some problem somewhere along the line and I got a phone call at home one night to tell me in absolute panic, 'FTBALL has been clobbered in the library!' I live a good half hour away from Dartmouth and I really wasn't about to jump in the car and trot in to reload it from cards, or paper tape was probably what we had for backup at that point. So I just sort of sleepily said 'Well, I'll put it back in the morning.' What did you really expect me to do."

J.K. "We were probably trying to recruit a new football coach and that seemed so terribly important. I don't remember who wrote it but I do remember when that program was written. It was written on a Sunday. Those were the good days when time-sharing still ran on Sundays. It was written on Sunday after a certain Dartmouth-Princeton game in 1965 when Dartmouth won the Lambert trophy. It's sort of a commemorative program."

R.M. "There was also an earlier program for baseball, which is well

remembered. I was just wondering, would somebody like to tell the jolly green giant story?"

G.D. "In the summer of '66 I got hired by GE research lab to make a Chinese copy of the 265 software and install it down there. And at that time GE itself had written a FORTRAN compiler to run on a 265 system for use with the commercial time-sharing service. But the time-sharing service people would not let the research lab have a copy of this program. So when I got the 265 installed and their software running at the lab one afternoon, I didn't have anything to do, so I said 'Well, I'll try this FORTRAN.' So I got out a manual and tried it. Ah-ha! With this line in the beginning of your program you can print the object code. So printing the object code, I pretty soon discovered that, yeah, with the right sequence of instructions, you're into an array. At that point I could write a disc read routine. So I wrote a discread routine which read the FORTRAN compiler off the disc, encoded it in a suitable form for transmission back to a teletype and sat and punched it out. It took about three teletypes for eight hours to punch the thing out. And then we read it in to our 265, converted it to a deck, and were running it quite normally. We had a program which we would run on the commercial time-sharing system once a week to discover whether there were any patches put in the programs, so we put them in too. And this went on for... we were merrily running this thing for about two months when my boss's boss goes to the time-sharing headquarters in Phoenix at that time and says, 'Well, our 265 is up and running. We even have FORTRAN.' 'Who gave it to you?! Where did you steal it? Who's the leak?' 'Very simple. One of our people wrote a FORTRAN program to copy it off the disc.' 'Don't believe it. Impossible. Can't be done.' So he came back a few days later and related the story. So we said, 'Well, sure it's quite possible. If we can read the disc we can write the disc.' So we proceeded to write a patch to the 235 exec onto the local time-sharing service in Schenectady which would type out a message once a minute. 'Veni, vidi, vici.' We waited half an hour and no response, so we decided well we'll try New York City IPC. The message was 'If you don't believe this, call Dan Scott at DialCom so-and-so'. We waited half an hour and saw no response so we decided we'll try a strike in Phoenix. So we put one in there which said, 'The jolly green giant strides through the valley of the sun.' No response. This was midday. So then I called up the trouble number for the Schenectady IPC and I said 'Is time-sharing running?' They said yes. 'Are you sure about that? Will you go and look?' They came back. Fine. OK. 'Well, would you tell me what's coming out on the 235 typewriter. The man goes away and there's a silence for ten minutes, and the phone gets put back on the hook. A similar thing happened when I called the New York center. So I called up Phoenix and I got a very agitated person on the phone. 'Who's this? Who's this?' and I said 'The jolly green giant.' Ha! Ha! Ha! In half an hour we had a phone call back saying 'You and the other fellow who did it, all expenses to Phoenix. Show us how you did it and we'll give you the FORTRAN compiler.'"

J.W. "In November 1964 in Phoenix my manager told me I had 90 days to completely turn off the Dartmouth experiment. 87 days later Dan Scott and I had gotten paying customers on the system and on Sunday at 5:00 I became unemployed."

K.O. "I recall that Phoenix suffered a number of minor casualties like that. In the ALGOL compiler for a long time, there was a way of making fixes on line. We put it in in a way that made it very hard for the students to find out how to do it, because using that, of course, you remember that thing about stuffing jumps to the boot strap program. Well, you could have done that yourself. And all of those variations could have been done. It's very odd. I never did figure out why no student never found that one. But it was very useful to us when we wanted to irritate the people in Phoenix who occasionally had irritated us back. We could call them up over the telephone lines that they so kindly were providing us with, and we could kill their system in a way that would cause the typewriter to type out almost anything we liked. And we would sit there, usually in the middle of the night, and compose little memos to them in their error messages, I guess you'd call it. That was kind of a favorite pastime of ours in various different ways. During the early days the computer center was used a lot for strict batch processing. Card-oriented stuff. No time-sharing going on, so the Datanet 30 wasn't being used at all. Ron wrote a program in the Datanet 30 that would just sit there and waste time and display something pretty in the lights. Until somebody went in there and modified it to use that feature of writing anything it liked into the 235. Only in this case, if I remember right, it waited fifteen minutes to do it and then what it wrote was in the wrong parity so the 235 came to a screeching halt."

R.M. "Actually it waited ten minutes because it was called 'ten minutes to live' and it was written especially for a very particular faculty member at the engineering school, who was not well enjoyed by the students, and he would come in and hog the machine during our development time for running his background free-standing FORTRAN programs. In order to discourage him from getting too much work done, we set about writing this program. He had seen the lights flash on a number of occasions before and never paid any attention to them. In this particular case we set the thing going and he didn't pay any attention and we went out and left. Came back a while later and he was still trying to find out why he couldn't start it from scratch or do anything on the 235. It was absolutely dead. Another little thing ...you saw before about being evasive and jumping around and watching when the boot strap was catching up to you. There was another trick that got pulled once that was similar but even worse. Rather than sitting there and trying to evade this captor that was trying to catch you, you can play another game. You can listen to all the requests in the mailbox and put back all the proper responses but never do anything about them. Just make believe you're doing the things. And then you can use all the rest of your time to do whatever you want. Meanwhile, the Datanet 30 would think that it was sharing time for everybody in the world. It would never get upset because you kept responding. It would never try to wipe you out."

Hobbs "I remember one of the first things I ever did when I was learning about the D-30. I put a brand new command in. They had old, new, run, save, .. I put a command in called zip. Zip looked just like any other illegal command. It typed WHAT? just like any illegal command did, but it guaranteed that the next person swapped in was you. It was extremely handy for the system programmers. They finally had to disable it. Two system programmers who really wanted to get something done would just alternate. One would come in. The next one would come in. And nobody else would get anything done. The system had something. If you did nothing for ten minutes they disconnected you, it assumed you were a dead terminal. Some poor person was running a relatively short program and after ten minutes of waiting for something to happen he got disconnected because he never even swapped in for ten straight minutes."

N.B. "There were actually other ways of causing the same thing. One problem was once we had the terminals upstairs, we still had a couple upstairs in the machine room and those were the ones that the programmers normally used. One fellow is here who one day was working, just doing some routine type work on the control teletype and it got to be time for him to rush off to class. He was in sort of a hurry and didn't think too much, and instead of typing GOO or BYE to sign off he typed OFF. Which leaves the control line working just fine, but no other users get anywhere. Kevin, Steve Garland suggested you should tell the Dartmouth ALGOL story."

K.O "All right. The compiler for ALGOL at the top of its output no matter what happened when you tried to do a run would type out just the phrase Dartmouth ALGOL and then would follow either output or program listing or whatever it was that you had asked for. Somewhere along the line Phoenix, the GE people in Phoenix asked for our whole system including ALGOL. We were worried at the time because we didn't like some of the ways the Phoenix people had been acting like they were going to take the system and make it theirs. We did in fact take things very personally. This was our system and it was not about to become their system suddenly. So, in the ALGOL compiler we went to considerable pains to hide the code to put out the phrase Dartmouth ALGOL. It didn't appear as text anywhere in the compiler. It was in there all right, but it didn't appear as text. And what's more the stuff that put it out was disguised in the middle of several places. In one place it was actually in the middle of the routine that loaded it on to the disc modified everything to link it all together. Routines that actually put it out looked like they were - I think the comment on them said something about padding the symbol cellar. So that anybody looking at the listing couldn't possibly discover what was doing all that, and sure enough about a month and a half after the first version of ALGOL got out there, we got a call asking, 'How in the world is that thing put out?' Sure enough, they had wanted to take it out."

Hobbs "It's interesting to look at the original manuals that came out. They came up with their original manuals for the 265 system and there was the General Electric BASIC manual, and the Dartmouth ALGOL manual. They never found out how to keep ALGOL from typing Dartmouth ALGOL."

G.D. "The manual titles are an even more interesting progression because the original ones for the commercial system were Dartmouth/GE BASIC. Then it went GE/Dartmouth. Then they dropped the Dartmouth off."

J.W. "All of that took place after mid-February, 1965."

S.H. "That keeping of the Dartmouth ALGOL a secret was even from the person who followed Mr. O'Gorman, Sarr Blumsen, who isn't here. He had no idea how Dartmouth ALGOL was being put out until we said gee, Here's a subroutine that the ALGOL compiler doesn't need. And he erased it. He loaded up this new ALGOL system and it typed out !?\* 'Ah. Now I know why it's necessary.' But the following student had no idea where the words Dartmouth ALGOL came out from until he stumbled on it by accident."

S.M. "I'd like to tell a story about some of these early days. You can sort of tell that we were sort of misguided high school kids going to college. One of the games we would play would be to go downstairs and there was a control teletype that could talk to any other teletype. One of the features of this was that the other person didn't know that this had happened to him so that suddenly he was talking to you rather than to whatever he thought was going on. And you'd generally go down there just before a football game or something when everybody was up there trying to impress their dates with the computer. You'd take control of the computer so they'd be talking to you. You got some very interesting conversations. One I remember was ...one of the games was to type as fast as possible so that they thought it was the computer rather than some stupid person downstairs. And somebody realized that there was a little more intelligence in this terminal than they thought and started asking questions. One of the questions they asked was 'What's the score of the football game going to be?' I typed something very fast. 14 to 7. And I was right! And there was a big argument after the game up in that room. 'The computer can predict things!' 'No, it can't!' I never did find out who it was. I just heard reports of it."

N.B. "Incidentally, while Len Smith and Sidney are exchanging here, I don't think Len went into too great detail when he was talking about all the things that you can learn about the system by typing EXPLAIN. There are some things they chose not to explain, but there are others that.. I believe that if you type EXPLAIN JGK you still get a portrait. However, if you type EXPLAIN SIDNEY it still says "There is no explanation for Sidney."

T.K. "I remember the time, I think it was in the day when we were in the computer room in the basement there, and this official looking fellow, I guess he was a fireman, came in and he had a radio in his hand. He was very upset because he said that he had been detecting unauthorized emissions on the official fire department channel. He'd been going all over town trying to figure out who was putting out these signals and he traced it to the computer room. He was right of course. It was a one micro second machine so I think it had a strong signal in the meg cycle, wave length, or something of that sort. We didn't really know what to do so we struck a compromise, that is if there ever was a fire and they had to use their emergency radio, they would call us up and tell us to turn off the computer and we would do it."

S.M. "We discovered that we could use that noise on an ordinary radio to make music. Put a little radio by the console, take control of the time sharing system. We never thought too much about users then. We had something to do. You'd always go up there and there would be a puzzled look. Gave one a great sense of power, but looking back on it, it probably wasn't too mature."

J.K. "I think one thing we haven't talked about yet is the effect users had on the development of the system. I would like to tell one story. When the system was first up on the air, my son was in grade school, a fairly early grade. Therefore, I was held up for giving a tour to his class through the computing center, and I think you saw some of the pictures. We were, of course, in the oldest building and the equipment was down in the basement, but the teletypes were upstairs. I gave a demonstration on a Model 33 teletype. I did a couple of programs that grade school kids would enjoy and some sat down and typed things in and it was very exciting. Then as a real treat I took them downstairs to show them the computer. My son's teacher called me up the next day and said you might like to know what happened in show-and-tell. You know, they have to go the next day and explain what their field trip was about. She said it was really amazing. All the kids spoke up, they all enjoyed it very much. Then she asked, 'Well, did you have any questions that didn't get answered?' One young lady raised her hand and said, 'Well, I understood perfectly everything Mr. Kemeny did on the computer.' And she summarized some of the things. 'But then he took us downstairs into the basement and showed us a great big box that looked like a refrigerator. I never did find out what it was.' I mention that as an example of user influencing the system because for awhile I told that purely as a joke, but it eventually changed my mind totally as to what the computer is once you go into a time sharing system. I mean, ask yourself what you mean by a telephone."

S.H. "It's amazing that that inconsistency carried on to Dartmouth students. They are given this BASIC manual as they sit down at the terminal and push the button and log in. We had some poor guy who came to College Hall where all the terminals were and the computer had a disc crash. It was down and there was a big sign "The computer doesn't work today." The guy went over to another classroom building and sat down and said how come this computer over here doesn't work? Everyone was convinced that those little grey boxes--each grey box was a computer. The BASIC manual gave the students originally that opinion. I think part of it's done

by keeping the computer in the basement of College Hall. Nowadays, the computer is behind the big glass windows which you can see when you get to the terminal room. Hopefully...Dartmouth students still don't know that terminals are not the computer?"

N.W. "Do they still ask for the big computer when they want the 35 teletype that has the paper tape instead of little 35, that little computer which is a 35 without the paper tape."

"That was a tough decision for you people to go the way nobody else went. Why didn't you stay safe with FORTRAN, etc.?"

J.K. "I just had a very nice compliment paid--that I've never played safe in my life. I think that has to be answered in two parts. I think we have to answer it both for time sharing and for BASIC, and the answer is much simpler on time sharing because it simply was totally impossible to teach hundreds of students each year on the batch processing system. I used to tell, and I did actually say this to the Board of Trustees when they made a presentation, that I had seen at the Rand Corporation, where I was a consultant, some of the world's most famous scientists stand in line for an hour to get five seconds of computing time. I could see such famous scientists putting up with it, but Dartmouth undergraduates would never put up with that kind of system. So the decision to time sharing was a very hard one, but was totally unavoidable. I don't think we had any doubt. We had to go to time sharing to do that job. So that we never doubled about. On BASIC it was a quite different kind of question, and maybe this gives me a chance to add just one thing to the very good presentation on BASIC that was done. We certainly never conceived of it as getting to be as large a language as it has gotten to be, but we did think of it from the beginning as being useful for more than just the beginners. But the key concept that we had in mind was that the language ought to be built in such a way that it's sort of in levels, and that the beginning user learns as little as possible, and then you learn the next level and the next level beyond that. The interesting thing as a result of that is that the programs that ran in 1964 on the system still run on the system today. The answers are not always the same because I wrote the first compiler and there were some errors in it. Basically the very same programs still run. I'm sure that the program I wrote for generating prime numbers by the sieve method would still run today. There's nothing in it that has been changed since that time. We really philosophically believe that there ought to be a different kind of computer language. FORTRAN was an enormous advance when it first came. A user oriented language is the big break through. But that does not mean that the first one is good. I'm afraid that FORTRAN was written with advanced users in mind, and we felt very much that there had to be a system written which gave first priority to the inexperienced user but also could accommodate the more advanced user."



M.B. "At the same time that BASIC came into existence, IBM had a system called Quicktran, which was a FORTRAN in a time sharing environment. A little over a year later, John McGeachie and I and a consulting company wrote the memo into IBM that permanently put Quicktran out of existence by essentially comparing the two systems in terms of ease of learning, ease of remembering, and growth complexity. At that point, IBM decided to standardize into the 360."

N.B. "One very embarrassing afternoon I suspect for IBM was when they decided to demonstrate Quicktran at Dartmouth. So they put the terminal up in our teletype room which was filled with students of varying sizes all working at terminals getting their work done and then there was this Quicktran terminal. I was sort of embarrassed, too, because I had previously worked for IBM and happened to know some of the people who were there. I'm sure that they just at that point were not expecting or prepared to demonstrate their time sharing system to people first of all, students in blue jeans and stuff, who had themselves written a time sharing system that was being actively used all around them. It was really rather awkward. Remember it?"

T.K. "Did anybody ever?....."

N.B. "They spent two hours and I don't think they got the little program running."

T.K. "Well, I hate to call things to a close here because I think there are probably many other things that people are recalling, but I'm getting some signals from some very important people in the audience. Carry on? All right."

J.K. "I was just thinking since such a beautiful...I haven't seen this slide show and I thought it was beautifully done, as you really got across the idea of a system whose primary design was to make it a simple and easy to use as possible. We mention it in terms of BASIC. Of course, equally important just the very simple commands like NEW, OLD, SAVE, RUN, were really a major breakthrough in computing. The thing that disturbs me is that this is now ten years later. When will other systems learn that this is a good philosophy for writing computer systems? I happened to visit another academic institution which I won't identify, but I'm very fond of. They have an operating system in there and in introducing me they said, 'Here is a college president who speaks BASIC' and then said, 'instead of speaking...' and mentioned the name of that particular operating system, at which point the entire student body starting laughing uproariously at the joke, that had to be explained to me afterwards because I didn't know what the reference was. I just don't understand why it is so...I am not saying that what we did couldn't be done much better today. I'm sure somebody will do it much better. Why is the philosophy of doing things so it's easy to use for the users so terribly hard to get across to other systems. That's something I just don't understand."

S.H. "I've noticed that the way the BASIC Manual was written, most people are convinced that NEW, OLD, SAVE, RUN, are part of the basic compiler. A lot of other time sharing systems you type

RUN BASIC and it comes back NEW OR OLD--we at Dartmouth always got a big kick. That was separate from BASIC. That was the operating system. I mean, that's what made you use the same command for ALGOL, TSAP, any of the other languages you want, and it's a real shame that other people thought that RUN, NEW, OLD, and SAVE were part of BASIC, not part of something above BASIC which you would use with other languages."

J.K. "I think you're quite right, Steve, and yet on the other hand, there is part of the design that the user didn't have to know there was such a thing as an operating system."

S.H. "I noticed that when General Electric came out they came out with a separate command system manual separate from the BASIC manual. You now have to wade through two manuals to learn the system. I think that's a very good idea, putting it in one manual. But a lot of people, computer science professionals misunderstood it. They started to think that all these things, the EDIT system, the COMMAND system, and BASIC was all one thing which was all written up in one manual. Whenever you have three separate things, you're supposed to put them in three separate manuals, not one."

M.B. "One of my recollections from that period of time was that although the time sharing system at Dartmouth had its genesis in the desire to make computing an integral part of the education process at Dartmouth, not all of the faculty was swayed over to the use of computers that quickly. For some years after the system became operational there was a certain level of hostility among certain factions of the faculty. I recall particularly the fact that--I guess our senior year--Kevin and I worked up the nerve to take a course in music composition. It's very interesting listening to Professor Appleton today. It was just very encouraging to see the difference in attitude that existed in the music department then and now. I recall at the time we were learning composition in the form of chorales, and we were studying Bach and we had exercises in composition in that form. One of our big problems was that we didn't have a synthesizer, and we didn't play the piano, and we had a very difficult time listening to the thing we composed. After we worked very hard to put together a composition as a homework assignment, we had to either two-finger pick it on the piano or to find a music major who would play it for us, and turned it in for grading. Instead of coming back with comments about how aesthetically pleasing it was, the corrections that were made by the teacher were always nit-picking about technicalities that we had a parallel fifth there, and a triad some place else, and I'm probably belying how much I forgot about the harmony course. But Kevin and I were kind of discouraged by this because we had gone into it not as purely a technical endeavor, but to express a little creativity. So we decided that we would solve that problem by simply writing a computer program that would check all our compositions for technical errors before we handed them in. After we composed something and were fairly satisfied with the way it sounded, we'd type it into the computer and get a list of diagnostics just

as if it were a program. And if there were any technical errors we could correct them, so that it was assured that everything we turned in was technically perfect. It may have sounded terrible, but it was technically immaculate. This puzzled the professor for some length of time, and when he finally discovered what was going on, I think he almost ejected us from the class. By the end of the trimester, we had swayed him over to our way of thinking, and he was starting to think about the use of a computer. But students who were motivated to make early use of the computer in doing their homework and so on, often met with this kind of resistance in the early days. Now, of course, the system is just tremendously well accepted throughout the university."

R.M. "I'd like to say something about the reliability that we mentioned earlier. I do remember one occasion on which the installation of a new monitor wasn't as successful as it usually was. In fact, there was a rather blatant flaw that whenever a student typed TEST, which is the way of determining whether a BASIC program did its job, it would crash, not the 235, but the D-30, which was a very unusual kind of software failure in those days, but it was also disastrous in the sense that it was almost impossible at that point to recover. We had live users all over outside trying to use the system. Once it tailed, went down, it would come back up by itself, and before we could fix it, it would fail again, because somebody else would type TEST. Fixing it meant putting an octal patch into Datanet 30 core from the control terminal, but it was just failing too fast to do it. At one point I was just about ready to tear my hair out when I finally found out what was happening. At that time it was Professor Kemeny, came storming in and said, 'Why doesn't the system work? Every time I type TEST it seems to fail.' I remember going off the deep end and my response was, 'Well, if people would stop typing TEST for awhile, I could fix it so it would work.'"

S.H. "It is interesting. I was a freshman when Mike was a senior, but by the time I was a sophomore taking sophomore physics lab, I got very ambitious and did my physics lab on the computer. This grad student, who was a lab assistant, said 'Computer? No. You're supposed to be doing this by hand. I don't want to see it.' By complaining to the lecturer, he immediately went back to the grad students and said you encourage kids to use the computer. If you don't know how to use the computer, they give a two hour course for faculty members. You must take it, and you must encourage people to use the computer. In just a very short period of time, a good deal of the faculty members were going out of their way. If you wanted to use the computer on some extra credit thing, they preferred you to do it that way."

R.M. "Along the line of physics, in my freshman year first term or second term, before the time sharing system was available, the only computing facility we had on campus that was available was the 1620 at the Tuck School. I was taking a sophomore physics course, and since I hadn't been through all the rigors of labs in a freshman class, (I skipped the courses), I really didn't know what was expected in terms of labs. There was one particular experiment where we were measuring the charts of the mass ratio of an electron, and I was the only person left in class that didn't have a lab partner. I decided that the way to get good results was to take a large number of measurements. I took about 25 measurements, and

I analyzed the data on the 1620. The only reason I got a B on the lab instead of an A I remember, was that the guy who graded it said I didn't show my work."

T.K. "On that note I'd like to turn the meeting over to a couple of very fine people from AFIPS who have a job to perform, I guess. Just one? OK. I recognize that good looking young man over there, that's George Blazer, President of AFIPS. George...."

G.B. "I feel like a heel. You're obviously having such a good time. I certainly don't want to break up a session like this because they are very rare, and it's nice to watch it going on. I also can do nothing but applaud your pleas for simplicity, and certainly in the interest of the user. But it is my very great pleasure at this point to present on behalf of AFIPS to both Drs. Kemeny and Kurtz an award to tell them just in a little bit how we feel about the contribution that they've made. And if I may, I'll read to you a citation. The citations are identical, but we did invest in two plaques. It reads as follows: "American Federation of Information Processing Societies, Inc. commends Dr. John G. Kemeny, President, Dartmouth College in recognition of his pioneering efforts in the development of time sharing and his co-authorship of the computer language BASIC. Pioneer's Day, May 8, 1974, National Computer Conference." Dr. Kemeny...Dr. Kurtz..."

T.K. "Thank you. I don't have an acceptance speech, but I do have an announcement from Ted Belon to the effect that those of you who are counting on getting bus service tonight, there will be bus service at 7:00 p.m. and also after the play "Oklahoma" gets out."

J.K. "Tom, may I give an acceptance speech on behalf of both of us then? It is customary when accepting an Academy Award to mention your mother, your father, your music teacher, and your coach, and everyone else who really deserves the award. And I'm going to follow that precedent by saying that for both Tom and me this award would not exist except for the work done by a number of undergraduate students, Dartmouth College, and one graduate student, many of whom are sitting right here. And some of them are not here. Therefore, I think they deserve about half, each one deserves about half of our award and the other half should be divided equally amongst the other students who did all the work."

T.K. "I guess that concludes the session. Thank you very, very much."