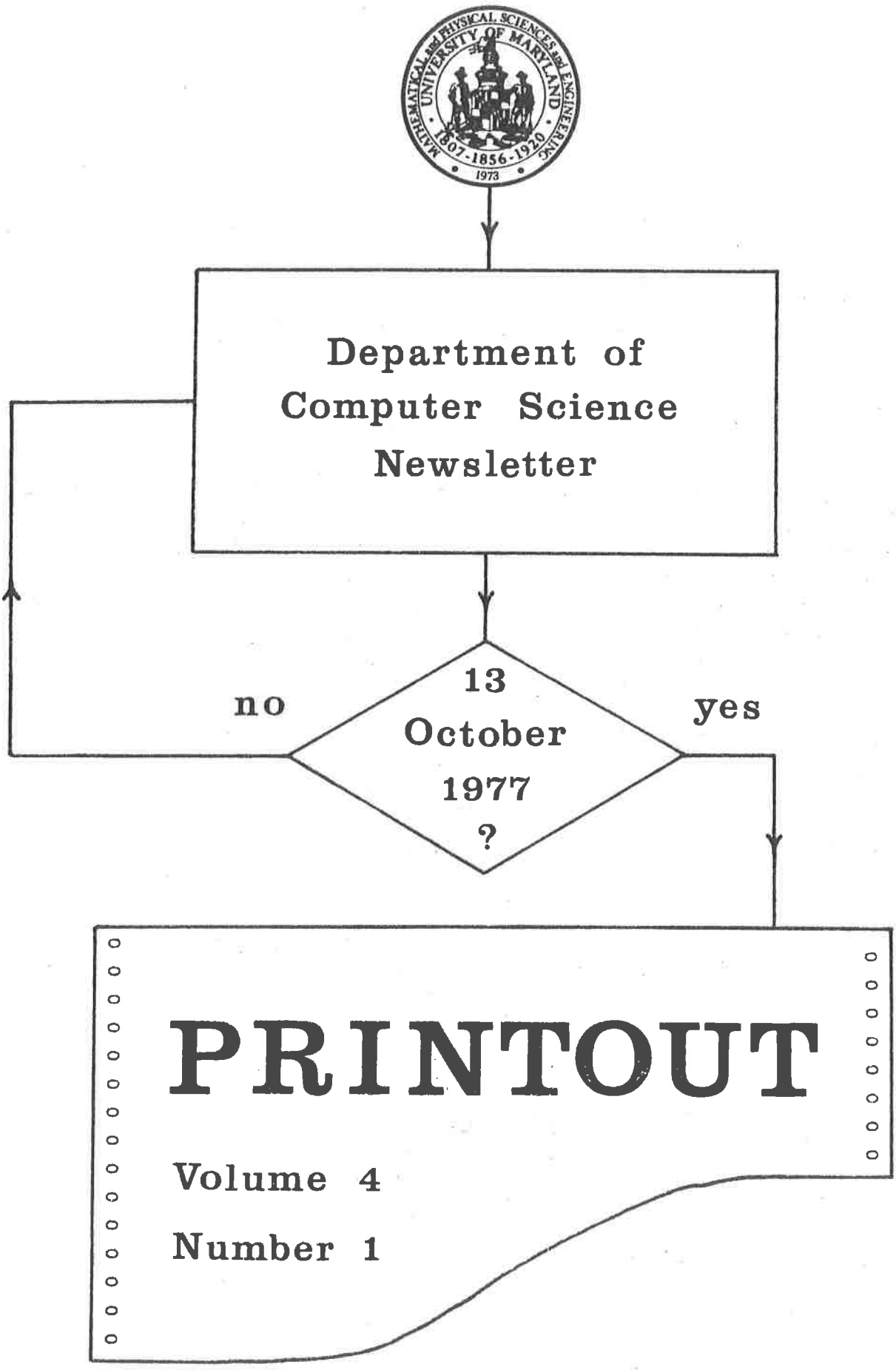

Department of
Computer Science
Newsletter
13
October
1977
?
no
yes
PRINTOUT
Volume 4
Number 1

PRINTOUT, the newsletter of the Department of Computer Science of the University of Maryland at College Park, is published sporadically and distributed to faculty, staff, and students in the Department. Opinions expressed in signed articles may be those of the author, but no opinions represent the policy of the Department, or of the College Park Campus, or of the University.

Contributions may be submitted to the editor, and unless they are obscene or seditious they will probably be used, but minor editing may be done. Complaints directed to the newsletter will be investigated and publicized when possible. It is well to keep in mind however that the Department is subordinate to higher levels of administration, not the other way around; and, the Department does not provide computing service to the campus. Complaints in these areas are best directed to other publications.

STAFF

EDITOR
Dick Hamlet

TYPING
Shiela Millhollon

PHOTOGRAPHY
Marv Zelkowitz
p 8-9
Barry Smith
p 14
E.D. Sewell
p 14

PRINTOUT

Volume 4
Number 1

CONTENTS

PROGRAMS & AUTOMOBILES

```
((LAMBDA (F) (F (QUOTE A)))
 (CONS (QUOTE LAMBDA)
       (CONS (CONS (QUOTE X) NIL)
             (CONS (CONS
                    (QUOTE EQ)
                    (CONS
                     (QUOTE X)
                     (CONS (CONS
                            (QUOTE QUOTE)
                            (CONS (QUOTE A) NIL))
                           NIL)))
                   NIL))))
```

How to Tell a Program from an Automobile

Computer programs and automobiles are both important in modern life, and people should learn to distinguish one from the other. Let us consider how "the man in the street" might try to find out the difference between them. This is not as easy as it sounds. We might think that he need only be told that automobiles are made mostly of metal parts, whereas programs are made of things like subroutines and assignment statements. Unfortunately, if he has never seen a subroutine or an assignment statement, he might think that they are just some kinds of metal parts. Being told what programs and automobiles are used for might not help either. What something is used for is an expression of intention, and intentions are often hard to discover by looking at what something does. (Seeing an automobile in a traffic jam, it might be hard to guess that it is used to move people from one place to another.)

Our man in the street could try asking an auto mechanic to tell him about automobiles, then asking a programmer to tell him about programs, and comparing the results. The mechanic might reply as follows. "An automobile is constructed by people. After they have finished constructing it, it runs. However, it may have some bugs. [Most mechanics would use the term "problems", but everyone understands this one when he speaks of 'bugs'.] These are usually minor [such as non-functioning windshield wipers], but occasionally they are serious enough to make the automobile stop running. The bugs are found by testing, and are fixed. The automobile then runs properly, and is sold to the user. The user then operates the automobile. However, as he uses it, new bugs may appear. Therefore, the automobile must be maintained. Maintenance is also required if the user wants to modify the automobile -- for example, to improve its performance [perhaps by 'souping up' the engine]. When the automobile becomes too difficult to maintain, the user has to buy a new one."

Unfortunately, the programmer might say almost exactly the same things about pro-

A Mercedes is not a Program (and a Good Thing, too)

Although I am not a "man in the street" (except when I have to get out and push), I have read Leslie Lamport's helpful advice, and I think I understand it. He says that the reason I can't prove my program correct is that it was probably written by someone who thought he was building an automobile. Now I believe in knocking programmers (they are such easy targets), but it occurs to me that it might also be interesting to imagine an automobile which cannot be fixed or maintained, because those who constructed it thought they were making a program. I have a particular automobile in mind, a 1962 Mercedes (1959 in the vicinity of the left front). This vehicle has a checkered history of association with friendly foreign-car repairmen of varying competence, and has been fixed and maintained a good deal.

I bought the car in 1969 from a very careful man who did all his own work, and only drove the car on Sunday afternoons. He was selling because on the previous Sunday he had run afoul of a 10-ton truck, and the Mercedes had sustained damages to the left front to about twice its market value. I found some used parts, but they were the wrong color, to mention only one of my problems. I used English bolts to put them on, because the metric ones were too expensive. But for an occasional fan belt and tire, the car then gave very good service.

When I brought the Mercedes to Maryland, it immidiately ran into trouble in the new environment. First of all, the state inspector didn't like German safety glass. He wanted to see the letters "AS" and instead there were "DN". I suggested that perhaps these meant the same thing, in English and German, but he was unmoved. So I found a used windshield which had been inspected in the District in October, 1970. After a short but bloody battle with the MVA, who didn't like the fact that the car had two different serial numbers, one on the right (the original) and one on the left (from the replaced fender), I was rolling down I-95 daily. Then my real difficulties began. The climate seems as unsuitable for cars as for people, and the car began to run badly. Eventually I discovered a person who put a convincing case that the carburetter had to be replaced. Alas! None were available. However, this person just happened to have a vacationing relative going to Germany, and she brought back the new carb in her purse. She said

grams! Our poor man in the street would be no further towards his goal. The reader has undoubtedly guessed by now that I am not interested in explaining the difference between a program and an automobile to the man in the street. He was just a pedagogical device, and will now be returned to his street. My purpose is to explain the difference between a program and an automobile to programmers.

An automobile runs, a program does not. (Computers run, but I'm not discussing them.) An automobile requires maintenance, a program does not. A program does not need to have its stack cleaned every 10,000 miles. Its if statements do not wear out through use. (Previously undetected errors may need to be corrected, or it might be necessary to write a new but similar program, but those are different matters.) An automobile is a piece of machinery, a program is some kind of mathematical expression.

Programmers may be disheartened to learn that their programs are not like automobiles, but are mathematical expressions. Automobiles can be loveable -- one can relate to them almost as if they were alive. Mathematical expressions are not loveable -- they are hard to relate to. Many programmers may feel like clinging to their belief that programs are like automobiles. However, further thought reveals that mathematical expressions do have certain advantages over automobiles.

Unlike an automobile, a mathematical expression can have a meaning. We can therefore ask whether it has the correct meaning. One cannot talk about the correctness of an automobile -- it may run properly, but it makes no sense to say that it is correct. Since a program is a mathematical expression, one can (at least in principle) decide if it is correct. The user and the programmer can agree in advance what the meaning of the program should be, and the programmer can prove mathematically that his program has that meaning. An automobile salesman can never prove that he sold a properly running automobile, he can only give enough evidence to satisfy a jury.

I have tried briefly to indicate the difference between an automobile and a program. The programmer should now apply his new knowledge in the field, and see if he can tell the difference by himself. He should first examine his automobile, and ask whether it is running properly, or if it has some bugs and requires maintenance. If he cannot answer this question, he may have to ask an auto mechanic to help him. He should then examine a program, and ask what its meaning should be, and whether he can prove that it has the correct meaning. If he can't answer these questions, it is probably because the program was written by someone who thought he was building an automobile. In that case, I suggest that the programmer ask these questions about the next program he writes -- while he is writing it! In this way, any programmer can learn to tell a program from an automobile.

-Leslie Lamport

the customs people just laughed. All was well once more.

I could go on through the new front end, the rebuilt radiator, not even to mention exhaust systems, or the left door hinge that M-B finally located in Chicago: the last one in existence in North America. The most recent alteration is the fuel pump. In the past it was rebuilt with Volkswagen parts, which were apparently inferior and lasted only two years; now I have a brand-new J. C. Whitney wonder ($10.68) made in Italy by scabs (utilizing French parts).

Now, how would things have gone for me and my Mercedes had it been a program? Probably very badly, since being a mathematical object, it would not have been composed of parts which could be replaced and substituted when circumstances demanded a change. It would have been proved at the factory, but I'm sure the collision or the windshield or the English bolts or the J. C. Whitney fuel pump would have voided the warranty, and I would not have wanted to ship it back to Germany for recertification. An automobile is a complicated object, and I expect things to be wrong with it. Why, it may be almost 1/10 as complex as some computer hardware, and that in turn may be almost 1/100 as complex as some software. So it is nice that it is acknowledged that failures will occur, and things are organized to make it possible to locate and repair them, if not always with factory-approved parts and techniques.

One of the friendly foreign-car repairmen that I have encountered has a boat powered by a pair of diesels taken from two 1959 180D Mercedes. He had to tinker with them a bit, but he says they are as steady as a rock, and require no maintenance at all. It just goes to show you what engineering can accomplish if it isn't too proud and uses the right methods.

-Dick Hamlet

J. P. Privitera

John Privitera was born in Jersey City, New Jersey in 1950. After an uneventful childhood and adolescence spent in the company of the Malay pirates (why else, do you suppose, is he usually addressed in correspondence as "Privateer"?) he enrolled in Stevens Institute of Technology, intending to study engineering. That lasted one semester, after which he became a math major, receiving his B.S. (let that one pass), and M.S. in 1972. For the last five years (before coming to Maryland) he has been at Cornell University in the Applied Mathematics doctoral program.

His current research interests are (in no particular order): algorithms and complexity, verification and denotational semantics, parallel processors, proof theory and constructive logic, and learning his way around the UM Campus.

Dr. Foonman

— Director of Virtual Personnel

Doctor;12/19/75;16:22:04/U2.2

--- Hello there. The Doctor is now in. Won't you please lie down on the couch; and tell me your problem?

* *Thank you Doctor. It's good to see you again.*

--- Why is it good to see again?

* *No. You don't understand. That's not my problem.*

--- Please go on.

* *Well, my problem is Pat Haggerty. We gave him a task and he never came through.*

--- What is it through which Pat Haggerty never came?

* *No, you still don't understand. Pat got an assignment and never finished it.*

--- By "Pat" do you mean "Pat Haggerty"?

* *Yes.*

--- Please go on.

* *Well, as I said, he never finished the assignment we gave him.*

--- What assignment did you give Pat Haggerty?

* *That's confidential.*

--- What is confidential?

* *The assignment we gave Pat.*

--- Who is "we"?

* *That's confidential.*

--- Are you telling me that "we" is the assignment that you gave to Pat Haggerty and that he did not finish?

* *No. You still don't understand. A group of us gave Pat Haggerty a confidential assignment which he never finished.*

--- Describe "a group of us" for me.

* *That's confidential also.*

--- Are you playing games with me?

* *No. I need your help.*

--- When did you first realize that you had this problem?

* *I haven't finished stating the problem yet.*

--- You have not finished stating the problem yet, and Pat Haggerty has not finished the assignment yet. How do you feel at the moment?

ANNOUNCEMENTS

ACM '78 will be held December 4, 5, 6, 1978, at the Sheraton Park Hotel in Washington, D. C. The conference will feature technical and government programs, book exhibits, and the U. S. Computer Chess Championship. The technical program will emphasize advances in all areas of computing. Papers and proposals for special sessions and panels are solicited. Five copies of all papers or proposals should be submitted by July 1, 1978. The government program will stress current issues and applications of computers in the Federal Government, and will consist of presentations by representatives from a large variety of federal agencies. For information on ACM '78:

Richard H. Austing, General Chairman
Department of Computer Science
University of Maryland

The schmooze schedule is first and third Thursday of the month (e.g., October 6, 20, etc.) until December 1. Hours are 2:30-4:00 in the third-floor lounge.

Schmooze

What is a schmooze?
When everyone lets looze,
To see whom they chooze,
Exchange technical nooze,
Enjoy cookies and jooze,
That is a schmooze!

-Anonymooze

-Dianne Martin
Schmoozperson (by decree)

The Department hopes to soon hire a kind of "software technician" to support both the fourth-floor Laboratory (mostly PDP-11s), and Univac 1100 software. The position will probably be full-time, but arrangements can be flexible if other committments are involved. Salary and working arrangements are open. The successful applicant will be a Wizard whose programs not only do all good things and no bad things, take no space and no time, but are also produced to order with a wave of his or her wand; and, he or she must perform these feats without the use of a PDP-10.

Suggestions for how to locate potential applicants, as well as tentative applications themselves, should be directed to Dick Hamlet.

WANTED! Computer Science Graduate Students, Faculty, and Staff -- Male, Female, and Otherwise -- To participate in Intramural Sports and represent Computer Science in the Grad-Faculty-Staff League. Last year Computer Science placed 5th overall out of more than 100 participating departments and captured a number of first place Gold Medals. This year we hope to do better, with your help. Both team and individual sports are available --participation is the key. It is a great way to meet members of the Computer Science Community in an informal and non-academic atmosphere. Both men and women are welcome. The following sports are available:

Fall Sports	Spring Sports
Tennis (s)	Table Tennis (s)
Horseshoes (s)	Weightlifting
Touch Football	Bowling
Soccer	Volleyball
Golf	Softball
Badminton (s)	Foul Shooting
Cross Country	Wrestling
Basketball	Handball (s)
One-on-one BB	Racquetball (s)
Swimming	Track & Field

(s) = Singles

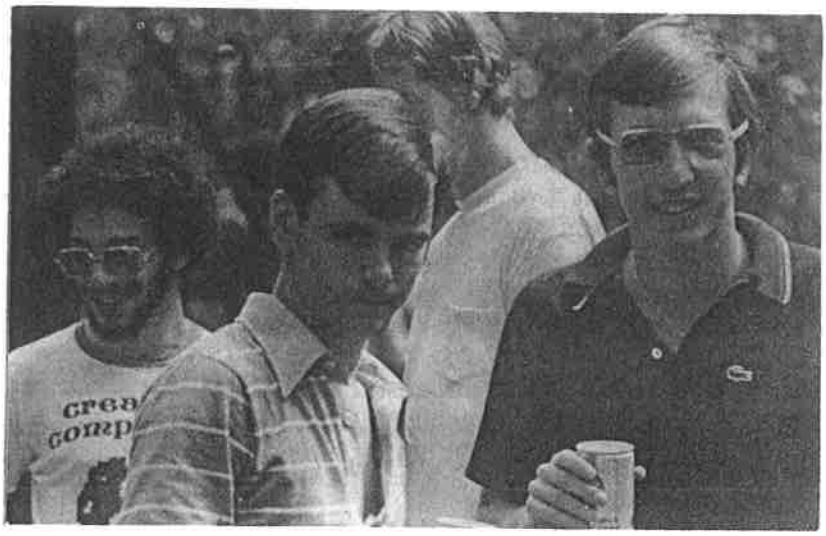
If you are interested in any of these, please give your name and phone number to:

Steve Rochell	Rm 3302	x4255
Bobb Budd	Rm 3365	x4251

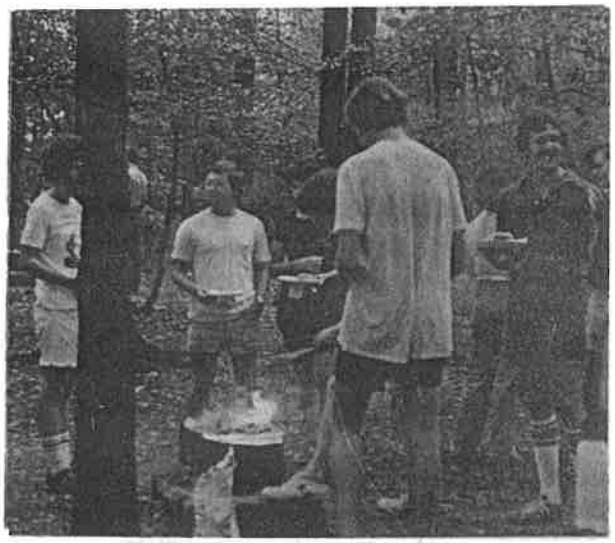
You may participate on any number of teams, but individual sports (i.e. tennis, bowling, golf, etc.) may have no more than 5 people entering from Computer Science.

The 1976-77 Annual Report is available (free) in the Education Office.

P I C



N I C



103 + 120 = 210 ?

The two descriptions below augment catalog information on two important computer science courses. CMSC 103 is a first course for non-majors (majors take CMSC 110 instead). CMSC 120 is the heart of the undergraduate program, and the first contact with real computer science (as opposed to FORTRAN programming). The descriptions were written by faculty teaching the courses in the fall semester.

103

CMSC 103, the introductory course for non-majors in Computer Science, is designed to be meaningful for a broad spectrum of students without sacrificing rigor. The course objectives are to provide a general understanding of the stored program computer, to emphasize algorithm development in problem-solving, to develop a proficiency in using FORTRAN to solve problems from varying application areas, and to foster an appreciation for the present and future impact of the computer on society.

The course requires completion of 6 or 7 FORTRAN programming projects of increasing difficutly, two hourly exams, and a final exam. Required texts are the Cooper and Smith, Standard FORTRAN: a Problem-Solving Approach, and the Adams and Haden, Social Effects of Computer Use and Misuse. This semester there are over 400 enrolled in the course with an expected droprate of 25%. There are over 50% freshmen in the course. Majors include journalism, dietetics, business, engineering, English, history, and politics.

-Dianne Martin

120

The objective of CMSC 120 is to teach the application of techniques and the use of tools that will aid in the solution of large, complex programming problems. Subjects covered in the course include structured programming, elements of style, testing and debugging, recursion, simple data structures, searching and sorting algorithms, and program verification.

Four programming projects (two small and two large) are assigned. There is one midterm and a final examination. Five quizzes are spread throughout the semester.

All programming is done in SIMPL-T, a structured programming language. Five lectures and four laboratories are spent teaching SIMPL-T. Three laboratories are spent teaching use of the computer center, including interactive use.

A complete syllabus for the current semester, including all programming project assignments, is available in the Education Office.

The final grade is calculated as follows:

Projects:	65%	(Design)	Exams:	30%
P1:	5%		(Mid-term:	10%
P2:	10%		Final:	20%)
P3:	10%	(2%)		
P4:	20%	(4%)	Quizzes:	5%
P5:	20%	(4%)		

-Mark Ardis

I know that at the end of last semester, I loaned my copy of Symbolic Logic and Mechanical Theorem Proving by Chang and Lee to someone who was studying for the comps, and now I can't remember who it was that I loaned it to.

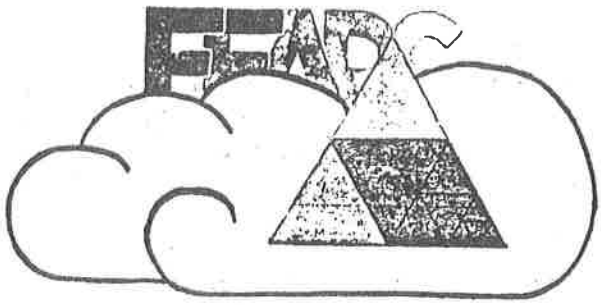
The comps are over, the grading of the comps is over, and all of the office moving is over, so everyone should be settled --maybe this is a good time for everyone to spend five minutes looking through the books they have collected, and return those that need returning! This would not only help me get my Chang and Lee back, but the library might be in better shape for awhile too.

-Paul McMullin

Animation

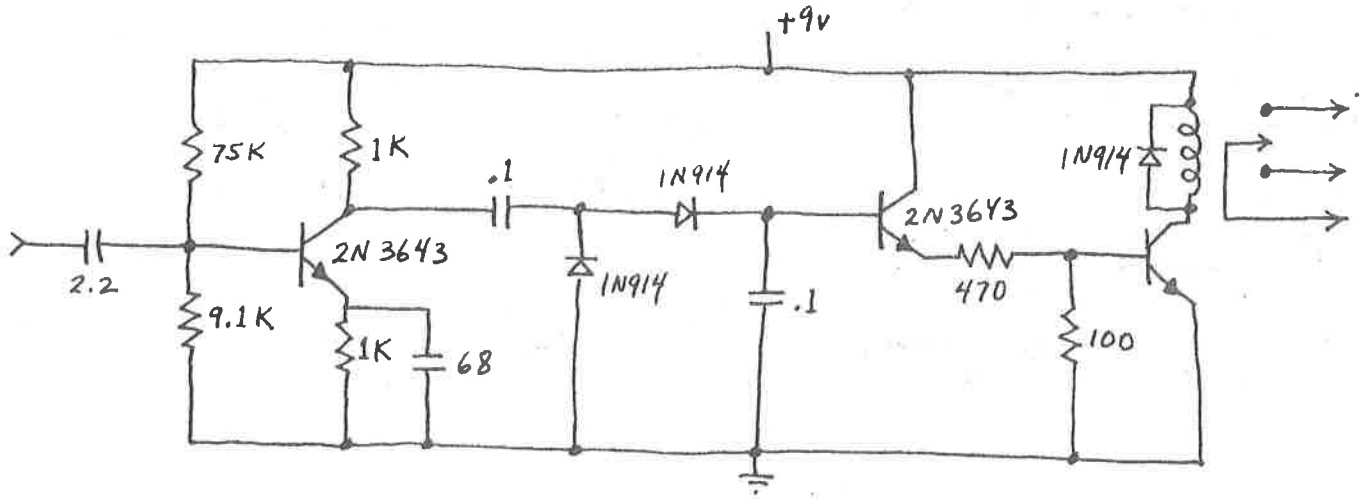
The circuit shown allows the use of a single-frame, automatic-film-advance camera system at a computer terminal. This system was designed for a Textronix 4010 terminal connected to the University of Maryland Univac series 1100 computer for the generation of computer animated films. The camera utilized was a Nizo, Model #561, filming in Super 8 format. The circuit is placed between the terminal bell and the camera relay release, enabling a program-sent "CTRL G" (ASCII 007) to act as a shutter release. The Textronix bell wires were paralleled to a female stereo chasis jack to insure electrical isolation. The power supply for the circuit is a standard 9-volt transistor battery. Total parts cost was less than $10. The system is fail safe since system slow-down only causes an increase in filming time. System service interruption yields the generation of several extraneous frames due to the bell prompting. The circuit contains an isolated output relay which would allow the unit to be installed on a camera without electronic relay shutter release through the addition of an external solenoid attached to a shutter release cable. It is not limited to the Textronix display terminals but can be utilized on any CRT terminal with an electronic bell. Thanks are extended to Bernard Wess for tolerating the modification of his terminal and Mike Petry for a lot of technical backup.

-Fred Stern



The finite element method is a numerical method used to ascertain that bridges won't fall down, nuclear reactor shields won't crack, nor airplane fuselages break in two. In this country alone this type of problem accounts for $2 billion/year in computer time. FEARS (the Finite Element, Adaptive Research Solver) is a prototype system which solves these problems in a radically new way, including parallelism and interactive adaptivity. FEARS will be built this year and used for experiments in summer, 1978. (Every experiment is accompanied by some fears about the outcome.) The FEARS team is Prof. Babuška (IPST), Prof. Rheinboldt (MAPL, CSC), Prof. Zave (CMSC), Charles Meszteny (CSC), George Cole (CMSC), and Bob Fini (CMSC).

-Pamela Zave



News

Dr. Pat Hagerty, who had a joint appointment in our Department, and was in charge of the Systems Staff in the Computer Science Center, has resigned effective August 28, 1977.

Two Department of Computer Science graduate students and one faculty research associate presented papers at the recent International Conference on Cybernetics and Society, held September 19 - 21 at the Mayflower Hotel in downtown Washington and sponsored by the IEEE Systems, Man, and Cybernetics Society. Graduate student Mike Lefler presented the paper "Automated Interpretation of ^{1}H NMR Spectroscopy." Faculty research associate Dr. Dave Milgram delivered a paper on "Automatic Object Detection in FLIR Images." Another DCS graduate student, Jerald Zeger, was a luncheon guest speaker at the conference. Jerry gave a talk entitled "The Computer Invades the Home."

And in the office...

Eleanor Waters has been out on sick leave since June 11, 1977. She is expected back in October.

Janice Keough has been in the Department since June 20, 1977 and is doing an outstanding job of handling her job and Eleanor's.

Shiela Millhollon (part-time TERP secretary) is here until Eleanor returns.

Brenda Bigwood has been account clerk to JoAnn Thompson since July 1, 1977.

Phyllis Hamilton has been secretary to Dr. Austing since July 5, 1977.

Where are they now?

Betty Donovan is now located in the Psychology Department as Secretary to the Assistant Chairman.

Alice Eichman is working in the Astronomy Department since November 7, 1976.

Peggy Strand is working for a governmental agency in Fargo, North Dakota since May, 1977.

Pat Young is working part-time at home typing technical reports etc.

Awards

DR. CHARLES RIEGER has received a renewal on the NASA contract entitled "Improving Communications Between Humans and Computer Directed Automatic Systems". NASA will provide Dr. Rieger with $43,000.00 effective July 1, 1977.

DR. CHUL KIM has received a new NSF contract entitled "Design and Analysis of Algorithms for NP-hard Problems". NSF will provide Dr. Kim with $26,900.00 effective June 15, 1977. Abstract: There are several objectives in the proposed research. One is to find good heuristics for solving computationally difficult problems approximately. Another is to identify problems that do not admit efficient algorithms within certain worst-case bound. Also, probablistic analysis will be attempted to some simple heuristic algorithms.

DR. LAVEEN KANAL has received a renewal NSF contract entitled "Interactive Pattern Recognition Research". NSF will provide Dr. Kanal with $50,000.00 effective October 1, 1977. He has also received a renewal AFOSR contract entitled "Pattern Analysis and Modeling". AFOSR will provide Dr. Kanal with $42,470.00 effective September 1, 1977. Dr. Kanal has also received additional funds in the amound of $20,160.00 from PHS/NIGMS for his Grant, "Resource-Related Research Graphic Formula Translation"... total funds $62,160.00. Duration of the grant is May 1, 1977 to April 30, 1978.

DR. YAOHAN CHU has received an extension of his Grant from the NSF to October 31, 1978. Funds provided previously were $104,909.00. Title of the Grant is "Language-Directed Computer Systems".

Publications, Etc.

ASHOK K. AGRAWALA

MACHINE RECOGNITION OF PATTERNS, (Editor), John Wiley & Sons, Inc., 1977.

"A Study of Shared Memory as a Communications Medium," PROC. OF 16TH ANNUAL TECHNICAL SYMP. ON SYSTEMS & SOFTWARE, National Bureau of Standards, Gaithersburg, Md., June, 1977, pp. 51-60 (with R. M. Bryant and J. R. Agre).

W. F. ATCHISON

"Curriculum Committee on Computer Science," THE ENCYCLOPEDIA OF COMPUTER SCIENCE AND TECHNOLOGY, Vol. 6, Marcel Dekker, Inc., 1977, pp. 486-503 (with Gerald L. Engel).

RICHARD H. AUSTING

"Curriculum Recommendations for the Undergraduate Program in Computer Science: A Working Report of the ACM Curriculum Committee in Computer Sciences," SIGCSE BULLETIN, 9, 2, June 1977, pp. 1-16 (with Bruce H. Barnes, Della T. Bonnette, Gerald L. Engel and Gordon Stokes).

"Curriculum Recommendations and Guidelines for the Community and Junior College Career Program in Computer Programming: A Working Paper of the ACM Curriculum Committee in Computer Science by the Subcommittee on Community and Junior College Curriculum," SIGCSE BULLETIN, 9, 2, June 1977, pp. 17-36 (with Joyce Currie Little, Harice Seeds, John Manicotes and Gerald L. Engel).

"Computer Science in Health Computing Education," PROCEEDINGS OF THE SIGBIO SYMPOSIUM, June 12, 1977, Dallas, Texas.

"Literature Resources and Computer Impact on Society and Computer Literacy Courses,", PROC. OF THE COMPUTER SCIENCE AND ENGINEERING CURRICULA WORKSHOP, June 6-7, 1977, Williamsburg, Virginia, pp. 55-59 (with W. W. Cotterman and G. L. Engel).

VICTOR R. BASILI

"Software engineering laboratory: objectives," 15th SIGCPR SYMPOSIUM, Arlington, Va., August, 1977 (with M. V. Zelkowitz).

LARRY DAVIS

"Iterated image enhancement," 7th EIA-AIPR CONFERENCE, College Park, Maryland, May 23-24, 1977.

H. P. EDMUNDSON

"Statistical Inference in Mathemetical and Computational Linguistics," INTERNATIONAL JOURNAL OF COMPUTER AND INFORMATION SCIENCES, Vol. 6, No. 2, June 1977, pp. 95-129.

JOHN D. GANNON

"Experimental Investigation of Programming Complexity," PROCEEDINGS OF ACM/NBS 16TH ANNUAL TECHNICAL SYMPOSIUM, June 1977, pp. 117-126 (with H. E. Dunsmore).

VIRGIL D. GLIGOR

"Architectural Aspects of Type Extendibility," PROC. OF TRENDS AND APPLICATIONS 1977: COMPUTER SECURITY AND INTEGRITY, Gaithersburg, Maryland, May 1977.

RICHARD G. HAMLET

"Testing programs with the aid of a compiler," IEEE TRANSACTIONS ON SOFTWARE ENGINEERING SE-3 (July, 1977), 279-290.

MATTHEW S. HECHT

FLOW ANALYSIS OF COMPUTER PROGRAMS, Elsevier, North-Holland, 1977.

CHUL E. KIM

"Heuristic Algorithms for Scheduling Independent Tasks on Nonidentical Processors," JACM 24 (1977), pp. 280-289 (with O. H. Ibarra).

STEPHEN E. MANN

"A Microprocessor-Based Digital Multitrack Recording Facility," 16th ANNUAL TECHNICAL SYMPOSIUM ON SYSTEMS AND SOFTWARE, National Bureau of Standards, Gaithersburg, Maryland, June, 1977.

HARLAN D. MILLS

"Software Engineering," SCIENCE, Vol. 195, #4283, March 1977, pp. 1199-1205.

JACK MINKER

"Control Structure of a Pattern-Directed Search System", SIGART Newsletter, August, 1977.

"The use of a semantic network in a deductive question-answering system," PROC. IJCAI, August, 1977 (with J. McSkimin).

C. RIEGER

"Viewing Parsing as Word Sense Discrimination," A SURVEY OF LINGUISTIC SCIENCE, W. O. Dingwall (ed.), Greylock Publishers, 1977.

A. ROSENFELD

COMPUTER METHODS IN IMAGE ANALYSIS, IEEE Press, 1977, (ed. with J. K. Aggarwal and R. O. Duda).

"Two-stage template matching," IEEE TRANS. ON COMPUTERS C-26, 1977, 384-393 (with G. J. VanderBrug).

"An application of relaxation labelling to line and curve enhancement," IEEE TRANS. ON COMPUTERS C-26, 1977, pp. 394-403 (with S. W. Zucker and R. A. Hummel).

"Iterative enhancement of noisy images," IEEE TRANS. ON SYSTEMS, MAN, AND CYBERNETICS SMC-7, 1977, pp. 435-442, (with A. Lev and S. W. Zucker).

"Picture processing, 1976," COMPUTER GRAPHICS AND IMAGE PROCESSING 1977, pp. 157-183.