



COMPUTER  
CENTRE  
**BULLETIN**

*Volume 3, Number 12.  
7th December, 1970.*

*Editor:  
S. J. Barry.*



## THIS EDITION

This is the last edition of the Bulletin for the year. Organizations and University Departments will continue to receive copies of Volume 4 automatically next year so that reapplication will not be necessary. Individual readers, however, receive the Bulletin on a yearly basis only, and therefore it will be necessary for individual readers wishing to receive Volume 4 to complete the application form on the last page of the Bulletin. Please notify any address changes to the Editor immediately so that the Bulletin can always be dispatched to the correct address.

This edition contains an index of all major articles and computing items published in Volume 3. The index for Volumes 1 and 2 can be found in Vol. 2 pp.167-169.

This year has been a year of extensive development for the Computer Centre. Work has been concentrated on releasing remote terminals for the PDP-10 for Computer Centre clients. This work is nearly completed and the article on *Multiaccess Computer Systems* has been included in this issue to introduce prospective users not familiar with timesharing to some aspects of this mode of operation.

I am grateful to all those people who have helped me in the production of the Bulletin, especially in the early days when I had taken over from the previous editor. My thanks also to those who have contributed to the Bulletin and have provided articles of interest to readers and clients.

The Computer Centre would like to wish everybody a happy Christmas.

## CHRISTMAS SHUT DOWN

The Computer Centre will shut down for the Christmas period at midnight on Wednesday 23 December 1970. There will be no processing of work on the 24 December to allow the staff to finalise end-of-year operations and accounting. As usual, the Centre will be closed over the Christmas - New Year period and will re-open for business at 9 a.m. on Monday 4 January 1971.

## STAFF NEWS

We were sorry to say goodbye to *Miss Angela Vidanovic* this month. Angela has been working at the Centre as a Computer Operator for the past 12 months.

We are happy to welcome *Miss Noela Goodwin* to the Computer Centre. Noela had worked at Commonwealth Engineering (Qld) Pty Ltd, Rocklea as a Secretary for five years.

## NOTES ON FORTRAN IV

### 1. GE-225 FORTRAN IV

#### (a) VARIABLE AND ARRAY EQUIVALENCING

When a variable or array is made equivalent to a variable or array assigned to COMMON, the variable or array in COMMON must be the first of the group to be equivalenced.

For example,

```
COMMON A
EQUIVALENCE (A,B)
```

is valid while

```
COMMON A
EQUIVALENCE (B,A)
```

is not.

This error is mentioned in the old GE-225 FORTRAN manual.

### 2. PDP-10 FORTRAN IV

#### (a) DEFINITIONS IN DATA STATEMENTS

Care must be taken that values in a data statement correspond to the definition of the associated variable, since automatic conversion does not take place.

For example: DATA A, B, I, J/2., 3, 4, 5./

Variables A and I have the correct values of 2. and 4 respectively, while variables B and J will be incorrect.

The data statement should be:

```
DATA A, B, I, J/2., 3., 4, 5/
```

(b) COMPARISON BETWEEN A COMPLEX FUNCTION AND A REAL QUANTITY

A logical IF cannot involve a comparison between a complex function and a real quantity.

example:

```
COMPLEX C
.
.
.
If (C(J).GT.1.5) GO TO 100
.
.
.
```

This coding is incorrect since the complex function C(J) cannot be compared with the real quantity 1.5. Unfortunately the compiler does not produce an error message during compilation. However, it does produce a call to the error reporting program EXER1 and during execution this causes the following message to be given when an attempt is made to execute the statement.

```
SOURCE LEVEL ERROR AT nnn
```

(c) WIDTH OF OUTPUT FIELD

It is a feature of most FORTRAN systems that, if a specified output field width is too small, leading digits will be suppressed to reduce the field width to that specified. In most cases this 'feature' will cause misleading results, in fact some systems in this case produce an error message. However, it is purposely used occasionally, to truncate significant digits.

The PDP-10 FORTRAN execution system has an error which on integer output causes a gradual build up in the internal push-down list if this feature is invoked.

If this build up is too great, then the run will be terminated because of overflow of the push-down list, with the error message:

```
PDL OV AT USER 400....
```

(d) VARIABLE REDEFINED AS A FUNCTION

This is a restatement of a problem previously reported in the Bulletin, Vol.3 p.72. In some cases, the FORTRAN compiler fails to recognize the use of a variable name both as a simple integer variable and as a function name.

In the following example, code is generated for the second use of IJK as if it were a function, and at execution time this may lead to conflicting results.

No diagnostic message is produced.

```
IJK = 92
.
.
.
L = J + IJK (2,3)
.
.
.
```

### NEW COMPUTER SCIENCE SUBJECT

In 1971 the University of Queensland will introduce Computer Science at the undergraduate level for the first time. The recently established Department of Computer Science has planned a three year undergraduate course within the Faculty of Science, leading to a Bachelor of Science Degree (with provision for a fourth year for an Honours Degree). The University Senate has approved the course and its introduction in 1971 through the first-year subject Computer Science I. Until it is known whether funds will be available in 1972 and later years, the introduction of Computer Science subjects in second and third year cannot be guaranteed.

Computer Science I has the normal Matriculation requirements for the B. Sc. course, as set out in the University Calendar. The subject has no prerequisite or corequisite university subjects and is a one-unit subject (many first-year subjects in the Faculty of Science are two-unit subjects) comprising one lecture, one tutorial, and one practical hour per week throughout the academic year. The subject Computer Science I may be taken by students who intend to major in Computer Science or as an option by students who major in other disciplines.

The subject comprises a course of one lecture, one tutorial, and one hour of practical work per week, with one three-hour examination.

This course is an introduction to computer science and provides the student with the basic knowledge and experience necessary to use computers effectively in the solution of problems.

Algorithms, programs, and computers. Basic programming and program structures. Programming and computer systems. Debugging and verification of programs. Data representation. Organisation and characteristics of computers. Survey of computers, languages, systems and applications.

Students who intend to major in Computer Science and perhaps proceed to an Honours degree, are advised to plan a course of study along the following lines:

- First Year : Computer Science I  
Pure Mathematics Ia  
Physics Ia  
Two other first year subjects
- Second Year : Computer Science IIa\* and IIb\*  
Pure Mathematics II  
One other second year subject
- Third Year : Computer Science IIIa\*, IIIb\*, and IIIc\*  
Two other third year subjects

It is expected that the optional subjects would normally include a selection of mathematics topics such as Numerical Mathematics, Mathematical Statistics or Applied Mathematics.

\* The offering of these subjects in later years will depend on the availability of funds and staff.

#### LIBRARY ACCESSIONS

These books were acquired by the University of Queensland Libraries in August and September.

MENDEL, Jerry M.	<i>Adaptive, learning and pattern recognition systems</i> 1970 (001.533 MEN Maths)
ROSE, J. ed.	<i>Survey of cybernetics</i> 1969 (001.53 ROS Biol.)
SMITH, Gloria L. ed.	<i>Library use of computers</i> 1969 (021.0018 SMI Main)
RECON WORKING TASK FORCE	<i>Conversion of retrospective catalog records to machine-readable form</i> 1969 (029.7 REC Main)

- U.S. LIBRARY OF CONGRESS *Information Systems Office. The MARC II format*  
1968 (029.7 UNI Main)  
*Advances in information systems science v. 1*  
1969 (029.7 ADV Engin.)  
*Project intrex. Activity report no 8* 1969 and  
onwards (029.7 PRO P.S.)
- CARNEY, James Donald *Introduction to symbolic logic* 1970  
(164 CAR Main)
- BESHES, James M. ed. *Computer methods in the analysis of large-scale  
social systems* 1968 (300.18 BES Main)
- FORD, Gordon William ed. *Automation: threat or promise?* 1969  
(338.0640994 FOR Main)
- BELTRAMI, Edward J. *An algorithmic approach to nonlinear analysis and  
optimization* 1970 (517.5 BEL Maths)
- GREVILLE, Thomas Nall *Theory and applications of spline functions* 1969  
(517.8 GRE Phys)
- PRANGE, Eugene *Some cyclic error-correcting codes with simple  
decoding algorithms* 1958 (519.7 PRA Elect. Engin.)
- SEMINAR ON THE ROLE OF THE COMPUTER IN THE SECONDARY SCHOOL, Adelaide, 1969.  
*Proceedings* 1969 (510.78340712 SEM Engin.)
- KRASNOFF, Sidney O. *Computers in medicine* 1967 (610.18 KRA Chem.)
- BARRON, David William *Assemblers and loaders* 1969 (651.8 BAR Engin.)
- COMPUTER SCIENCE *FORTRAN language* 1970 (651.8 COM Engin.)
- FLORES, Ivan *Computer organization* 1969 (651.8 FLO Engin.)
- GRISWOLD, R. E. *The SNOBOL 4 programming language* 1968  
(Qto 651.8 GRI Engin.)
- MURRAY-SHELLEY, Richard *Teach yourself computer programming* 1967  
(651.8 MUR Chem.)
- KLEIN, Imrich ed. *Computer programs for plastics engineers* 1968  
(668.4018 KLE Engin.)
- HILLIER, Frederick S. *Introduction to operations research* 1967  
(001.4 HIL Main)
- INTERNATIONAL BUSINESS MACHINES CORPORATION, Data Processing Division.  
*Bibliography on simulation* 1966  
(Qto 016.001424 INT Biol.)
- LEISENRING, A. *Mathematical logic and Hilbert's  $\epsilon$ -symbol* 1969  
(164 LEI Main)  
*Data processing for education administration* 1968  
(Qto 371.20018 DAT Cent.Med.)
- STOLUROW, Lawrence M. *Computer assisted instruction* 1968  
(371.3944 STO Educ.)
- MATHISON, Stuart L. *Computers and telecommunications* 1970  
(384.018 MAT Engin.)
- BRITISH MEDICAL ASSOCIATION. Working Party on Computers in Medicine.  
*Report of the Working Party on Computers in  
Medicine* 1969 (610.18 BRI Clin.)
- BOVET, Daniel Pierre *Memory allocation in computer systems* 1968  
(Qto 621.3819533 BOV Engin.)
- CONTROL DATA CORPORATION. [Control Data]  
*3600/3800 computer systems INFOL reference manual*  
1966 (Qto 621.381958 CON Engin.)

- INTERNATIONAL CONFERENCE ON COMPUTER AIDED DESIGN, University of Southampton, 1969.  
*Proceedings* 1969 (Qto 621.38195 INT Engin.)  
*International journal of man-machine studies* v. 2 1970 and onwards (620.8 INT P.S.)
- AMERICAN INSTITUTE OF CERTIFIED PUBLIC ACCOUNTANTS.  
*Management information systems for the smaller business* 1969 (658.501 AME Main)
- ASSOCIATION FOR SYSTEMS MANAGEMENT.  
*Management information systems* 1969 (Qto 658.4 ASS Engin.)
- CALMAN, Robert F. *Linear programming and cash management* 1968 (658.503 CAL Agric.)
- DEAN, Burton Victor *Operations research in research and development* 1963 (658.57 DEA Main)
- DEARDEN, John *Computers in business management* 1966 (651.8 DEA Main)
- FARINA, Mario V. *Computers* 1969 (651.8 FAR Main)
- GOSSLING, W. G. ed. *Input-output in the United Kingdom* 1970 (651.8 GOS Main)
- HEIN, Leonard W. comp. *Contemporary accounting and the computer* 1969 (657.018 HEI Main)
- HUGHES, Marion L. *Decision tables* 1968 (Qto 651.8 HUG Main)
- INTERNATIONAL BUSINESS MACHINES CORPORATION.  
*The considerations of data security in a computer environment* 1969 (658.47 INT Engin.)
- KATZAN, Harry *Advanced programming; programming and operating systems* (651.8 KAT Engin.)
- LEE, John A.N. *The anatomy of a compiler* 1967 (651.8 LEE Engin.)
- MASSACHUSETTS INSTITUTE OF TECHNOLOGY. Operations Research Center.  
*Notes on operations research* 1959 (658.54 MAS Engin.)
- ORLICKY, Joseph *The successful computer system* 1969 (658 ORL Engin.)
- PRINCE, Thomas R. *Information systems for management planning and control* 1970 (658 PRI T.M.L.)
- SANDERS, Donald H. *Computers and management* 1970 (658.501 SAN Engin.)
- SCHERR, Allan Lee *An analysis of timeshared computer systems* 1967 (651.8 SCH Engin.)
- SEMINAR IN ADMINISTRATIVE DATA PROCESSING, London, 1967.  
*Working papers* 1969 (Qto 658.0018 SEM Engin.)
- STABLEY, Don H. *Logical programming with System/360* 1970 (651.8 STA Engin.)
- THIERAUF, Robert J. *Decision making through operations research* 1970 (658.4 THI Main)  
*EDP analyzer* 1969 and onwards (658.1 EDP P.S.)

## NEW DISKS

New disks have been placed in service on the PDP-10 system, but their use will not be immediately apparent. The disks have undergone thorough testing and the likelihood of any errors occurring is remote. However, users are asked to check their results and report any discrepancies that may chance to appear.

## REMOTE TERMINALS TO PDP-10

Over the past 6 months, staff of the Computer Centre have been working on the developments necessary to make remote terminals available to users. Much of this work is now nearing completion and most major sections are currently undergoing extensive testing.

Provided that no major delays are encountered, the first 16 remote terminals will be physically installed and connected to the PDP-10 this month. A formal announcement on the availability of the Timesharing Service will be made shortly, giving details of starting date, times of operation, administrative procedures and introductory courses.

The first release of the Timesharing Service will enable a user at a remote terminal to create a program as a file stored on disk, have it compiled and run, and the results returned directly to his terminal. He will be able to modify or correct the program with an editor program and immediately rerun the program. A number of utility programs will be provided to copy files, change passwords, print directories of disk files, etc.

Immediately after the first release of terminals, work will commence on the commissioning for subsequent release of new or extended facilities such as new programming languages (e.g. COBOL), increased file storage, and new services (e.g. plotting). It is intended that these and other facilities will be made available gradually during 1971.

## MULTIACCESS COMPUTER SYSTEMS

R.A. Cook

*Rob Cook is a Systems Programmer in the Computer Centre. Before coming to Queensland he studied computing at the University of Cambridge where he gained experience with the ICL Titan computer system. Later he worked for a Government Research Establishment on generalized desk calculator program languages.*

*Rob joined the centre in January this year. Part of his work has been concerned with developing a Decoder for the remote terminal command language.*

### Abstract

This paper describes the reasons for the growth of multiaccess systems and the methods adopted by different groups in implementing them. The CMAS (Cambridge Multiaccess System) and DEC's (Digital Equipment Corporation) PDP-10 system are discussed as examples of these methods.

### Introduction

In the early 1960's the computer world was full of powerful batch processing machines and small single-user computers. Small computers have become adapted to specialized applications and are outside the scope of this paper. Batch processing was found to have severe limitations for users who made heavy use of the computer, and discouraged those who would have run a few small problems had they had easy access. User's programs are typically held on paper tape or punched cards and output is via a line printer. Being electromechanical, these devices are necessarily slow compared with magnetic devices such as disks and tapes. This, and the red tape that builds up around batch processing systems means that turnaround time can be frustratingly long, and that program development time gets stretched out. Another fault is that batch methods kill any idea of man-machine interaction, since jobs are run in a continuous stream and all the user can do is goggle through a plate-glass screen as his program stops through some simple error. Working with the computer, the user can make the computer aid him in debugging his program, as everyone who has used a single-user computer will realize.

### Development of Multiaccess

Multiaccess systems were developed to combat these deficiencies. With multiaccess a number of users can communicate directly with the computer's operating system without necessarily being aware of any other users. This

has been achieved at the supervisor level firstly by straight timesharing. This technique involves having several user programs in a runnable condition held on a backing store, but only having one program resident in fast memory along with the supervisor program. This program continues execution until one of four conditions comes up:

1. The program ends.
2. The program halts on an error.
3. The program wants to do I/O.
4. The current time slice is exhausted.

At this stage the complete current status of the program would be written from fast memory to backing store and a new program retrieved from backing store and put into execution. The earliest example of a multiaccess system was the CTSS (Compatible Timesharing System) and was of this primitive form. CTSS was developed under Project Mac at MIT (Massachusetts Institute of Technology) on an IBM 7094, which was one of the foremost batch processors of its day. Soon after this, the first multiaccess system for a DEC computer was written. Stanford University wrote a system very similar to CTSS for the PDP-1. This accommodated only 5 users.

The next step was to add multiprogramming to the basic system described above. The difference here is that several programs can coexist in the fast memory and that the supervisor can switch between these programs as the four conditions, listed above, occur. There is no need to swap programs to and from the backing store, but this approach requires a large main memory. This feature existed before the days of multiaccess, when several computers were built with multiprogrammed batch systems; an example is the ICL Titan computer at Cambridge University, which has 128K of fast memory and could accommodate up to 20 user jobs (depending on size) in the fast memory. There will still be programs that cannot be accommodated in the memory and these are kept on backing store as before, but there are always a sufficient number of programs in main memory to have one of them executable - hopefully.

### Implementations

With a given computer system, there are two ways in which multiaccess can be implemented. Either an existing system is taken and modified by the addition of a terminal servicing program, as with CTSS and CMAS, or a supervisor is written specifically with multiaccess in mind, having the terminals as a basic unit in the system, as done in the DEC/PDP-10 monitor and the IBM/360 TSS. In the PDP-10 monitor, the system revolves around terminals, and batch mode is added more or less as an afterthought, thus reversing the order of priorities in the Cambridge and MIT systems.

Multiaccess has a large overhead of supervisor program and this usually implies a large scale computer system for its effective operation. However, some computer bureaus have begun operation using a minimal computer (e.g. a PDP-7) and a large fast backing store. This enables them to do a large number of relatively small jobs (of the sort submitted by remote users with quick problems) slowly but effectively and cheaply.

## Requirements and Features

To operate a multiaccess facility successfully it is necessary to possess a system that includes a large fast-access backing store, for example a drum or disks, for quick retrieval of filed information. The PDP-10 at the University of Queensland has two levels of fast backing store, with fast fixed-head disks (with one read/write head per track) and slower moving-head disk packs (with one read/write head per surface).

A batch processing service needs to provide systems for compilation and execution of programs written in a variety of languages, and very little else. It leaves the programmer to make any necessary changes in his source program by correcting his deck of punched cards. Multiaccess with its close man-machine interaction is much more versatile, and this makes machine assisted program debugging possible. The multiaccess system must have some means of putting information into files kept on backing store, which is normally the slower disks or magnetic tapes; and of altering the information once it is in the files. It must have a means of manipulating files, of copying and listing them, and of transferring them from one type of store to another.

A number of other useful features are possible because of the interactive nature of multiaccess. Some of these are:

- (i) so called conversational compilers, which compile code as soon as it is received, allowing incorrect lines to be altered without recompiling the whole program.
- (ii) dynamic debugging by monitoring the performance of the program during execution, and thus following the consequences of individual instructions in a step by step fashion.
- (iii) backup for disk based files in case information on the disk gets accidentally destroyed, which is only too easy to do. This is done by writing files onto magnetic tape as well as the fast backup store, but only using the tapes for retrieval of lost information.
- (iv) simultaneous update and lookup of data bases by many different users. An example of such a data base would be the accounting statistics in a computer centre.

## The Cambridge Multiaccess System

This system was developed by adding to the multiprogrammed batch processing supervisor already existing on the ICL Titan. Thus many of the facilities described below already existed before multiaccess was added. Of the 128K of fast memory attached to Titan, 80K is available for user programs. This gives some idea of the scale of the total operating system, although much of the extra space is used for I/O bufferage. The 80K user section is divided into two halves, and a user program can only use one of these 40K slots. Hence there are always at least two programs in the fast memory at once. The system is capable of accommodating 16 terminal jobs at once and

there are a large number of teletypes scattered round the university departments. Since only 16 of these terminals can be communicating with the operating system at one time, there is a problem selecting which 16 users are to be allocated to these jobs. When a user attempts to log in he may try giving himself one of four priorities, if he has been authorised to use them. If he uses a high priority and the system is full, the supervisor will select a lower priority user and give him notice to quit. Of course, only a limited number of people are authorised to use limited amounts of the higher priorities. This is effective in spreading the workload fairly evenly throughout the day. Statistics show that there are still definite peaks of usage around 12 noon, 3 p.m. and 11 p.m.(i) but that the workload is indeed spread and it is rare (apart from the early hours of the morning) to find many of the 16 jobs idle.

There are two modes of operation, called 'normal' and 'expensive', the latter because it really is expensive and places a heavy load on the system. Normal mode does not allow true interaction in the sense that a user can communicate with a program in execution. Expensive mode does allow this. Thus, as nearly all work is done in normal mode, the system is largely one for editing and running batch type jobs from a terminal, although the directness of contact improves matters immensely. There is no dynamic debugging facility as this requires full interaction. The overall approach is still slanted towards batch processing and when several big background batch jobs are running, teletype response can get very slow. At times it is possible to wait for several minutes for a response.

System crashes are infrequent, but when they do occur it's amusing to sit and watch a roomful of teletype users all furiously pressing buttons waiting for the instant that the terminal becomes 'live' so that they'll be among the first 16 users to log in.

The I/O to devices other than teletypes is all stream orientated and is extremely well organized. This enables a terminal user to imagine that he can use an I/O device directly and further that there is a virtually unlimited number of each device available. In fact, all I/O files are sent to the disk and then have to queue up to use the required physical devices when they become available. Thus a user at a terminal who had asked for a file to be listed might have to wait for an hour for his output to come off the line printer during peak operating hours. This is one feature that has come straight over from multiprogrammed batch.

Batch and terminal systems use different sets of commands that are complicated and inconsistent and hence it is difficult to learn the full range and to use them effectively. Owing to the lack of normal mode interactive facilities, the nonsupervisor software is mostly identical with the batch software, with the addition of an editor, and a method of taking a teletype originated task and adding it to the batch queue so that a user isn't bound to wait for the results of a long calculation to come back to his terminal.

## DEC PDP-10

The PDP-10 timesharing monitor is designed specifically for remote terminal usage. It occupies about 18K of the 48K fast memory available in the U.Q. configuration. 24K is the limit for a single user program, so that with only 48K of fast memory there is no guarantee that two or more runnable programs will be resident in memory at any one time.

There is no priority system, so that if there were more terminals than jobs some users would find it extremely difficult to get use of the system. At present there are 16 jobs available at any one time and only 16 terminals will be released. However, this can be expanded and experimentation will eventually determine the maximum for a reasonable computing service. A study of the Project Mac system indicated that a severe degradation set in with more than 30 jobs, but this would be largely system dependent. All jobs run in a mode equivalent to the 'expensive' mode at Cambridge and all programs are thus capable of interacting with the user at a terminal. There is a batch processing system which will in due course run a single stream of batch jobs concurrently with the terminals.

At the time of writing, I/O is not well organised in the PDP-10 system and I/O devices such as the line printer are not freely available to the terminal user. In fact terminal users are restricted to using their terminals and disk files. All activities requiring the use of the card reader and line printer will have to be run through Batch.

Batch and terminal processing will use the same set of commands which means that Batch will have a considerably extended use and it will be possible to create and list files as well as to execute them using the batch system. Similarly, all the present batch control cards will be available in an extended form through the terminals. The staff of the Computer Centre have concentrated on providing a system that is easy to use and the command language is straightforward and coherent compared with the ad hoc and unnecessarily complicated approaches of many other systems.

Since the system is interactive the Computer Centre can provide BASIC, an interactive programming language, and in due course a dynamic debugging package.

## Conclusions

The greatest advantage of multiaccess is that files are almost instantly available for editing and execution and that the results come back direct to the user at his terminal instead of waiting inside the computer room, to be delivered to the user outside. This is reckoned to cut down program development time by factors of as much as five in real time, and to reduce frustration a hundredfold. However, the amount of computer time used to solve a given problem increases because of a tendency to give the program

another whirl before thinking hard enough about the cause of a bug. A teletype is no substitute for thought. After a period sitting at a terminal concentrating on the same program and listening to the raucous chatter of teletypes, there's a marked increase in this tendency and the approach becomes slapdash. Consequently, a user should always plan, before the terminal session, just what he intends to do so as to get the maximum out of the computer. He should carry out only the planned programme, log out and analyse the results. This requires considerable strength of mind.

Computer (CPU) time is getting cheaper as against I/O charges and it is possible that buying more CPU time (and less real time) in a terminal session, will soon be cheaper than putting a tray of cards into a reader and receiving a large wad of paper from the printer. And we haven't begun to measure the cost of the programmers' time and happiness.

Multiaccess systems still have many detractors, but in terms of extracting the maximum use from a computer system, I believe that they are way ahead of the old and relatively inflexible batch approach. The exception is the performance of long routine jobs involving fully developed programs without dynamically changing data bases. The range of applications for computer systems is taking off now, and multiaccess systems are going to be mandatory for program development and data manipulation of the near future. Because of the pace of the technological advances the further future is unpredictable.

#### References:

A Timesharing Debugging System for a Small Computer  
S. Boilen *et al*  
Spring Joint Computer Conference 1963

The Design of Multiple Access Computer Systems  
M.V. Wilkes and R.M. Needham  
Computer Journal 1967 and 1968 (2 papers)

System User's Guide MNT-8 (in preparation)  
University of Queensland Computer Centre.

INDEX OF COMPUTER CENTRE BULLETIN VOLUME 3

		page references
A	ADMINISTRATION OF THE COMPUTER CENTRE	
	Consultation with programmers	14
	Data Preparation bookings	75
	Organizational distinction between Computer Science and Computer Centre	44-45
	Revised charges for computer services	14
	Service enquiries	46
	Use of card punches	46
	ARTICLES	
	Application of computers to Biometry in animal production	4-12
	Features of Atlas, Hardware and Supervisor	80-83, 87-90
	Multiaccess Computer Systems	99-104
	PDP-10 absolute overlays	55-66, 73
	Problem solving by simulation	33-36
	Student fee assessment	24-27
	The simulation of mineral treatment processes	15-19
	ATLAS	
	Features of ATLAS - Hardware and Supervisor	80-83, 87-90
B	BULLETIN BAFFLER	24, 32, 41, 50, 67
C	CARD FORTRAN	2
	COMPUTERS	
	Application of ~ to Biometry in animal production	4-12
	COMPUTER SCIENCE COURSE	94-95
	CONSOLE SWITCHES	52
F	FACILITIES RELEASED FOR THE PDP-10	
	Disks	98
	FORTRAN compilation options:	
	<i>NOLIST</i> , <i>MACRO</i> , <i>CREF</i>	30
	<i>NOBIN</i>	55
	Loader options:	
	<i>NOMAP</i> , <i>SYMBOL</i>	31, 55
	New version of BATCH	54, 78
	PDP-10 absolute overlays	55-66, 73
	Remote terminals	98
	Scratch files	42-43
	FILE SEPARATORS	73

F	FORTRAN IV FOR THE GE-225	
	Array references in subroutines	79
	Card reader misfeeds	52
	Truncation of real numbers	49
	Variable and array equivalencing	92
	FORTRAN IV FOR THE PDP-10	
	CREF option	39, 55, 73
	Common sub-expressions	78
	Comparison between a complex function and a real quantity	93
	Constants in subroutine calls	53
	Data input in programs	39
	Deep parentheses nests	73
	Definitions in Data Statements	92
	Identifiers as subprogram names	39
	Integer exponentation in real expressions	49
	NAMELIST facility	39
	Printing of double precision values	54
	Reading free field input	71
	Subscript calculations within expressions	53
	Truncation of real numbers	49
	Use of registers when evaluating expressions	53
	Value of DO loop index	72
	Variable array dimensioning and Arithmetic Statement Function	84
	Variable redefining a function	72, 93-94
	Width of output field	93
I	INDEX OF VOLUMES 1 AND 2	vol.2 pp.167-169
L	LIBRARY ACCESSIONS	see U.Q. Library Accessions also 3, 15, 31, 40, 47, 66, 70, 85, 95
N	NEWSLETTER	75
O	OPERATIONS SCHEDULE BOARD	75, 84
	OVERLAYS	
	PDP-10 absolute ~	55-66, 73
P	PROGRAMMING ADVICE	
	A key to FORTRAN debugging	22-23

P	PROGRAMS	
	PDP-6 Programs:	
	<i>Principal Components</i>	77
	<i>Canonical correlation</i>	77
	<i>Multiple regression</i>	77
	<i>Multiple discrimination</i>	78
S	SCRATCH FILES	42
	SIMULATION	
	The ~ of Mineral Treatment Processes	15-19
	Problem Solving by ~	33-36



BULLETIN MAILING LIST

The Bulletin is distributed according to two mailing lists, one for organizations and departments which is permanent, and the other for individuals, maintained on an annual basis.

To receive Volume 4 of the Bulletin, please complete the appropriate section of the following form, and return it to:

The Editor,  
Computer Centre Bulletin,  
Computer Centre,  
University of Queensland,  
ST LUCIA,  
Queensland, 4067.

---

Application for Individual Circulation      Application for Organizational Circulation

NAME: ..... POSITION: .....

POSITION: ..... ORGANIZATION: .....

ADDRESS: ..... ADDRESS: .....

.....

.....

NUMBER OF COPIES: .....

---

SIGNED: ..... DATE: .....

---

