UNIVERSITY OF QUEENSLAND
COMPUTER CENTRE

COMPUTER CENTRE BULLETIN

Volume 3, Numbers 5 & 6.
1st June, 1970.

Editor:
S. J. Barry.

THIS EDITION

The resignation of Mrs. H.L. Smythe meant that April's Bulletin was behind schedule. Apologies are made for this delay. The May and June issues of the Bulletin will be combined in this edition. I would like to remind all readers that contributions, whether articles or Letters to the Editor, will be most welcome.

Items include a notice about forthcoming seminars, some errors in the operating system, new facilities for the PDP 10, a list of recent library accessions and another Bulletin Baffler.

STAFF OF THE COMPUTER CENTRE

We would like to welcome *Mrs. Beverley McLean* as Program Librarian and *Mrs. Sarah Barry* as Technical Writer.

Bev graduated with a B.A. from the University College of Townsville in 1965 and received her Diploma of Librarianship in 1966 from the University of New South Wales. She is a member of the Library Association of Australia.

Sarah graduated with a B.Sc. from the University of Sydney in 1967.

SEMINARS IN COMPUTER SCIENCE

A further three seminars will be sponsored by the Department of Computer Science in second term. The aim of these seminars is to present some of the newer techniques that are emerging in the computer field. Any enquiries or suggestions should be directed to the Seminar Convenor, Mr. John Williams, extension 690.

Although the seminars are open to the public, each speaker, however, will assume the audience has some general knowledge of computing, although no knowledge of the specific subject area would be necessary.

The three seminars will be held in

Room G13, Engineering Administration Building

commencing at 2.00 p.m. and finishing no later than 4.00 p.m.

The first seminar will be held on

WEDNESDAY, 24TH JUNE, 1970

The speaker will be Mr. R.D. Nilsson, Senior Lecturer in the Department of Civil Engineering who will discuss *Dynamic Loading - an Overlay System for the PDP 10.* The following abstract outlines the theme of the seminar.

Dynamic loading of programs is referred to in various programming systems as overlaying, chaining, linking etc. and refers to the splitting of a program into segments that may use the same area of core at different times, the process being controlled by the program itself.

Such systems are necessary in the PDP-10 when a program is too large for the core storage available. They may also be useful for minimising the users core requirements resulting in an improved time sharing response time and a reduced core charge.

The problem will be discussed in general and details will be given of an absolute overlay system for the PDP-10 which will give at least the same facilities as are presently available on the GE 225.

The second seminar for the term will be held on

WEDNESDAY, 8TH JULY, 1970

The speaker will be Mr. M. McLean, Senior Demonstrator in the Department of Computer Science, who will discuss *Features of Atlas - Hardware and Supervisor.*

The third seminar will be held on

WEDNESDAY, 29TH JULY, 1970

The speaker will be Mr. J.D. Noad, Senior Systems Analyst in the Computer Centre, who will discuss *The PDP 10 Accounting System.*

SYSTEM ERRORS

Users are asked to take note of the following errors in the Operating System. These errors passed unnoticed in recent changes to the system but will be corrected in a subsequent update.

1. The CREF option in FORTRAN will at present suppress the entire listing if any syntax errors are detected during the compilation. It is therefore advisable not to use this option until all obvious syntax errors are removed from a program.

2. The NAMELIST facility does not work, due to the improper definition of some global symbols.

3. The following identifiers cannot be used as subprogram names because of their improper inclusion as global symbols in the FORTRAN operating system.

```
ILL      LEGAL    IFIX     IFIXB    TNAME    INIFLG
HDRADD   TTYLPT   DOLFLG   RERDFL   INPDEV   DEVNUM
BFDAT    LASTFL   TEMP     FOBPDP   LASTLP   SAVEAC
SAVFAC   DNAME    RPTR1    RPTR2    RCNT1    RCNT2
LINBUF   LINHDR   REL      INT      BUFHDR   ITMCNT
EXIT     BUFBLT   FILNUM   DEVNAM   DIGPTR   GRPRPT
SAVSCN
```

DATA INPUT IN FORTRAN PROGRAMS

There is a peculiarity about the input of data in FORTRAN programs. This oddity has been remarked upon before but, because of recent enquiries received by the Computer Centre, the following comments will be restated.

There exists a discrepancy between the BATCH controlling program and the FORTRAN operating system. BATCH will suppress trailing blanks to decrease the amount of input. The operating system, however, does not always handle these preshortened records correctly. Portions of the next record are sometimes used as part of the record being interpreted; the result being that subsequent reads get out of step.

This error is very context sensitive, but can be avoided (particularly with A format) by ensuring that the last field requested is terminated, or followed by, a non blank character.

LIBRARY ACCESSIONS

This section lists the books on Computer Science that were acquired by the Libraries of the University of Queensland in March 1970.

Gallagher, Robert G. *Information Theory and Reliable Communication.* 1968. (001.539 GAL. Elect.Engin.)

Systems Theory Research. 1968. (Qto 001.53 SYS. Engin.Lib.)

Pritchard, Alan. *A Guide to Computer Literature.* 1969. (016.6518 PRI. Main Lib.)

Logic and Foundations of Mathematics. 1968. (164 LOG. Mathematics Lib.)

Coplin, William D., ed. *Simulation in the Study of Politics.* 1968. (320.0184 COP. Main Lib.)

Stone, M.G. *Computer Privacy.* 1968. (323.44 STO. Main Lib.)

Suppes, Patrick Colonel. *Computer-assisted Instruction.* 1968. (372.7018 SUP. Engin.Lib.)

Flores, Ivan. *The Logic of Computer Arithmetic.* 1963. (510.78 FLO. Engin.Lib.)

Gnedenko, Boris Vladimiroovich. *Introduction to Queuing Theory.* 1968. (519 GNE. Maths. Lib.)

Sarkisian, Artem Sarkisovich. *Theory and Computation of Ocean Currents.* 1969. (551.4701 SAR. Maths.Lib.)

Bashkow, Theodore R. *Engineering Applications of Digital Computers.* 1968. (620.0018 BEC. Engin.Lib.)

Blaquiere, Austin. *Nonlinear System Analysis.* 1966. (620.7 BLA. Maths.Lib.)

Davis, Samuel. *Computer Data Displays.* 1969. (621.3819532 Engin.Lib.)

Digital Equipment Corporation, Maynard, Mass. *The Digital Logic Handbook Flip Chip^{tm} Modules.* 1969. (621.381958 DIG. Engin.Lib.)

Lewin, Douglas. *Logical Design of Switching Circuits.* 1968. (621.381537 LEW. Engin.Lib.)

Wolfendale, E. *Computer-aided Design of Electronic Circuits.* 1968. (621.3815 WOL. Elect.Engin.Lib.)

Barron, David William. *Recursive Techniques in Programming.* 1968. (651.8 BAR. Main Lib.)

Fletcher, Richard N. *Computer Science for Management.* 1967. (658.1 FLE. Main Lib.)

Horngren, Charles T. *CPA Problems and Approaches to Solutions.* 1969. (657.076 HOR. Main Lib.)

Jordain, Philip B. *Condensed Computer Encyclopedia.* 1969. (651.8 JOR. Main Lib.)

Moore, Peter Gerald. *Basic Operational Research.* 1968. (658.4 MOO. Maths.Lib.)

Mumford, Enid. *Computers.* 1968. (658 MUM. Main Lib.)
Nato Advanced Study Institute, Villard-de-Lans, 1966. *Programming Languages.* 1968. (651.8 NAT. Engin.Lib.)
Smith, Wilfred Nye. *Integrated Simulation.* 1968. (658 SMI. Main Lib.)

BULLETIN BAFFLERS

We hope that all our readers received the Addendum to last month's Bulletin, and did not waste many hours puzzling over the incorrect version of the Bulletin Baffler.

The answer to last month's Baffler is:

The values output are 4 and 5

The Reason

Output (ADD (INCV(x))) causes ADD to be entered. This makes two successive calls on INCV. Since the argument in INCV is called by value, a value of 2 will be returned on both calls.

Output (ADD (INCN(x))) also results in ADD calling INCN twice. However, the argument in INCN is called by name. The first call will set variable 'a' to 2 and return a value of 2. The second call therefore sets 'a' to 3 and returns a value of 3.

This month's Baffler is:

What value is printed by the following?

```
begin

procedure P(n,L1,L2) integer n; label L1,L2;
   begin if n > 0 then P(n-1,L2,L3); L3; go to L1; end;
P(10,A,B)
A: output (1); go to C;
B: output (2);
C: end
```

The solution to this Baffler will be given in next month's Bulletin with the last one of this series.

NEW FACILITIES ON PDP 10

As from Monday, 25th May, 1970, scratch files will become available to PDP 10 users.

A scratch file on the PDP 10 is similar in concept to the scratch tapes on the GE 225. The file can be written and read during the job, but the information on the file is *not* saved at the end of a job. As with scratch tapes, a scratch file is a sequential file which can be 'backspaced' or 'rewound', and the information on the file is retained until it is overwritten or the job terminates. The only difference between the two is that the scratch file is on disk and so is always available.

There are four scratch files presently available in the system. These correspond to the logical FORTRAN device numbers 10, 11, 12 and 13. To reference a scratch file, a user need only WRITE to and READ from one of these logical unit numbers.

A total space of 128K words is available on the disk for scratch files. A user may use this space for his scratch files as he wishes. For example, one job may use one scratch file of 128K words while another may use four scratch files each of 32K words.

There are 2 modes of recording information on scratch files, namely, BINARY or ASCII. If information is written to the file under FORMAT control, then the file is an ASCII file. If the output is unformatted, the file is BINARY.

EXAMPLE

(i)
```
      WRITE (10) I,J,K,X
```

File 10 will be a BINARY file

(ii)
```
      WRITE (12,2) I,J,K,X
    2 FORMAT (3I6,F8.2)
```

File 12 will be an ASCII file

Files must be read in the same mode as they are written.

A BINARY *logical* record is the result of one WRITE statement and consists of at least one physical record, no matter how little information is output by the WRITE statement. An ASCII *logical* record is one line (i.e., <CR>.... <CR>). These are packed into physical records. One WRITE statement may output several logical records as the result of /'s in the format statement.

As well as WRITE and READ, the instructions BACKSPACE and REWIND operate on scratch files. BACKSPACE will always backspace the file one logical record. REWIND initiates the file to start the next operation from the beginning of the file. It should be noted that files can only be backspaced when the last data transfer was a READ operation. Any attempt to backspace when the last data transfer was a WRITE operation will result in an appropriate error message and an EXIT from the program.

Finally, users must note that, at the moment, no charge is being made for use of files. When suitable accounting system facilities become available, a charge may be introduced for scratch file input/output.