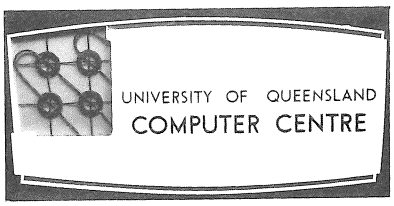

COMPUTER CENTRE BULLETIN

Vol. 5 No 3
6 March 1972

Editor:
Mrs Sarah Barry

LOW PRIORITY BATCH

[WN-78]

1. INTRODUCTION

On Monday 6 March, the Computer Centre introduced a 'low priority' batch processing service. This new service means that work submitted for low priority batch processing on any day will generally be processed overnight and be available for collection on the following working day. Such work will attract a 20% discount off the normal charges.

The Centre has two aims in introducing this service:

(a) to provide a computing service for teaching purposes at the lowest possible charge,

(b) to encourage users with large and costly jobs to process their work via the batch processing service in off-peak periods.

Naturally any user may make use of this service.

2. RUNNING OF LOW PRIORITY JOBS

New job identification cards and a number of special card input trays have been provided for low priority batch work.

2.1 Job Identification Cards

Users who wish to run their work via the low priority batch service should place a new 'LOW PRIORITY' job identification card on the front of their deck, instead of the normal purple job identification card. The new job identification cards are white with green printing and green top edge, and are clearly labelled LOW PRIORITY as shown overpage:

B5-3
8Mar72

0004485

UNIVERSITY OF QUEENSLAND COMPUTER CENTRE

PDP 10 JOB IDENTIFICATION CARD

Project Number ..... Date .....

Authorised User's Name ..... Phone .....

Authorised User's Signature .....

The present purple job identification cards are retained and should be used for all standard priority work.

2.2 Input Card Trays

Two sets of input card trays are now available for PDP-10 batch work. Low priority batch work should be placed in the trays labelled:

PDP10
LOW PRIORITY

The existing input trays are to be used for standard priority work.

3. CONDITIONS OF SERVICE

The new low priority batch service is provided under the following conditions:

(a) no low priority work is processed before 8 p.m.

(b) standard priority work takes precedence over low priority work at all times

(c) any low priority work remaining unprocessed when the Centre shuts down is held over until the following evening

It is proposed that these arrangements will be reviewed with the introduction of third shift operation within the Centre.

BATCH AND LOGIN

[WN-78]

New versions of Batch and Login have been implemented to allow low priority batch processing.

STUDENT ACCOUNTING SYSTEM

[WN-76]

1. IDENT AND QUIT

IDENT and QUIT commands now clear the loadlist and error count so that a student job is unaffected by uncompleted tasks and compilation errors in previous student jobs.

[WN-78]

2. INCORRECT IDENT COMMANDS

If an IDENT command for a student job is missing, or contains a syntax error which prevents it being processed as an IDENT, that student job will become part of the main job with the balance of the job limit as the only cost limit in force. This could result in large expenditure by an uncontrolled student job.

The solution to this problem has been to modify QUIT so that, instead of clearing the cost limit, it will now reset the student cost limit to the value specified by the latest SACS command.

Thus if the main job is to include any processing with a cost limit other than that set by the initial SACS command, the administrator should insert a second SACS command, defining the

new limit, at the appropriate point in the job. This command must be followed by a QUIT to set that limit. This limit may be greater than the balance of the job limit without being rejected as it would be by an IDENT. A QUIT has the function of clearing the loadlist and job error count and setting the student limit, whether or not there was a corresponding IDENT.

Note that all effective SACS commands will be echoed as part of the main job output.

ASCII FILE FORMATS

[WN-75]

It is an unfortunate fact that there is not compatibility between the formats required by various programs for ASCII files. The following note sets out the various formats that are required and the best way of achieving them. It is necessary to know that disk file storage is based on a physical record or block of 128 words. A file will be created as an integral multiple of such blocks and it may be that portion of the last block is unused. Eight such blocks make up the kiloword that is the basis of estimation of file storage and I/O.

1. SOURCE PROGRAMS AND FORTRAN INPUT DATA FILES FOR SEQUENTIAL ACCESS

Each logical record or card image is terminated by the characters carriage return (15_8) and line feed (12_8). The subsequent logical record may follow immediately, without any intervening null characters and without regard to the physical block structure; that is a logical record may be split between two physical blocks. Any inserted null characters are ignored and will be squeezed out on input. Thus files created in any fashion and subject to the limitations described below may be used in these cases. Files without inserted nulls are created by the Editor (i.e. EDIT or CREATE) and by COBOL for sequential access output files.

2. RANDOM ACCESS FORTRAN INPUT FILES

Each logical record should be of fixed length and is terminated by a carriage return and line feed. The logical record is padded

out to a full word boundary by the insertion of null characters after the <cr> and <lf> (each word can hold 5 characters). Logical records may not be split over a block boundary.

Files for this mode of use should be created by FORTRAN using random access output or by sequential access processing with a fixed length logical record. The use of Editor to modify a file so created will remove the inserted null characters discussed above and will also truncate any logical records that have trailing blanks.

3. RANDOM ACCESS COBOL INPUT FILES

Each logical record should be of fixed length and is terminated by a carriage return and line feed. Each logical record will start a new physical block, the previous block being filled with null characters. The most effective way of creating such a file is with COBOL, using random access.

Note that this mode of usage can be very wasteful of file space, unless consideration is given to the size of a physical block in planning the structure of the logical record. In this mode a physical block can hold 638 characters.

COBOL MANUAL

[WN-77]

The COBOL manual, MNT-13, is now available from the Computer Centre. The cost of the manual is $4.20.

PDP-10 COBOL

[WN-77]

1. TELETYPE INPUT/OUTPUT

There are two methods of addressing the teletype in COBOL:
(i) READ and WRITE
(ii) ACCEPT and DISPLAY

The following points need to be noted when using the teletype:

(a) READ (from the teletype) and WRITE (to the teletype) may both be used in the same program, provided the input and output teletype files are not open simultaneously.

(b) There is nothing to prevent the use of any combination of all four verbs in the one program, provided point (a) is observed. However, special care must be taken with the WRITE verb. It is dangerous to follow it closely with either a DISPLAY or an ACCEPT, as they can interrupt data output via the WRITE verb.

If WRITE, and DISPLAY and/or ACCEPT verbs are being mixed in the same area of program, it is safest to CLOSE the output file before using DISPLAY or ACCEPT.

(c) After a WRITE statement, COBOL always does a carriage return. Therefore, it is best to say

```
WRITE record-name BEFORE !
```

so that the teletype does a carriage return - line feed.

With the statement

```
WRITE record-name AFTER !
```

the teletype positions itself at the beginning of the line just output.

2. DISPLAY VERB

The COBOL manual does not state any restriction to the number of characters that a user may care to DISPLAY. However, the size of the field to be displayed is stored in a field !Ø bits long. The maximum number of characters that can be displayed is thus !Ø24 characters. Field sizes greater than this will be truncated to !Ø bits. For example, only 776 characters will be displayed from an !8ØØ character field.

The field named in a DISPLAY statement can be defined in either the working-storage section or the file section. This is also true for the ACCEPT statement.

3. COBOL ERRORS

(a) Condition names

Condition names may not work correctly in some instances. For example, the following coding

```
77  N-1          PIC 9(10)   USAGE COMP.
88  N-1-C        VALUE 0.
.
.
.
IF N-1-C GO TO . . .
```

does not create correct code.

COBOL THROUGH BATCH

[WN-78]

Users submitting COBOL jobs through batch must specify the name of the source file on the COBOL command. There is an error in the present decoder so that commands such as

```
.COBOL(LIST)
```

will not assume the source program to be coming from the job input device.

Users must first copy their source program to disk and then compile the source file

```
.COPY TO=SOURCE/CBL
  -
  -
  -
.COBOL(LIST) SOURCE/CBL
```

RUNOFF

[WN-77]

RUNOFF is a PDP-10 program that assists in the production of memorandums and documents. The material to be documented is prepared as an ASCII file. Runoff takes this file and produces a final copy that is formatted, numbered, titled and generally well presented. The advantage of this method of reproduction is that changes and amendments can be quickly and easily carried out on the source file and a new copy produced, without extensive

retyping of the document.

The Computer Centre is endeavouring to prepare as much of its documentation this way as possible. Users may have noticed that the Newsletter and the Bulletin are now being prepared with the help of Runoff. A manual on Runoff is currently in preparation and it is hoped that it will be released soon.

BASIC ERRORS

[WN-77]

There are a number of typographical errors in the Basic manual, MNT-9. These will be corrected in due course when a revision is brought out for the manual.

(a) page 1-8, line 4

10*X↑N should be 10↑N*X

(b) page 4-1

After the lines

NEW↵

if you are going to create a new program.

the words: 'BASIC responds with the following: NEW OR OLD--' are superfluous.

(c) page 4-4

Before section 4.6, the reentry command, '.BAS↵' has been omitted.

(d) page 5-2, line 2

10*X↑D should be 10↑D*X

(e) page 8-5, line -8

the line '(-<-YZ' should be ' (-<-YZ'
(that is the first character on the line should be a space).

COPY(COMPRESS)

[WN-74, 75, 78]

The following comments should amplify the use of COPY(COMPRESS) from both remote terminals and via Batch.

(a) Batch truncates trailing blanks on all input records, giving some degree of compression.

Thus, use of COPY within Batch will create the minimum sized file without deleting sequence numbers, (i.e. .COPY TO=FILEA).

(b) The COMPRESS option is intended to replace sequence numbers in columns 72-80 with blanks and then suppress trailing blanks.

The error reported in WN-74 and WN-75 where some records were being improperly handled and were merged with succeeding records has now been corrected.

EXCESSIVELY LONG COMMANDS

[WN-75]

Users should note that unduly long commands issued from a remote terminal can cause problems. The problem is caused by the fact that the operating system has only a finite sized buffer to hold the command being decoded and any characters that will not fit are discarded. Thus, while it is quite acceptable to have comments (preceded by a semicolon) going up to column 80, commands to be performed are truncated at column 78.

LIBRARY ACCESSIONS

The following books may be of interest to Bulletin readers.

CLARK, Frank James — Accounting programs and business systems 1971 (HF5661.C56 Main)

FOX, Leslie, ed.	Advances in programming and non-numerical computation 1966 (651.8 FOX Engin.)
RUDWICK, Bernard H.	Systems analysis for effective planning 1969 (658.5Ø2 RUD Engin.)
THOMPSON, Gerald E.	Linear programming 1971 (T57.74.T45 Main)

PDP-1Ø FORTRAN IV COMPILER ERRORS

[WN-78]

1. COMMON SUBSCRIPT EXPRESSIONS

Use of a common subscript expression as in

```
A(I,J)=B(I,J)
```

where A is real and B double or complex, will yield improper results.

A possible cure is to split the statement into two parts or to use an expression such as

```
     A(I,J)=SNGL(B(I,J))
or   A(I,J)=REAL(B(I,J))
```

2. MIXED MODE EXPRESSIONS

(a) expressions such as

```
complex = complex-subexpression + real-subexpression
```

may yield incorrect results. A cure is to replace such an expression by

```
complex =
complex-subexpression + CMPLX(real-subexpression,Ø.Ø)
```

(b) the statement

```
double = double/FLOAT(integer)
```

creates incorrect code. The statement

```
double = double/integer
```

however produces correct code.

JOBTIM

[WN-78]

A new function, JOBTIM, has been added to the FORTRAN library. JOBTIM returns the total CPU time since login in milliseconds.

example:

```
MSECS = JOBTIM(Ø)
```

the result is an integer number of milliseconds

NEW VERSION OF CREF

[WN-78]

A new version of CREF, version 45(12)-2 was implemented on 7 March 1972.

MAIL - A MAILBOX PROGRAM FOR THE PDP-1Ø

Chester Wilson

1. ABSTRACT

MAIL provides a facility for interchange of short messages between users of the PDP-1Ø at the University of Queensland. These messages may be directly typed in from a terminal, pre-typed on cards (for batch use) or paper tape, or be in the form of an edited disk file. A message may be up to 1ØØØ (decimal) characters long.

The date of implementation of the MAIL facility on the PDP-1Ø will be announced in the weekly newsletter. The MAIL system is

classified as type 4 software.

2. GENERAL DESCRIPTION

MAIL operates by setting up mail files for users on the special mail file area. A file is set up for each user for whom there is mail. This file is added to as more mail is sent to a user, and is deleted when the user collects all his mail.

When a mail file exists for a user, login will send a message 'THERE IS MAIL FOR YOU'. This message appears with the other login messages at the start of a job.

2.1 Examining Mail

The user's mail file, if it exists, is read in, and for each message the project number of the sender and the date of sending are typed out.

2.2 Collecting Mail

A user may, if he so desires, collect mail from all sources at once, or from a specific project only. The user's mail file is read in, and the requested messages typed out on his teletype. If he is collecting mail from a particular source, a new mail file is created consisting of the old mail file less the collected mail. Should there be no remaining mail, the mail file is deleted. Mail may be examined without deleting it from the box of messages.

3. MAIL COMMAND

The format of the MAIL command is as follows:

```
          BOX
          COLLECT
MAIL(     LOOK    ) {PROJECT=} project-number {FILE=}filename
          SEND
```

each option may be abbreviated to its first letter.

3.1 Options

(a) BOX

BOX is used for seeing what mail is available. This is the default assumption if neither option nor arguments are specified.

example:

```
.MAIL(BOX)<cr>

FROM        SENT
166         30-AUG-71
317         14-DEC-71
12          13-NOV-71

EXIT
↑C

.
```

(b) COLLECT

COLLECT is used for seeing what mail is available and deleting it from the mail file.

```
MAIL(COLLECT) {PROJECT=}project-number
     C         PJ=
```

The argument is optional and signifies that the mail sent by this project number is to be collected. If the argument is not specified, ALL is assumed and all mail available to the user will be typed out and his mail file deleted.

example:

```
.MAIL(COLLECT)<cr>

MAIL FROM PROJECT 100, SENT ON 30-SEP-71 AT 15:14

DEAR FRED,
          THE DEPARTMENTAL MEETING IS AT 3.30 PM ON
WEDNESDAY.
                         JEAN

MAIL FROM PROJECT 237, SENT ON 15-NOV-71 AT 12:36

FRED - HAVE BORROWED YOUR FORTRAN MANUAL.
```

```
          JOE

     EXIT
     ↑C

     .
```

(c) LOOK

The LOOK option is similar to COLLECT but it does not delete the messages from the user's mail file. It is preferable to use COLLECT as much as possible as otherwise the mail files could get untidy, large, and expensive to manipulate.

```
MAIL(LOOK) {PROJECT=} project-number
     L      PJ=
```

The project number is optional and if unspecified will be assumed to be ALL.

If ↑O is typed part way through a message, the remainder of that message will not be typed. Subsequent messages will, however, be typed unless they too, are interrupted with a ↑O.

example:

```
.MAIL(LOOK) 23<cr>

MAIL FROM PROJECT 23, SENT ON 5-NOV-71 AT 17:46

PLEASE CHECK YOUR FILE DIRECTORY, SOME OF
YOUR FILES HAVE BEEN ARCHIVED.
          COMPUTER CENTRE

EXIT
↑C

.
```

(d) SEND

SEND is used for sending mail to other users

```
MAIL(SEND) {PROJECT=} project-number {FILE=} filename
     S      PJ=
```

One argument is mandatory for this option; that is the project number to which the mail is to be sent. An optional

argument is the name of the file from which the mail is to be sent. If this is omitted, the job input device is assumed. The default argument order, if assignments are omitted, is assumed to be project and file.

MAIL recognizes the end of a message being transferred by one of the following:

(i) A double carriage return

(ii) A ↑Z character through terminals

(iii) A file separator card through Batch.

The message is transferred to the mail file belonging to the specified project.

examples:

```
.MAIL(SEND) 34<cr>

INPUT:
HAVE YOU SEEN THE DETAILS OF THE NEW PROGRAMS YET?<cr>
I HAVE ONLY JUST RECEIVED THE LATEST EDITION OF<cr>
THE MAGAZINE.<cr>
      D.B.<cr>
<cr>

EXIT
↑C

.

.MAIL(SEND) PJ=558 FILE=MESS/TXT<cr>

-SENDING-


EXIT
↑C

.

.MAIL(S) PROJECT=46 $ASR<cr>

INPUT:

EXIT
↑C

.
```

The argument FILE=filename is only necessary for the SEND option. The filename may be the name of any ASCII disk file, up to 1000 (decimal) characters long, or $TTY or $ASR. If a disk filename, it must be in the standard format of

name/processor-program

This argument may be preceded by the assignment FILE=. If the argument is omitted, the job input device is assumed; that is the terminal for remote terminal users and the card reader for batch users. $ASR can be used if the user wishes to input a file through the paper tape reader on an ASR 33.

4. ERROR MESSAGES

4.1 Syntactical

SPECIFY REQUIRED OPTION, PLEASE
: something was typed to MAIL but it was not a legal option in parentheses. This could also happen if the system is slow, and the user is typing ahead.

")" MUST TERMINATE OPTION
: option string was missing a closing parenthesis.

ILLEGAL OPTION: garbage
: option wasn't legal.

SPECIFY REQUIRED PROJECT, PLEASE
: if sending, a project number is a mandatory argument.

ONLY 1 PROJECT NUMBER, PLEASE
: on sending, only 1 project number is allowed. On looking or collecting, any extra project numbers are ignored, and this message will not occur.

ILLEGAL PROJECT: garbage
: projects must be legal decimal numbers up to (but not including) 100000. ALL is a legal project number on looking or collecting.

ONLY 1 FILE, PLEASE
: sending, can only send one file.

ONLY 1 ASSIGNMENT AT A TIME, PLEASE
constructions like PROJECT=PJ=32 are not allowed.

EXPLICIT FIELD MUST FOLLOW ASSIGNMENT
default assumptions are not used once an assignment has been specified.

ILLEGAL ASSIGNMENT: garbage
the only legal assignments are PJ and PROJECT (for sending, looking, and collecting), and FILE (for sending).

4.2 Errors

FILE TOO BIG - MAX IS 1000 CHARS
1000 characters is a little over 12 full lines.

FILE filename NOT FOUND
filename specified in sending has not been found. Files to be sent must be on the user's own disk area, with no group or project number specified.

NO MAIL AT PRESENT
user's mail file not found (COLLECT, LOOK, or BOX).

NO MAIL FROM nnn
specific collect or look project has not sent mail to this user.

CAN'T COLLECT NOW - ONLY LOOK
probably means that someone is writing mail to you while you are trying to collect mail, or that someone logged in on the same project is accessing your mail file.

GPO BUSY - PLEASE WAIT
the mail file you wish to access is busy. MAIL will try to access this file five times, each 10 seconds apart. If it succeeds, it will continue with the task. If not 'STILL BUSY - TRY LATER' will be typed, and MAIL will exit.

CAN'T DELETE MAIL FILE
all mail has been collected, and MAIL tries to delete the existing mail file, as it is no longer required. This will usually mean someone else is accessing the user's mail file also.

4.3 Errors Indicating System Problems

MAIL FILE OR FACILITY UNAVAILABLE.

? SCRAMBLED MAIL FILE

DISC ERROR

INPUT DEVICE ERROR

TELETYPE INPUT/OUTPUT ERROR

? DISC UNAVAILABLE
? TTY UNAVAILABLE

?? NULL FILENAME