UNIVERSITY OF QUEENSLAND

Prentice Computer Centre

NEWSLETTER

authorization: Director of the Centre

1   DISK QUOTAS

On the public structures DSKD and DSKE all users have been allotted
a logout quota of 400 blocks.   Recently, we have had problems with
shortage of disk space caused by users extending this 400 block limit
by copying files from one area to another to avoid an over-quota
logout.

This practice penalizes all users through lack of free disk space.
In future, when space becomes short on the public structures, any
files over the 400 block limit on any PPN will be transferred to
magnetic tape.   At the request of the owner, they may be recovered
at the cost of $5 per file.

2   NEW PLOTTING SOFTWARE

As part of its forward planning, the Computer Centre is providing
facilities for the connection of plotters to batch stations. Since
current format plot files would not be easily processed by the mini-
computers at the batch stations, the standard format for plot files
has been changed.   For some time, the plotter spoolers have been
able to correctly process files produced in both the current and
new formats.   However, on Monday 1st November, both PLO:CALCMP.REL
and PLO:CALF10.REL (the plotting libraries) will be updated to
contain the new software as standard.   The WINDOW program will be
updated at the same time. Currently, the new versions exist on the
NEW: directory.

Core images producing current or old format files should be reloaded
or the files produced by them processed by WINDOW to produce new
format files.   Any software which processes plot files will have
to be updated to process the new format files.

All the programs on the PLO: directory which produce plot files
have already been reloaded to produce new format files. At the
same time, some new items of software have been implemented.

2.1   SYMVU

SYMVU has been modified so that the FDATA field supplies a number
specifying the file produced by SYMAPs elective 21. The convention
used is the same as SYMAP uses, e.g. if the number 104 is specified,
a file MAP104.DAT is used.

2.2   VIEWPL

VIEWPL is a program to preview plot files on a TEKTRONIX terminal.
The program will accept plot files in either the current or new
formats. VIEWPL asks for the name of the plot file (without an
extension) and assumes an extension of .PLT. Then a scaling
factor is requested allowing the user to reduce or enlarge the plot
in order to fit on the screen.   Sample dialogue:

```
.R PLO:VIEWPL

VIEWING ON TEKTRONICS TERMINAL

FILE:   QET1   <cr>

FACTOR:   <cr>

<bel> C                     ;a '?' lists the available options

END OF PLOT                  ;type <return> and program will exit.
```

2.3   3-D Plotting

Subroutines to plot surfaces and contours are now available in
PLO:CALF10.REL the Fortran-10 plotting library.   Full details of
these subroutines and sample programs can be obtained by typing
or printing the file DOCH:CONTOR.MEM.

2.4   WINDOW

The version of WINDOW on NEW: converts new or current format plot
files to the new format. It will become the system standard on
Monday, 1st November.

Users requiring details of the new plot file format should contact
Arthur Hartwig at the Prentice Computer Centre.

3   NEW EDIT PROGRAM

The version of EDIT on NEW:, version 4(1)-2, replaces an earlier
test version for which an error was reported. The version on
NEW: differs from the standard version as follows:

(a)   Commands are checked to be correct abbreviations of valid
       commands.

(b)   Error messages are more complete and are preceded by "?"
       or "%".

(c)   Line-numbered files can be edited. The line numbers are
       ignored.

(d)   The DIRECT command produces FOROTS rather than FORSE direct
       access files.

(e)   A new RUN command is equivalent to FILE, followed by the
       last COMPILE, LOAD or DEBUG command.

Please test this version by setting NEW: in your search list.

       .LOGIN pj.pg /NEW

       or

       .R SETSRC

       *NEW ↑Z

Report any errors for correction before this becomes the standard
on 1st November.

4   FILE STORAGE

If you use our system regularly, you will have noticed that we
continue to run out of space on the public disk system - DSKD and
DSKE.   The inevitable result is that we have to purge them with
increasing frequency and to increasingly shorter periods (currently
less than 21 days).

Our strategy for making disk space available presumes a reasonable
degree of co-operation from you, our users, in that you won't use
file space if you don't need it.   In other words, you will delete
'garbage' and temporary files and archive or copy elsewhere files
you are no longer working with.

We realize it is inconvenient to find that some of the files you want have recently been purged, but the only criteria we have to decide on purging is the access date, and in most circumstances if you have not accessed it in 3 weeks, then the chances are that you are working on something else.

It appears that some users are not giving us much help and so we have to do more to maintain space than we desire. The result must be increased overheads, therefore worse response and some inconvenience to you the user. Besides purging infrequently accessed files, periodically also we delete the 'garbage' files from disk. The definition of such files is

*.CRF, *.TMP, *.MAP, JB????.CTL

These files are deleted because they are normally transient and their presence is usually caused by a crash of the system or a particular job.

*.LPT, *.PLT, *.PTP, *.CDP, *.LOG

These files are spooled and deleted as they would be on a successful logout.

It has been noted that some people are using various subterfuges to bypass the normal system limits. The Centre has taken action on one of these (refer first article in this newsletter) and will be instituting procedures to stop other abuses on logged-out quotas.

Similar space problems exist with the off-line file storage system. To hold a reasonable quantity of off-line files on disk, it is necessary to remove further 'garbage' files. In this case, the definition of garbage is somewhat wider and the following files will be removed

*.CRF, *.TMP, *.MAP, JB????.CTL

*.LPT, *.PLT, *.PTP, *.CDP, *.LOG

*.BAK, *.TEM, *.LST, Q??.CDR

Except for *.BAK, all of these files are transient in nature and should not be left behind by normally terminating jobs.

The last cleanup of the off-line system was initiated during the weekend of 9/10 October and will take several weeks to complete. If it happened that you had been using files of the types specified for a permanent function and would like to retrieve them on this occasion, please contact the Operations Manager.

5   EFFICIENT CREATION OF RANDOM ACCESS FORTRAN FILES

FOROTS and the FORTRAN compilers (both F40 and F10) allow the user
to specify a number of parameters for file access. If a parameter
is not specified, FOROTS assumes a 'reasonable' default value which
may not be the optimum value in a particular application. One
example is the serial creation of a binary file to be accessed
randomly.   Such a file cannot be created using sequential writes
and the FOROTS defaults, since FOROTS assumes the file consists of
variable length records and inserts special control words to assist
with file spacing operations such as BACKSPACE and SKIPRECORD.
This particular control word is not required if the file contains
fixed-length records (and hence does not appear in random-access
files).

The RECORDSIZE = parameter may be specified in the OPEN statement
to cause FOROTS to read or write a file of fixed-length records.
This is of particular advantage when serially writing a file for
subsequent random access.   For example, the program

```
      DIMENSION A(10)
      OPEN(UNIT=20,MODE='BINARY',RECORDSIZE=10,ACCESS='RANDOM')
      DO 10 I=1,1000
   10 WRITE(20#I)A
      END
```

is a program to serially create a file for subsequent random access.
However, creating the file read 2129 disk blocks, wrote 1065 and
cost $1.87.   1000 records were written and the file was under 100
blocks long.

The large number of reads and writes is because for every user WRITE
statement FOROTS

(i)   positions the file to the desired block;
(ii)   fills its input buffers (default is two hence two blocks
        read);
(iii)   updates the appropriate part of the buffer;
(iv)   writes the buffer back.

The additional blocks read and written are for monitor directory
operations.   Changing the Open statement to

```
      OPEN(UNIT=20,MODE='BINARY',RECORDSIZE=10,ACCESS='RANDOM',
           BUFFERCOUNT=1)
```

reduced the number of blocks read to 1087 and the cost to $1.34.

Changing the access to 'SEQOUT' and the write to a sequential WRITE
reduced the number of blocks read to 4, the blocks written to 100 and
the cost to $0.11.   The files created in the 3 cases were the same.

* * * * * *