UNIVERSITY OF QUEENSLAND

Computer Centre

NEWSLETTER

authorization: Director of the Computer Centre

1   DISK SYSTEM DEVELOPMENT

There has been substantial growth in the demands on the disk file
system resulting not only from the general growth in the number
of users but also from growth in user file sizes.   An additional
RPO3 disk drive (80,000 blocks) will shortly be brought into
service.   Disk I/O is also causing a systems bottleneck and
orders have been placed for an additional disk controller, a
channel and a memory port multiplexer.   The delivery of this
equipment, together with the implementation of a dual processor
system later in the year, will substantially improve this area.

2   LOGGED-IN QUOTA

It has been apparent that some users have been inhibited by the
present logged-in quota of 1,500 blocks.   It is proposed to
increase the logged-in (F.C.F.S.) quota to 2,500 blocks and to
provide a separate structure (DSKS) on which users may have a
logged-in (F.C.F.S.) quota of 10,000 blocks but no logged-out
quota.   This will enable users to manipulate temporary files
much larger than at present but special procedures which are
outlined later in this newsletter will be required.

3   LOGGED-OUT QUOTA

With the availability of an additional disk drive users may feel
it would be appropriate to increase the logged-out quota beyond
the current 400 blocks.   This has some disadvantages if
universally applied to such a large and diverse user population
that we have.   It would encourage rather than inhibit the storage
of unwanted files.   To make sufficient file space available,
the Centre may then be forced to reduce the archival time for
unaccessed files from 32 days to some lesser period.   The Centre
would be reluctant to do this as we feel users have probably by
now organised their usage around the current archival system.

(It should be emphasized, however, that the Centre does not provide
a guarantee regarding the purge period as special circumstances
may force moving files from online to offline areas). Rather than
increase the logged-out quota at this stage, we feel that we should
provide a means of immediately purging "rubbish files" (e.g.
unwanted log files, temporary files etc) unless explicitly protected
by users, and modifying the archival system so that archived files
are removed immediately from the users area and an improved response
is given to retrievals. Our current strategy, therefore, is to
ensure that procedures exist so that present quotas can be used
effectively.

As such procedures are introduced, they will be announced in the
newsletter. In the longer term, our aim is to increase logged-out
quotas substantially but balanced by archival procedures that will
have regard to the size of file as well as the access date.

4   DISK SYSTEM REORGANIZATION

Over the weekend, 27th and 28th July, the structures comprising
the disk file system will be reorganized to give effect to these
changes. We will also take this opportunity to divide the public
disk system between two structures so that we do not have all of
our eggs in the one basket. This together with a disk based
system for first level backup should substantially improve recovery
in the event of a disk failure.

We presently have two types of removable disk pack devices installed,
the RPO2 with a capacity of 40,000 blocks and the RPO3 with a
capacity of 80,000 blocks. At present, the public disk file system
(DSKC) is made up of one RPO3 and one RPO2, giving a total online
capacity approaching 120,000 blocks. After the reorganization,
the public disk file system will be made up of two structures, DSKD
and DSKE. Each of these structures will be made up of one RPO3
each, so that the total capacity will then be 160,000 blocks.
Naturally, users will be allocated to one or other of the structures
and it has been arranged that all programmers of a particular project
number (i.e. [x,*]) will be allocated to the same structure.

In the case of failure of either of the two RPO3 drives, a backup
for both DSKD and DSKE (DSKF and DSKG) respectively has been created.
Each of these backup structures is made up of two RPO2's having
then the same capacity as the structure they backup and they will
be kept in synchronism with their master by copying regularly all
files from DSKD to DSKF and DSKE to DSKG. It should thus be
possible to reconfigure the system in a minimum of time, and while
some users will not have latest versions of files, these will be
restored rapidly from the incremental dumping scheme based on BAKA.

In most cases, this rearrangement will be entirely transparent to users. The generic name DSK will still cover all files, be they on DSKD, DSKE, DSKF or DSKG (or DSKB - the system structure). When a user logs in, his structure search list will be setup to include his structure first but will also include whatever other structures are part of the public disk system. However, users who have special requirements may find it convenient to address structures explicitly or change their search list, as is discussed below. Such users must be aware of the implication of this re-organization and are likely to find it necessary to modify existing programs and command procedures.

4.1 Storage Quotas

Under the rearrangement, the following quotas will be given to all users:

	LOGGED-OUT BLOCKS	F.C.F.S. BLOCKS
DSKD or DSKE	400	2,500
DSKF or DSKG	400	2,500
DSKS	0	10,000

Note: Either DSKD/E or DSKF/G will be accessible to any user.

Further, in order to equalize the utilization of both structures, it may be necessary at any time to transfer all files of a particular project from one structure to another.

4.2 Use of the Spooling Structure DSKS

Special procedures will be necessary to make use of DSKS since the system does not allow files to be distributed across structures. Thus a user who is going to use DSKS must ensure that the file destination is correct, either by explicit reference to DSKS or by rearranging his search list. Furthermore, in the event of RPO2 drive failures, it may be necessary to remove DSKS to allow priority to other functions which require RPO2's. Hence, any user of this facility should not assume that it is available but test for its availability and take appropriate action depending upon whether DSKS is mounted or not.

4.2.1 Testing for the Availability of DSKS

This is easiest done by attempting to assign DSKS, for example -

          .ASSIGN DSKS:21

If this fails, DSKS is not mounted and the user in most cases could then assign DSK with the logical name desired.

3

4.2.2 Altering the Search List

The system will create files on the first available structure in
the user's search list, though it will search all structures in
trying to find an existing file. If a user wishes to create a
large temporary file, the following procedure could be used:

```
.R SETSRC                    ←this program manipulates the search list
*T                           ←type out present search list
DSKD:,DSKS:,DSKE:,DSKB:      (assuming user's file on DSKD)
*C DSKS:,DSKD:,DSKE:,DSKB    ←interchange DSKS & DSKD so that files
                                              are created on DSKS
*↑C
EXIT
.
```

Files that are created will now be on DSKS, but it should be noted
that permanent files will also be created on DSKS and will have to
be copied back onto DSKD before the user can logout.

4.2.3 Explicit use of DSKS

There are many ways a file can be explicitly created on DSKS, for
example the Cobol user can include a clause such as:

        SELECT PRINTFILE ASSIGN DSKS RECORDING MODE ASCII

        or

        SELECT PRINTFILE ASSIGN TMP

        combined with the command

        .ASSIGN DSKS:TMP

would achieve the same objective. Either of these could be made
to work in the absence of DSKS by the command:

        .ASSIGN DSK:DSKS

        or

        .ASSIGN DSK:TMP

In Fortran, several ways can be used. The FORTRAN-10 statement -

        OPEN(UNIT=21,DEVICE='DSKS',FILE='DATA',ACCESS='SEQOUT')

would directly create the file DATA.DAT on DSKS, or if DSKS were
not mounted, the command -

        .ASSIGN DSK:DSKS

        or

        .ASSIGN DSK:21

would create the file on the user's disk area.

The statement -

        WRITE(21,20) ---

would create a file FOR21.DAT on DSKS in the absence of any OPEN statement provided the run had been preceded by the command -

        .ASSIGN DSKS:21

A similar procedure to that outlined above would create the file on DSK if DSKS were not mounted.

Use of DSKS by other commands is much along the same lines, for example:

        .RENAME DSKS:ABC.CTL=DATA.DAT

has an obvious effect and by prefacing a run including such commands with -

        .ASSIGN DSK:DSKS

would allow the same commands to operate on files on DSK when DSKS was not mounted.

5   BMD MANUALS

A very limited stock of BMD manuals (1 Jan 73 edition) have been purchased by the Computer Centre for sale to interested users. The cost of these will be $8.50 each.

*   *   *   *   *