UNIVERSITY OF QUEENSLAND

Computer Centre

Weekly Newsletter

Date   :   Week ended 6 July 1972

Authorization :   Director of the Computer Centre

1.   OPERATIONS

1.1 PDP-10 System

Wednesday   5 July   System failure, offline 1940-1950, monitor reloaded.

Thursday   6 July   Teletype controller unit offline, 0930-0940.
                         Disk pack failure, offline 1750-1752, monitor not reloaded,
                         system failed 1756, monitor reloaded, online 1806.
                         System failure, offline 1815-1835, monitor reloaded.
                         Dispack failure, offline 2054-2100, monitor reloaded.

Schedule for forthcoming week: Maintenance 0700-0900
                                  Operations 0930-2300.

1.2 GE-225 System

Schedule for forthcoming week: Maintenance 0700-0830, 2000-2130
                                  Operations 0900-2000, 2130-2400.

2.   NEW COMMAND DECODER

A new version of the command decoder (version 2E(36)-3) will be implemented on
the PDP-10 on Wednesday 12 July. A number of important changes have been made
with this decoder. These have been fully documented in WN-95. The changes to
directory names will affect the format of any RUN command referencing library
routines.

In addition the Editor will not yet recognize all numeric file names or all
numeric processor program names.

3.   PDP-10 FORTRAN

3.1 DO Loops

The present version of the compiler does not create proper code when a function
is used as one of a DO loop's indices.

example: DO 283Ø J = 1, MINØ(6,NN)
                -
                -
                -

No diagnostic message is producing during compilation but an improper
relocatable binary file is produced and during loading, the message

> ? ILL.FORMAT BLOCKTYPE NNN PROG.name1 FILE name2/REL

will be produced. This error has been remedied in a later version of the
compiler but until this version is adopted, this construction should be avoided.

3.2 Unary Minuses

Use of the unary minuses in logical expressions or mixed logical and arithmetic
expressions do not always produce the correct results, for example,

$$J = (J1.AND."777).OR.((-(N+1)).AND."777000)$$

$$J = (J1.AND."777)+((-(N+1)).AND."777000)$$

This has been corrected in a later version of the Fortran compiler which will be
implemented in due course. Until it is available, it is suggested that this
construction should not be used and that the expression be split into two as

$$JJ= -(N+1)$$

$$J=(J1.AND"777).OR.(JJ.AND"777000)$$

4. OUTSTANDING PROBLEMS IN COBOL V3

The following note, prepared on 3 July 1972, gives a summary of known problem
areas with COBOL. The large majority of these are corrected by patches issued
by Digital and these will be incorporated as effort becomes available.

4.1 The Cobol Compiler

(a) This version does not allow qualification of condition names (to be corrected
     with version 3A).

(b) "RETURN name1 INTO name2" as part of a sort does not work (patch exists).

(c) Comparison for equality of 6-character display-6 fields may not give
     correct results (patch exists).

(d) Some external references from non-resident segments are not setup properly.
     The reported situations were

     SORT verb in resident code, RETURN and/or RELEASE in non-resident;
     TALLY in Examine generator; TODAY in the USING generator; Size error
     and display of a COMP-1 variable (patches exist).

(e) "MULTIPLY name1 BY fraction GIVING name2" where fraction is less than 1.0,
     e.g. 0.0125, will return the value 0 to name2 in some circumstances,
     depending on the typing of name1 and name2 (a patch exists).

(f) A numeric literal in a "VALUE OF DATE WRITTEN" clause fails (patch exists).

(g) The compiler allows a quoted literal for a PROGRAM-ID, but if the literal
     is less than 6 characters, the remainder of the name is random (a patch
     exists).

(h) The compiler does not successfully restart after a catastrophe dump if
     it was using a command file.

(i) The compiler will not accept lower case for the first character of a
     reserved word, although lower case characters are satisfactory in all
     other positions (a patch to allow the first character to be lower case exists).

(j) Under some circumstances, a spurious warning message "Redefinition not the same size as redefined item" may be given. This can occur when the two fields occupy an integral number of words (patch exists).

(k) Under some circumstances, cobol compiler tables are not expanded correctly. This error may manifest itself as compilation or execution error for which no simple explanation may be found. A patch exists for one situation such as this, but at least one other may exist.

(l) "PARITY IS EVEN" does not compile properly (patch exists).

(m) Improper line numbers are given for the procedure division map (patch exists).

(n) Compilation using the /P switch will produce a catastrophe dump if "TRACE ON/OFF" is used (patch exists).

(o) It appears as if the construction
     ....

     "OCCURS n1 TO n2 TIMES DEPENDING ON namel" will not create an output record of the size specified by namel but rather the maximum size n2.

4.2 The Cobol Execution package LIBOL

(a) When a random file is opened for output only, any attempt to write to it will terminate the run with an error message

     "It is illegal to change the record size when the access mode is random" (patch exists).

(b) When a Cobol program attempts to open two files which show the same buffer area, the resultant error message is garbled (patch exists).

(c) When an IO file encounters an EOF, a data location is not reset, with the result that the next sequential file input will fail (patch exists).

(d) Multi-file magnetic tapes are not positioned properly (patch exists).

(e) Some advisory messages do not conform to the specifications for Digital's 5 series monitors (patch exists).

(f) IO access to an indexed sixbit file will fail at the first read (patch exists).

(g) Code created for "USE AFTER ERROR" procedure is incorrect (patch exists).

(h) File record areas are improperly cleared under some circumstances when adjacent files are opened.

(i) Some improper checks are made when error situations arise, for example, when a file is not found. (Some patches exist).

(j) Under some circumstances the wrong quantity is used to calculate record size when a blocking factor is specified. This will cause an undeserved error message at execution (patch exists).

(k) Open immediately after close on magnetic tape fails (patch exists).

(l) Sort of an indexed file may fail (patch exists).

(m) Some problems exist with indexed files, e.g. the size of an ascii key is incorrectly computed and under some circumstances the channel used for the index file is not closed (patch exists).

(n) Under some circumstances, there may be an interaction between sort and the use of indexed files which will cause the program to fail (Patch exists).

(o) "NEXT GROUP NEXT PAGE" may cause the output of a blank page (patch exists).

(p) An EOF is found when attempting to read an indexed file sequentially (patch exists).

(q) A sort which reads a file created by the same program may get an error (patch exists).

(r) When a null record is found in a random file, subsequent inputs or outputs may fail (patch exists).

4.3 The SORT package

(a) Under some circumstances, a command file is not accepted (patch exists).

4.4 RERUN

Rerun has several problems, some of which are related to the version of the operating system currently being used and it is doubtful if satisfactory results will be achieved. A number of patches exist and when these have been incorporated consideration will be given to further testing.

4.5 ISAM

The Isam utility has some problems in handling ASCII input files and in conversion from ASCII to Sixbit files (patches exist).

4.6 GENERAL

4.6.1 Supplement to Digital's Cobol Manual

Attention is drawn to a note in Volume XII No 5 of the DECSYSTEM10 Bulletin which states that a supplement to the COBOL manual is now available. This supplement describes ISAM, COBDDT and COBOL table handling and is available on request from Digital.

4.6.2 Factors affecting the Size of a compilation

In response to a query regarding the size of the largest program that can be compiled within the present core limits, the following suggestions were offered by Digital.

(a) Identifiers longer than 6 characters in length require additional storage

(b) Condition names (level 88) should be avoided

(c) Keep paragraph names (number and size) to a minimum

(d) Do not use values on data items

(e) Keep literals as short as possible

(f) Avoid ALTERs

(g) Do not use the report writer

It is not likely that great advantage can be taken of these suggestions, but it is worth noting that Versions 3 of Cobol has proved to be able to compile substantially longer source programs than any of its predecessors.

4.6.3 Recompilation of Existing Programs

A note in the DECSYSTEM10 Bulletin on April 1st 1972 indicates the next release
of Cobol will retain the existing version of the execution time package for
those programs that presently use it. Programs compiled with later versions of
Cobol will compile a version of Libol identified by its version number. In this
way, existing compiled programs will continue to run without recompilation.
It is, however, recommended that programs previously compiled with versions
earlier than version 3 be recompiled.