

Newsletter of the SERC Central Computing Facility

No. 17 November 1981

1. INTRODUCTION TO VNET

Users of the IBM system at RAL will recollect the FORUM article 10.4 announcing that VNET installation would start shortly. The following article has been abstracted from the chapter on VNET to be published in the next release of the RAL VM/370 Reference Manual.

VNET is the name of a virtual machine running under VM on the front-end IBM at RAL. The program is an IBM supported product called Remote Spooling Communications Subsystem Networking (RSCS Networking). It has the advantage over HASP, which it will gradually replace, that it is supported and may be developed further. It should be more efficient and be able to support more than 100 workstations. The limit on HASP is 99. It will provide other functions which will be of use under VM and MVS.

VNET will provide Network Control Program support (NCP), an improved command interface and terminal support for non-networked workstations. Thus the facilities provided by extended HASP will not be lost.

Documentation is provided in the VM/370 Reference Manual, the RAL Workstation Manual and an Operators Guide. There are HELP facilities under CMS (Conversational Monitor System) and a series of EXECs will be provided to carry out most commonly required functions. Error and information messages are documented in IBM VM/370 RSCS Program Reference and Operations Manual (SH24-5005) and the RAL Workstation Manual, both of which will be distributed to workstations.

With the introduction of VNET it is now possible to control the routing of output from jobs submitted to MVT. SUBMIT and PLANT have been changed to take advantage of this but it is required to call OSROUTE before using either. It is suggested that this is included in PROFILE EXEC (the file that is automatically obeyed when logging in). PRINT and PUNCH output can also be routed to other places via VNET. The easiest way of doing this is to call VROUTE initially and then use VPRINT and VPUNCH instead. Alternatively, it is possible to set up the correct routing information directly with the CP (Control Program) commands SPOOL and TAG.

The VNET Operator commands Query, Msg, Purge, Transfer and Change are available to CMS users using the CP command SMSC.

The standard supported EXECs such as SUBMIT, PLANT and NPRINT will supply correct SPOOL and TAG commands for jobs submitted to the MVT system. The /#ROUTE record in existing MVT jobs will be upwards compatible with VNET. Additional destinations including users VM machines are allowed. Special forms are supported under VNET.

Jobs may be "punched" directly to the front-end MVT system (FEM) but they should have appropriate TAG information or a /#ROUTE record.

More VNET commands are available to the Operator at a workstation console:

- Accept - output files of particular category/device
- Backspace - restart outputting to particular device
- Change - change file attributes (eg priority)
- Flush - stop outputting a file
- FREE - start a device
- FWDspace - forwardspace pages/records within a file
- HOLD - stop output and hold the file
- Msg - send a message to a remote user
- ORDER - reorder files in the queue
- Priority - set a priority band for outputting
- PURge - remove files from queue
- Query - display status information
- Suspend - interrupt output and replace in queue
- TRANSFER - requeue files to another workstation or category

2. SOME NOTES ON UNIX

Much has appeared recently in the Computer Press on the rise in popularity of the UNIX operating system and the increasing number of machines, especially micros, which can run it (or some similar product). The Computing Division also has been involved with UNIX in a number of areas so far:

- (a) The Distributed Computing Systems Programme supports UNIX on the 11/34 and 11/70 at RAL and has supplied copies to Research Workers.
- (b) It is intended to mount UNIX on the new PERQ single user machine in collaboration with ICI.
- (c) A programme of work has begun to mount UNIX on the CEC 4000 series, on top of OS4000.
- (d) The SOFTWARE TOOLS package on the PRIMOS is implemented as a UNIX look-alike system, on top of PRIMOS.

5. COMPUTER STATISTICS
IBM SYSTEMS 7/9/81 - 4/10/81

Weekly availability is uptime/168.
MVT availability is based on 40% contributions from each 195 and 20% from the 3032.

SYSTEM AVAILABILITY

MVT - 92.5%, CMS - 96.2%, ELECTRIC - 91.8%.

MVT THROUGHPUT

Average jobs/week 13519
Average CPU hrs/week 228

TERMINAL SYSTEM USERS

Registered users 610 1267
Active users 254 598

SERVICE LEVELS

Percentage of prime shift short jobs not turned round inside guideline:

MVT Batch Core size F12 P10 P8

0 - 210k 1.9 6.8 2.7
212k - 350k - 10.8 3.1
352k - 560k - 2.4 3.1

TERMINAL SYSTEMS

Response to trivial command during peak period:

CMS	Week 1	Week 2	Week 3	Week 4
% < 1 sec	99.9	99.8	99.7	93.0
% < 3 secs	100.0	99.9	100.0	99.0
ELECTRIC				
% < 2 secs	75.0	77.3	81.7	82.3
% < 5 secs	88.7	90.2	94.2	93.8

USAGE FOR CURRENT FINANCIAL YEAR - 27 WEEKS TO DATE

Board	MVT 195hrs	ELECTRIC AUS	CMS AUS
ASR	366	454	62
Engineering	470	234	17
Nuclear Physics	3905	3592	1110
Science	847	1227	506
Central Funding	119	590	3971 *
NERC	93	279	238
External	93	406	42
TOTAL	5893	6782	5946

ICF SYSTEMS

AU USAGE BY BOARD - periods 8104-8110

Board	Prime	GEC	DEC-10	TOTAL
ASR	105	197	22	324
Engineering	7211	2969	3480	13660
Nuclear Physics	0	1	0	1
Science	130	88	1206	1424
Central Funding	5064	1015	3074	9153 *
External	225	96	140	461
TOTAL	12735	4366	7922	25023

* These entries include some usage due to "service" functions which are strictly an overhead and should be accounted separately.

6. DIARY OF FUTURE MEETINGS

USER MEETINGS

GEC User Group - 25 November 1981 (provisional)
at Atlas Centre

IBM Representatives - January 1982
in RAL Lecture Theatre

Prime User Group - March 1982
at UMISR

ERCC DEC-10 Users Committee - 26 November 1981
at Edinburgh Regional
Computer Centre

7. CORRECTION TO ITEM 5 IN FORUM 16

Under the subheading Systems several incorrect statements were made. The corrections are given as follows:

- (a) MVT will not be supported from Rev. 19 of PRIMOS.
- (b) SOFTWARE TOOLS has been purchased and will be distributed to any site requesting it. It runs under PRIMOS and provides a UNIX-like environment including a full screen editor which supports a wider range of VDUs with cursor addressing facilities.
- (c) Batch jobs running on PRIME A and PRIME B at RAL and on UMIST PRIME will be charged at 0.8 of the interactive charge rates in force at the time of running.

(e) The SOFTWARE TOOLS package is available on STARLINK, running directly under VMS.

(f) The AMDAHL version of UNIX, UTS, has been tried experimentally on the 3032 under VM.

So what is so special about the system and what does it give the user?

The major article describing UNIX is to be found in the Communications of the ACM, Volume 17, Number 7 (July 1974), written by the original authors Dennis Ritchie and Ken Thompson of Bell Laboratories. Some of the main features are:

(a) Size
For a 'modern' operating system it is fairly small and is therefore mountable on many machines.

(b) Language
It is written in the language C, which is a high-level assembler, although looking a bit like a cross between BCPL and PASCAL. The GEC Assembler, BABBAGE, looks similar. This language is available to the user and opens the door to a lot of existing software (except in the case of the PRIME where the look-alike version is implemented in FORTRAN).

(c) Filestore
A hierarchical filestore is available, where users may have many directories. There are no special file types so the same structure is used to store text, binary programs and even directories. Individual files may have protected access.

(d) Devices
One novel aspect is the way that output to a device is handled. As far as the user is concerned this is done in exactly the same way as a file. Reference is made to a special filename (such as LP, PPT) in a special directory (/DEV) and the output goes to the correct device.

(e) Command Environment
Access to UNIX gives entry to a Command Processor called the Shell. This is similar to the type of facility available in OS4000 on the GEC 4000 series but is more flexible. Commands are files which contain binary programs or processes (eg CMS MODULES) or macros (eg CMS EXECs). Each user may modify a copy of the Shell to provide a tailored command system for his or her own use. However, a major feature is the ability to start more than one process at once and to make these processes run conditionally on each other. The Shell may be called explicitly to run a command, which will cause a copy of the Shell to execute that command while the user gets on with something else. The filter system (see next section) makes use of this by starting all the commands together and organising things so that one process can proceed as soon as some output is made available from the previous one (rather than having to wait for the first command to finish before even starting the next).

(f) Standard I/O
Each command has a standard input and a

standard output file available to it, which will normally be connected to the keyboard. It is possible to string together commands on a line, so that the output of one automatically becomes the input of another without the need to mention any temporary files explicitly. A command that reads from standard input and puts processed data on the standard output is called a filter. By having a set of filters which perform single operations (such as sort, layout, etc) it is possible to build complex operations on any user program with hardly any interface worries. A communications link between two processes is called a pipe.

This short article can only give a brief review of UNIX but it is intended to show why there is so much interest in it.

3. EDIT TO XEDIT MIGRATION

This article has been written for FORUM by John Simkin of Technology Division, RAL. He has stated his qualifications for writing it as "I objected to the move from EDIT to XEDIT!"

I have now been fully won over to XEDIT even though the command names are illogical, it has too many commands and some features of EDIT are not reproducible. It can be very powerful and learning a few simple techniques will allow very complex edit functions to be specified.

The following notes and suggestions are particularly important:

(1) Rename the XEDIT command to EDIT. Start the way you intend to continue. This can be done in your SYNONYM file.

(2) Have your own PROFILE XEDIT file. Every time you issue the XEDIT command this file is used to set up your profile for the editor. You should start from the default file on the R disk and to this base add your own defaults and new commands. Here are some I have used:

```
SET ARBCHAR ON $ - $ acts as an arbitrary length character string.
SET STAY OFF OR ON - to define where the line pointer should end up if a locate fails.
SET LINEND ON \ - define local line end.
SET TABS C1 C2 C3 - define tab positions, or find the file type and define the tabs in your PROFILE XEDIT.
TRANSFER FTYPE - get file type
&READ &VARS &FTYPE - sets &FTYPE=file type
&IF &FTYPE EQ FORTLAN etc
SET SYNONYMS - define commands
SET SYN EDIT 2 XEDIT - gets rid of XEDIT again
```

Define new commands:

```
SET SYN LINE 2 QUERY LINE - gives current line (Line)
```

(3) ARBCHAR can be very convenient. To locate the first occurrence of AAA followed by BBB with any intervening text:

```
L/AAA$BBB/
Similarly it can be used with the change command. To change all PRINT n, etc to WRITE C/PRINT$/WRITE(6,$)/# *
```

(4) Logical constructions can be added to locate targets (see System Product Editor Command and Macro Reference, page 3-63). This can be useful, but macros are generally more applicable. Multiple targets can be specified before a command.

For example:

```
:H3/FRED/C/FRED/JIM
Locates the first occurrence of FRED after line 43 and changes it to JIM.
```

(5) To repeat the last command:

```
= - repeats the last command
REPEAT NTIMES - this advances the line pointer before each repeat!
```

(6) Write your own temporary macros for global edits. The following file could be used to find all occurrences of COMMON/A/, then the preceding subroutine statement, and insert an integer declaration.

```
-START
L:COMMON/A/ - locate next COMMON/A/
&IF &RTCODE NE 0 &EXIT - exit at end of file
L-/SUBROUTINE/ - find preceding subroutine (L- search backwards)
&BEGSTACK 1
INTEGER AB - put input line in console stack
&STACK - put null line in console stack
INPUT - add input from stack
L:COMMON/A/ - go back to COMMON/A/
&GOTO -START
```

Warning: this macro will fail unless SET WRAP OFF is in effect as otherwise it will wrap around instead of stopping at the end of file.

Note that the use of &BEGSTACK prevents removal of surplus blanks between "words" from the stacked lines as happens with &STACK.

There are many inelegant features of XEDIT but together with EXEC 2 it gives a powerful system. Access to manuals is essential for both and armed with these it is possible to find many other unexpected features. IBM give a chart showing command equivalences in Appendix C of the System Product Editor Command and Macro Reference.

4. FILE PROTECTION

The default file access on the SERC PRIMES permits any user to access files in any other user's directory. Two mechanisms can be used to restrict access to files by other users. Users who wish to protect their files are recommended to use them.

Passwords

Passwords permit protection at user file directory (UFD) level and sub UFD level. They can be specified for both owner and non-owner access. The PASSWD command is used to set passwords.

```
PASSWD
To clear the passwords to blanks.
```

```
PASSWD owner
To set the owner password to 'owner' and clear the non-owner password.
```

```
PASSWD owner non-owner
To set the owner password to 'owner' and the non-owner password to 'non-owner'.
```

With no password set any user can gain access to all files. With the owner password set other users need to know the password in order to modify or create files. They can still read (copy) and execute files. With the owner and non-owner passwords set other users cannot access files unless they know either the owner or non-owner password.

File Access Protection Codes

Individual files can be protected at owner and non-owner level. Even if the non-owner password is known, access can still be barred for specific files. The protection code takes 2 parameters:

```
(1) Owner protection
(2) Non-owner protection
```

The PROTECT command is used to specify the protection level for a particular file. PRIME Manual Version 4, issued by RAL Computing Division, gives full details of the protection codes on page 134.

This article was reproduced from UMIST Computer News, No. 195.