csiro dms Newsletter 70
November 1980

The UNIX timesharing system in DMS

1. Background
The major part of DMS computing is done on the computers of CSIRONET, but these facilities are supplemented by mini-computers at major DMS centres. The first of these were purchased in 1974 and were either PDP11/10s running the CAPS-11 operating system and programmed in Focal, or were Hewlett Packard 9830 desk-top computers programmed in Basic. By 1976 some of the PDP11s had disc drives that permitted the use of the RT-11 operating system, and the major programming language became Fortran, although Focal and Basic were also available.

These systems provided a highly interactive environment for tasks where the computational load was not too heavy, and became increasingly popular for some applications. Consequently, by 1979 the single-user systems were becoming quite inadequate, particularly in larger centres. Two of the PDP11/10s had now been replaced by PDP11/34s, and it was only necessary to install extra memory to enable multi-user systems to be operated on these machines.

Two systems were considered. RSX-11M is a product of Digital Equipment Corporation and has recently been described in an article entitled "DEC operating systems for the DCR research PDP11/40" by Donald Fraser in the *Division of Computing Research Newsletter* No. 148 (Aug. 1979). The other contender was UNIX, a product of Bell Laboratories that operates on the PDP 11 range as well as some other computers. Eventually UNIX was chosen and is now operating at three sites.

2. The UNIX system
The UNIX system is similar to other time sharing systems in that each user at a terminal is able to make use of all the system facilities regardless of what the other users are doing, so in a sense the users appear to have the complete machine to themselves. The central part of the system maintains a file system, handles input and output from executing processes, allocates memory to processes, shares the CPU time between processes, and swaps process-images out to disc when necessary. The user does not need to know details of how all this is done.

Features of UNIX that tend to distinguish it from other systems include:

(a) The file system is organised as a hierarchical tree with no limit on the number of levels. This allows related files to be grouped in one directory, related directories to be grouped in another, and so on.

(b) The file system is relatively good at maintaining a large number of small files and performance does not deteriorate when the device is nearly full. This encourages users to partition programs and data into small parts. It is also worth noting that the command structure makes it easy to use the concatenation of a series of small files wherever a large file is expected.

(c) The command structure allows small program modules to be plugged together using data paths called *pipes*, which encourages the development of modular building blocks rather than larger monolithic programs.

(d) The name of a command is the name of a file containing the program to be executed. Several directories (both user and system) are searched for the file, so that it is easy for individual users or system managers to redefine the effect of certain commands for some or all users. Thus testing of revised commands is easy during normal system operation.

(e) The complete manual is kept on line so that this copy is the definitive copy of the documentation, and printed copies are simply a snapshot of it at some point in time.

(f) The system is almost entirely written in a well structured high-level language (C) and is consequently easy for the system manager to understand and modify.

(g) The system is highly reliable and almost never crashes as the result of user actions.

3. UNIX configurations in DMS
UNIX was implemented in Sydney in March, Melbourne in April, and Canberra in July. The version implemented is a much enhanced level 6 that has been developed at the Universities of New South Wales and Sydney. It is referred to as Australian UNIX Share Access Method (AUSAM) and provides various additional controls such as disc quotas and print limits.

The peripherals on the three systems at this stage differ, but the eventual configurations of each system will probably include the following facilities:

(a) VDU terminals connected at high speeds (usually 9600 baud) with the standard ASCII character set.
(b) graphics display terminals connected at high speed (at present Tektronix 4010 terminals are used).
(c) graphics hard copy devices which may be either the hard-copy unit for a Tektronix terminal or a plotter.
(d) connection to CSIRONET providing access to the system for one UNIX user at a time (at this stage).
(e) dial-in modes allowing access to the system from remote sites.
(f) line printer capability provided by 100 char/sec matrix dot printers which seem to be adequate at present.
(g) quality printer for output of text prepared using word processing software.

The source code of our UNIX systems is maintained in Sydney with specific code for local variations being conditionally included during compilation where appropriate. Thus one master copy of the source code can generate the several required configurations.

In a similar way the source code for application packages can be maintained from one site and distributed to the others. However, the total effort of software support can be distributed amongst sites.

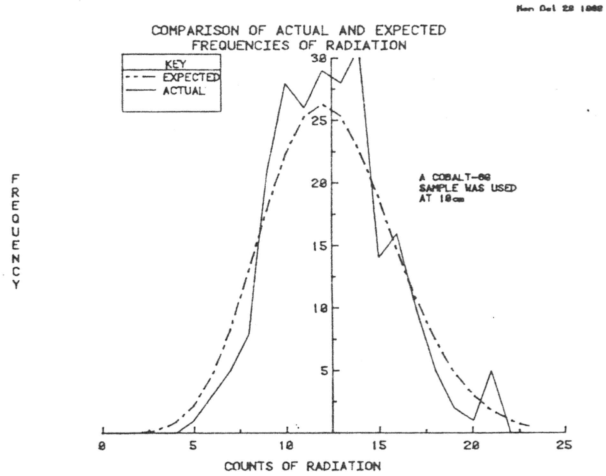
4. Application software used with DMS UNIX

The following list is not a complete list of software but gives some idea of the more frequently used items.

(a) *Compilers.* The Fortran, Basic and Macro compilers are DEC PDP11 systems adapted to UNIX. The C compiler is an integral part of UNIX, and the Pascal comes from Berkeley. The *bc* arbitary precision desk calculator language is a standard application package with UNIX systems.
(b) *Word processing software.* Most of the software for text formatting (*nroff, troff, eqn, tbl,* etc.) comes from Bell Laboratories and is quite comprehensive. The scope of this software is outlined in a separate article in this *Newsletter.*
(c) *Glim.* The RT-11 version of the Fortran source was easily compiled under UNIX but the definition of the overlay structure required some attention. The workspace has been extended to 3000 double precision words so this is a big improvement. Although it would be convenient to have more workspace, problems that require more than 3000 are more suitable for the larger computers of CSIRONET.
(d) *Net.* Our CSIRONET communication has been developed from a program written in the C language by Doug Ryan of the Division of Computing Research. He used the well known RT-11 *net* program as a model. The program can be used interactively, or else a prepared script of commands can be fed into it. In this way automatic *send* and *fetch* commands have been implemented for file transfer.
(e) *Ted.* The CSIRONET editor as implemented for minicomputers is called *ted* and has been adapted to UNIX. It is written in Fortran and is less efficient than the UNIX *ed*, particularly for extensive searching through large files. Nevertheless, it is useful for operations such as "remove characters from positions *m* through *n* on every line" which are easy in *ted* and not so obvious in the UNIX *ed*, and assists occasional users who are already familiar with CSIRONET.
(f) *Gap.* Graphics Assistance Package written by Ross Nealon at the University of Wollongong. This consists of a set of Fortran subroutines with both low and high level cells that offer very extensive facilities. The Fortran programs generate device independent output which is then processed for a particular device by an interpreter written in C. This software is well written so that addition of extra interpreters for new plotting devices will be straightforward. However, it is fairly slow to execute on machines without floating point hardware.
(g) *Plot.* This graphic package is quite different and very easy to use, although not quite as flexible as *gap*. It has a simple command language to define the plots and normally operates on data from one or more files. (In some ways it is analagous to using *Glim* for data analysis instead of writing Fortran programs.)
An example of input to *plot* is:

```
! Example 4
legend key at 2
    "COMPARISON OF ACTUAL AND EXPECTED"
    "FREQUENCIES OF RADIATION"
message at 7 5
    "A COLBALT-60"
    "SAMPLE WAS USED"
    "AT 10cm"
xaxis
    "COUNTS OF RADIATION"
yaxis
    "FREQUENCY"
at 12.4978
plot x=1 y=2 "ACTUAL"
line = solid
data "actual5"
plot x=1 y=2 "EXPECTED"
line = short long
data "expected5"
```

This causes data to be read from files *actual5* and *expected5* and then produces the following plot.



This software has been written by Richard Grevis at the University of New South Wales.

(h) *Plot-10.* These subroutines for plotting on the Tektronix allow Fortran programs to easily use the Tektronix terminals. A number of convenient plotting programs have been written that use this library.
(i) *Other software.* Fortran subroutines of the SSP, Linpack and other libraries are available.

5. Documentation

The UNIX documentation is terse but systematically organised so that once a little familiarity with the manual is achieved, information is easy to find. The other major advantage with UNIX documentation is that it is on-line. This means that while the printed versions may become dated, the definitive manual for our system should be the on-line copy.

The system as received had documentation that did not correspond closely to the software included – there were entries for software not available, and no entries for some software that existed – but this is now largely remedied.

The "DMS UNIX User's Guide", a printed copy of a subset of documentation and tutorial guides selected for our environment has already been produced. The next project will be to produce a small number of printed copies of the complete manual for our systems and then some time later a second edition of the User's Guide will be needed.

6. Conclusion
While RT-11 is a quite effective operating system, the move to UNIX represents a major step forward for DMS, not only because it is a multi-user system, but also because it provides a better programming environment with more powerful facilities for software development, and further extends our range of application software particularly in areas such as document preparation and graphics.

Another advance is that our geographically scattered systems are being maintained in a more closely coordinated way than has been the case with earlier systems. This coordinated maintenance of the system and application software permits local modifications to be easily made where needed, yet avoids the wastage of man-power that would be involved in maintaining separate systems. Further economy of man-power will be achieved by continuing a policy of carefully coordinated equipment purchases. The system is still young, the pace of development has been rapid, and further efforts with software development and documentation are needed. However, as the productivity of the users becomes more apparent we would not be surprised if other minicomputer users in CSIRO become interested in UNIX.

CSIRO, DMS,
PO Box 218
Lindfield, NSW 2118

Ron Baxter

Staff and visitors news

Staff News
Dr N. G. Barton, presently with the School of Mathematics, University of NSW, has been appointed Senior Research Scientist with the DMS Applied Mathematics group. He will take up his position in mid 1981.

David Cook, DMS Administrative Officer, transferred to CSIRO Minerals Research Laboratories at North Ryde, NSW, on 20 October.

Mr B. P. Durnota has been appointed Experimental Officer in Melbourne for 12 months from 1 December, 1980. Mr Durnota has completed an Honours degree in Pure Mathematics at La Trobe University, and is just finishing an Honours degree in Computing Science. He will be working mainly on renewable energy modelling with Ian Saunders and Richard Tweedie.

Visitors
DMS Canberra has heard seminars from several visitors in late October:

- **Professor F. Chatelin**, IMAG, Grenoble, France;
- **Dr G. N. Lance**, Avon Universities Computer Centre, Bristol, UK;
- **Professor O. Axelsson**, Mathematical Institute, Catholic University, Nijmegen, The Netherlands;
- **Professor G. W. Gear**, Computer Centre, University of Illinois, Urbana, USA.

CSIRO, DMS,
PO Box 1965,
Canberra City, ACT 2601

Barbara Hartley

Genstat 4.03

A test version of the latest release of Genstat has been loaded on the CSIRO Cyber 76, and by the time this newsletter is published Genstat 4.03 should be the default version. For the time being version 4.02 will continue to be available as cycle 12. Anyone who requires a manual should order it from Numerical Algorithms Group Ltd, NAG Central Office, 7 Banbury Road, Oxford OX2 6NN.

Please send details of problems with the new version to Ron Baxter, PO Box 52, North Ryde, NSW 2113.

CSIRO, DMS,
PO Box 1965,
Canberra City, ACT 2601

Jeff Wood

Word processing in UNIX

Nroff and *troff* are programs used to format text, according to commands embedded in the text, into a form suitable for output to a typewriter-type terminal (*nroff*) or a photo typesetter (*troff*). They are used in conjunction with pre-processors *tbl*, which provides an easy way to set up tables, and *neqn* (or *eqn* with *troff*) for mathematical expressions. The detailed commands generated by these programs are passed to *nroff*. *Tbl* and *neqn* will be discussed in a future *Newsletter* article.

Commands are usually basic requests i.e. an instruction understood by *nroff*, or a call to a macro. A macro is a set of commands which may be basic requests or calls to other macros; it is analogous to a function in Fortran or to a macro in Genstat. Basic requests recognised by *nroff* include all aspects of page control, text filling and adjusting, vertical spacing, line length, margins and hyphenation.

A call to a basic request begins with a control character (usually a period) followed by a one or two character name (lower case letters). There may be an argument. A call to a macro is similar, but the name is usually upper case letters; there can be up to nine arguments. For example, .sp 4 is a basic request with an argument; it inserts four blank lines.

Preparation of documents is facilitiated by using the *-ms* macro package with *nroff*. This was developed at Bell Laboratories to produce manuscripts, memoranda, reports, etc. It includes instructions to structure titles, paragraphs, abstracts, footnotes, references and so on. The macro .TL (for formatting a title) will take a new page, centre the text and insert appropriate line spaces. .LP specifies a left-aligned paragraph (no indenting of the first line).

User defined macros can also be used with *nroff*. Any combination of two upper case letters may be used as a name, except those used by *-ms* for macro names or for internal register names. A user defined macro to print keywords might look like

```
.de KY
.sp
.PP
Key words:
..
```

and be in a file called *ownms*.

User defined macros may consist of calls to *-ms* macros, basic requests and text. The text in the data file is inserted after all the commands.

Simple example:

```
.LP (left paragraph)
.TL (title)
MARGINAL SKEWNESS AND KURTOSIS IN TESTING
MULTIVARIATE NORMALITY
.AU (author)
N.J.H. Small
.AI (author's institution)
Imperial College, London
.sp 2 (space down 2 lines)
.ce (centre)
[Received November 1977. Final Revision April 1979]
.AB (abstract)
A method is suggested by which marginal third- or fourth-order moments
may be combined to produce a statistic to test for multivariate
normality.
.AE (abstract end)
.KY (user macro defined above)
Skewness; kurtosis; multivariate normality.
.IP 1. (indented paragraph to be numbered 1)
INTRODUCTION
.LP (left paragraph)
A necessary, though not sufficient, condition for a multivariate
distribution to be normal is that all the marginal distributions
should be univariate normal. .....
.IP 2. (indented paragraph to be numbered 2)
THE METHOD
.PP (paragraph)
An improved method of testing is obtained from a combination of
the marginal coefficients of skewness or kurtosis by use of the
appropriate quadratic form. .....
```

Output:

```
           MARGINAL SKEWNESS AND KURTOSIS IN TESTING
                   MULTIVARIATE NORMALITY

                        N.J.H. Small

                  Imperial College, London

   [Received November 1977.  Final Revision April 1979]

                          ABSTRACT

     A method is suggested by which marginal third-  or
     fourth-order  moments may be combined to produce a
     statistic to test for multivariate normality.

     Key words: Skewness; kurtosis; multivariate normality.

1.   INTRODUCTION

A necessary, though not sufficient,  condition  for  a  mul-
tivariate distribution to be normal is that all the marginal
distributions should be univariate normal.    .....

2.   THE METHOD

     An improved method of testing is obtained from a combi-
nation  of the marginal coefficients of skewness or kurtosis
by use of the appropriate quadratic form.    .....
```

To use *nroff*, assuming the document has been prepared and stored on *file*, the following commands are used. The vertical line indicates that the output from one program is "piped" to the next program as input. See section 2(c) of Ron Baxter's article on UNIX in this *Newsletter*.

Documents with just text	*nroff -ms file*
Text and user macro file	*nroff -ms ownms file*
Equations and text	*neqn file* | *nroff -ms*
Tables and text	*tbl file* | *nroff -ms*
Equations, tables and text	*tbl file* | *neqn* | *nroff -ms*

CSIRO, DMS,
PO Box 218,
Lindfield, NSW 2070

Elaine Smith