

Weekly availability is uptime/168.

SYSTEM AVAILABILITY - % of 672 hrs available

MVT - 69.5%, CMS - 74.8%, ELECTRIC - 47.6%.

MVT THROUGHPUT

Average jobs/week 8221  
Average CPU hrs/week 131

TERMINAL SYSTEM USERS

CMS ELECTRIC  
Registered users 899 1215  
Active users 466 419

SERVICE LEVELS

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

MVT Batch	Core size	P12	P10	P8
0 - 210K		7.6	22.4	5.5
212K - 350K		-	10.4	8.3
352K - 560K		-	3.1	2.1

TERMINAL SYSTEMS

Tabular presentation is replaced by the graphical presentation shown on the previous page.

USAGE

Cumulative totals are for current financial year - 30 weeks to date.

Board	MVT 195hrs	ELECTRIC AUS	CMS AUS
ASR	240	242	330
Engineering	431	176	357
Nuclear Physics	3951	1462	1748
Science	524	408	577
Central Funding	141	284	4223 *
MERC	85	57	196
External	70	71	172
Overheads	-	-	458
TOTAL	5442	2700	8061

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

ICF SYSTEMS

AU USAGE BY BOARD - periods 8204-8211

Board	Prime	GEC	DEC-10	TOTAL
ASR	136	162	36	334
Engineering	7735	2619	5031	15385
Nuclear Physics	123	131	0	255
Science	425	455	680	1561
Central Funding	4974	1067	934	6976
System Overheads	9266	233	1603	11103
External	303	186	121	611
TOTAL	22962	4853	8405	36225

### 11. DIARY

#### IBM PREVENTATIVE MAINTENANCE DATE

Routine Preventative Maintenance will take place on 16 December from 1800 - 2200 hours. Login messages on the ELECTRIC and CMS services will be issued prior to the maintenance session.

#### FORTHCOMING EVENT - SERC Computing Summer School

An intensive two-week residential Summer School will be held at The Cosener's House, Abingdon, from 11 - 22 April 1983, on the theme "Good Practices in the Production and Testing of Software". It is restricted to SERC staff, SO/HSO, who have a science degree or equivalent and who spend more than half their time computing. Application forms will be available through the Establishment later this year, but if you are interested please contact R E Thomas, Computing Division, Rutherford Appleton Laboratory and you will receive a form direct. Places are limited to 20.

#### CHRISTMAS ARRANGEMENTS

The IBM system will be shutdown from 12.00 hours on Friday 24 December until 08.00 hours on Wednesday 27 December. There will be no New Year interruption to the service.

### 12. IBM DOCUMENTATION FOR CMS USERS

As stated in CIGAR part A, individual users registered to use CMS are provided with an RAL VM/370 reference manual, reference card and 3 IBM reference cards:

SX20-4401, VM/SP Command Ref Card for General Users  
SX24-5122, VM/SP Editor Command Language Ref Card  
SX24-5124, EXEC2 Language Ref Card

Group Representatives should hold a reference set of more detailed IBM documentation covering VM/SP CMS as follows:

- SC19-6210, VM/SP CMS Users Guide
- SC19-6211, VM/SP CP Command Ref for General Users
- SC19-6209, VM/SP CMS Command and Macro Ref
- SC24-5221, VM/SP SP Ed Command and Macro Ref
- SC24-5219, VM/SP EXEC2 Reference

The documentation officer is attempting to identify all group representatives who have not yet received their copies of the above manuals. She would be pleased to hear from any group representative. The cost of a complete set of the 5 manuals is £74. It is not economical to distribute the manuals more widely.

Release 2 IBM and RAL VM/SP manuals will be issued to individual users and group representatives as soon as the software is in production. Tentative timescales are as follows:

VM/SP CMS Rel 2.0 January 1983  
VM/SP CP Rel 2.0 July 1983

J Brown - User Interface Group

# Rutherford Appleton Laboratory

# FORUM

## COMPUTER NEWSLETTER

### Newsletter of the SERC Central Computing Facility

No. 29 November 1982

#### 1. 360/195 WAKE IN THE ATLAS CENTRE, RAL



Bill Walkinshaw with his finger on the button, watched by (l to r): Harry Hurst, David Livermore and Geoff Manning.

On Thursday 21 October 1982 the original 195 purchased in 1971 was laid to rest.

About one hundred guests and staff who had been closely associated with the 195 project from the beginning were invited to the closing ceremony by the Director of the Laboratory and IBM. Speeches were made by The Director Dr G Manning, Mr David Livermore the IBM Sales Director and Mr W Walkinshaw, extolling the virtues of the people and the machine. Those who got a special mention were John Burren, Bob Taylor and Harry Hurst. After the speeches the assembled company moved into the computer room for the final act. Bob Taylor then ran the last job which was one of the first run on the machine in 1971. Finally, with cameras flashing, Bill Walkinshaw powered off the machine.

It was a nostalgic occasion. Many friends and colleagues laid to rest a good servant after eleven years hard labour, as the statistics show. The evening ended in true wake style with a wine and buffet supper. It was a fitting end to such a fine computing engine. R.I.P.

#### STATISTICS

Total jobs run 6,738,168 jobs  
Total CPU time 79,945.89 hours  
Total cost of jobs 24,569,383 pounds  
(at prevailing Government rate)

#### BREAKDOWN BY SERC BOARD

Board	CPU Usage (hours)	Jobs (number)
ASR	3871.25 (4.8%)	540,908 (8.0%)
ENG	3935.79 (4.9%)	215,441 (3.2%)
NP	55004.09 (68.8%)	3,398,371 (50.5%)
Science	12114.35 (15.2%)	776,452 (11.5%)
Others	5020.41 (6.3%)	1,806,996 (26.8%)

D G House - Head of Operations Group

### 2. DBOS - DATABASE MANAGEMENT SYSTEM

DBOS is a library of FORTRAN subroutines providing the application programmer with the ability to construct a disc orientated database management system within his program. It was developed by Applied Research of Cambridge (ARC) from their computer-aided architectural design software OXSYS-BOS following a recommendation of the IOF Pre- and Post-Processors Working Party (now SIGTIP, Special Interest Group on Tools for Interactive Programs). DBOS is presently mounted on a GEC4070. Data maintained by DBOS may be complex and variable in structure. A file security system protects databases against system malfunction and ensures that inter-linked files are kept in step. Databases may be large and long-lived. DBOS is designed to keep them compact while at the same time optimising data transfer between program and disc storage. It should be stressed that this is not a simple, high-level, user-friendly package for quick results. However, the power and flexibility of the data handling facilities would repay the time necessary to require a knowledge of the system.

For further details, advice and a copy of the manual contact Brian Read (ext 6492) or Ken Smith (ext 5667) in the Computing Division Database Section.

B J Read - Applications Group

### 3. AIR-CONDITIONING SHUTDOWN, 22-27 October 1982

The return of full central computing services was seriously delayed following the shutdown. A detailed explanation is necessary so that you can get the feel for the magnitude of the task that was undertaken during the week, 22-27 October.

First let me apologise for the inconvenience caused. We are acutely aware of the importance of the Computing Service to your work.

A great deal of detailed planning went into this project but like most major work some unforeseen problems arose throwing a spanner into the works.

The major cause for the delay was failure to get electrical power when scheduled. In fact it was one and a half days late. This was due to a variety of reasons, the main one being that wiring was inadvertently stripped out and therefore had to be replaced.

On Friday 22 October the first job was to remove 195/1, 195/2 having already been removed. It took two days to extract two lorry loads of surplus cabling from under the false floor. The second job was to resite all the peripherals in R6 and this involved moving all disc, magnetic tape and unit record equipment with their controllers.

Major structural changes were made to both office accommodation and air-conditioning trunking. This was in addition to the normal air-conditioning maintenance programme. Throughout the week the staff, particularly Paul Thompson, K Benn, A Cox, D Rigby and all the manufacturers' engineers, worked extremely long hours from 08.30 - 1.30 the following day.

You can see from this that effort was not lacking. We have learned some valuable lessons from this project. The one bright spot is that the installation of the Atlas 10 next year should have minimal impact on the service.

D G House - Head of Operations Group

### 4. REPORT ON THE SYMPOSIUM ON 3D-COMPUTER GRAPHICS RUTHERFORD APPLETON LABORATORY - 29 JUNE 1982

This is the summary of the remainder of the talks given at the Symposium, the first of which was published in FORUM 28.

Mr Hornung from the Technische Hochschule Darmstadt gave a talk on his current research into a generalised visibility concept.

Many visibility algorithms have been developed, some of which are appropriate to a limited range of applications or devices. The underlying data structures used consist typically of a set of plane polygons. Simpler approaches to the visibility problem compute the visibility of all primitives explicitly. This is in general wasteful. In Mr Hornung's approach, the surface primitives of the data structures are first collected into regions which are then tested against each other. This leads to the determination of regions which are either completely visible or completely invisible and hence do not need further

consideration. The basic algorithm is described in Mr Hornung's paper in Proc. EUROGRAPHICS '81 ed J L Encarnacao (North Holland Publishing Company 1982).

Mr Hornung's recent work has been concerned with hardware support for the algorithm through an attached processor. The attached processor is a hierarchically structured multi-microprocessor system with the different parts of the algorithm occupying different levels of hierarchy.

Timing results were presented which were very encouraging.

After lunch, Prof Forrest (University of East Anglia) gave an impressive illustrated lecture of work by his group on raster graphics in engineering design. The work is based on two premises:-

- a) Memory is becoming progressively cheaper.
- b) VLSI will enable currently expensive operations to be performed cheaply.

The slides shown depicted various applications of anti-aliasing technique to lines, text and area shading. Increasingly good results are obtained at the expense of algorithm complexity and computational cost.

Dr Woolfson of the University of Sussex gave a talk on a flight simulator system being developed at Sussex. Surface texture is very important in flight systems. 'Conventional' systems use TV cameras and models to display the terrain visible from the cockpit, the challenge is to replace such technology with a computer system. The problem is to display images in 'real-time', ie 25 frames per second.

Sussex have designed a pipelined system for this task, each stage of which can be a multi-processor. Typical stages are world model store, transformation and perspective processor (ZMP). The ZMP is a special purpose multi-processor which will display a set of polygons in real time.

A basic version of the system has been working for some time. A massive hardware construction exercise is now in progress to build a prototype which is large enough to handle realistic scenes. A film of a simulated landing using a simplified terrain model was shown which admirably illustrated the technique.

Dr Bettels from CERN then gave a talk on applications of 3D Graphics to data analysis in high energy physics. He described data analysis techniques for bubble chamber, counter and colliding beam experiments. This work was illustrated with a film.

Finally Mr Hewitt from the University of Manchester Computer Graphics Unit spoke on 3D graphics work at Manchester. The facilities at the Graphics Unit include Tektronix displays, a vector general display and genisio GC13000 colour raster display. The software system is based on GINO-M, a library of FORTRAN callable subroutines written in Macro-11.

After a fascinating illustrated discussion of perception and illusion, Mr Hewitt described routines for polygon fill and hatch and for displaying colour pictures of spheres (eg molecules). A description of the work of the unit

appears in Computer Graphics Forum, Vol 1, No 2.

The UK Chapter of Eurographics intend to organise similar one day symposia in the future. For advance notice of such activities please write to:

Dr D A Duce  
Computing Division  
Rutherford Appleton Laboratory  
Chilton, DIDCOT  
Oxon OX11 0QX

### 5. CMS PERFORMANCE IMPROVEMENTS - IDEAS FROM SHARE

The items in this article have been extracted from a report written by I G Pett of RAL after his visit to the SHARE59 conference. General users of CMS will find them useful.

- (1) Long Ready Message.

This uses diagnose OC. It also requires extra CONWAIT which causes extra queue drop.

- (2) ASCII terminals.

Switch off extra line feed which wastes space on a screen. We do this with NOLF option on SET RDWMSG.

- (3) GENMOD

OS option is the default. ALL option avoids OS simulation and saves the loading of TXLIB directories. ALL should be used where possible.

- (4) File Lookup

For a disk in R/W mode, the filing system does a linear search. For a disk in R/O mode, the filing system does a binary search and saves CPU. However, it does a linear search for a R/O disk if there is an auxiliary directory, which is not very frequent.

- (5) Abbreviations

Command handler does synonym check twice if it is a module, once for EXEC and once for MODULE. Therefore do not use abbreviations inside EXEC files.

- (6) Disk Organisation

Use disks with 2K blocksize for large files, eg the S-disk. Keep the number of files small as this minimises the cost of disk I/O.

- (7) Blocking Factors

Efficient blocksize save CPU, SIOs and elapsed time. The optimum blocksize for a file is about 2 x disk blocksize. The record length of a file should divide integrally into the block size.

I G Pett - Systems Group

### 6. CERN LIBRARY

#### New Version of MINUIT Package

The MINUIT package is now available on WVT and CMS as part of the standard CERN library. The double precision short and long versions have been installed but the short version is only available on WVT. In addition to the subroutine FCN, users must supply a trivial main program which calls either MINTSD (for the short version) or MINTLD (for the long version). The MINUIT package can then be linked in using the autocall library mechanism.

On WVT, the following JCL is needed:

```
.....  
// EXEC FCNLIB,REGION=G=190K  
//C.SYSIN DD *  
CALL MINTLD  
STOP
```

```
END  
SUBROUTINE FCN  
.....  
//G.SYSIN DD *  
- User data -  
/*  
/*
```

The short version needs a region of 100K while the long version needs 190K. The libraries SYSI,CERNLIB and CR.PUB.PRO.GENLIB4 both contain MINUIT.

In CMS, the following commands should be used:

```
FILEDEF 5 DISK MINUIT DATA  
FILEDEF 6 TERM (LRECL 133  
GLOBAL TXLIB FORTLIB GENLIB KERNLIB  
RUN MYMINU
```

where it is assumed that MYMINU contains the precompiled MAIN program and that FCN TEXTI is available. MINUIT is held in GENLIB TXLIB. MINUIT will run interactively online in CMS if Fortran stream 5 is directed to the terminal.

#### Change to the CERN Library

The second stage in the rationalisation of routines common to RHELIB and CERNLIB has been implemented. The routines UZERO, UBLANK and UFILL in SYSI,CERNLIB and KERNLIB TXLIB will now issue a warning message if they are called with 2, 2 and 3 arguments respectively but the routines will still work correctly. You are reminded that in 3 months time the routines will be modified to cause an abnormal termination if they are called with the wrong number of arguments. If you get the warning messages you should modify calls to those routines by either calling VZERO, VBLANK or VFILL instead of by adding an extra second argument.

```
eg CALL UBLANK (A,N)  
should be modified to  
CALL VBLANK (A,N)  
or CALL UBLANK (A,1,N)
```

Unless the routines are overlaid, the messages will only be issued once.

C P Wood - User Support Group

## 7. CMS ARCHIVE

The CMS archive system has been installed. It runs permanently disconnected in the virtual machine ARCHIVE and manages users' archived files. Archived files are kept on a CMS mindisk which is unloaded to tape when it fills.

When files have been read in by the archive, a message or file is sent to say that the archiving is complete. For security, files that have been sent to the archive should not be erased from the users' own disk until at least 24 hours after this acknowledgement has been received - this is so that the file will also appear in the CMS dump system. Otherwise only one copy of the file would exist. When the archive disk fills, it is dumped to tape and at this stage a backup tape is also written for security.

Files are kept by the archive until the user erases them. A directory of archived files is kept for each user and contains the following information:

Original fileid  
Archive date and time  
Record format (F/V)  
Record length  
Number of records  
Number of MK blocks occupied by the file  
Current location (DISK or a tape number)  
Creation date and time  
Virtual address of the file that was sent to the archive  
Generation number of the file  
User specified tag.

Multiple copies of a file with the same fileid can be kept in the archive. In this case the archive assigns a generation number to each copy of the file. Users may specify a tag of up to 8 words or 32 characters which will be stored in the directory entry for the file and could be used, for instance, to group together a set of files. Selection criteria that may be used to specify a file or group of files to be retrieved or erased are fileid, archive or creation date, generation number, address or tag. Files may also be retagged using the same selection criteria.

The archive has access to its tapes only at certain fixed times of day - at present this is 10.00, 15.00 and 17.30. Requests which require tape mounts will be held until the archive next obtains a tape drive. This may include dumping of the archive disk itself when it is full. The response to the ARCHIVE command may be delayed if the archive is in a tape session or has many files on its reader - waiting to be read in. The status of the archive machine can be examined using the command QARCHIVE.

See the ARCHIVE and QARCHIVE help files for further details.

J M A Elades - Systems Group

## 8. NETWORK STATUS MACHINE

There is a machine on the SERC network, currently situated at Daresbury, which contains information on the status of the RAL and DL batch systems. It is accessed by making a network call to NETSTAT, where the appropriate information will be printed at the terminal.

Now the above is true if you have a direct access to a network machine, but what do you do if you want information about the IBM system when it is down and you have a non-networked link? There are three main cases to consider.

(a) Users connected to the RAL PACX system, either directly or over the telephone lines will need to call a networked machine on PACX to make the NETSTAT call. Suggested machines are:

Machine	PACX Code	Action
PRIME	RLPB, RLPA	Press RETURN until you get a response then type LOGIN -ON NETSTAT
GEC	RLGB	Press RETURN until you get a request for 'user' then type !NETSTAT

(b) Non-networked workstations. Such workstations are dependent on the IBM host machine and can only make calls to NETSTAT when the IBM host is running. If the workstation is connected to HASP then the NETSTAT machine may be contacted (to read notices about availability) as follows:

```
++B DEST N  
!NETSTAT
```

and when the call completes remember to restore

```
++B DEST E
```

If the workstation is connected to VNET then precede the above by

```
++V DIAL CMS  
DIAL FEM
```

and finish with ++V DROP

If the IBM host is down there are two alternatives you may consider. If you have access to PSS, possibly via a local area network, you may make a PSS call to SERCNET. Details on how to do this are given in the document The SERCNET - Switchstream 1 Gateway, User's Guide, available from Roy Platon (ext 6188). The other method involves making a switched public network call to PACX. That requires a terminal with a modem attached to a standard telephone line. The telephone number for PACX is 0235-834531, then follow the instruction under (a).

(c) 3270 Screens. These are connected locally. It is necessary to find another terminal on PACX to access NETSTAT.

R E Thomas - Head of User Interface Group

## 9. THE SUSSEX UNIVERSITY POPLOG SYSTEM

### Introduction

The Research Area Review Meeting on Intelligent Knowledge Based Systems, held in London on 22-23rd Sept recommended that POPLOG be added to the SERC's common software base. This article gives a brief overview of the system. Further details may be obtained from the author.

POPLOG combines two languages POP2 and PROLOG, in an interactive environment including screen-editor, HELP system and teaching system. At present it runs on VAX computers under the VMS operating system and a reduced version is available on a Z8000 under ZEUS. There are plans to transfer it to a PERQ and other machines. A VAX UNIX version will be available later. The system is portable as most of it is written in itself, apart from a small machine-dependent kernel.

The core language, POP-11, is an extended dialect of POP-2 originally developed at Edinburgh University for research in Artificial Intelligence. POP-11 is very similar in function to LISP, though with a syntax more like conventional block-structured programming languages. Like some LISP systems, POP-11 is compiled for efficiency. Interactive program development makes use of the fact that the compiler is part of the run-time system and that reclamation of storage is automatic.

PROLOG, the programming language based on Logic is also included. POP-11 procedures can call PROLOG procedures and vice-versa. Thus in a complex design it is possible to implement modules in whichever language is more appropriate. This means that PROLOG procedures can access procedures which manipulate floating point numbers, arrays, strings and other data-types, which is not normally possible in PROLOG.

Although most AI research in the USA is based on LISP, POP is found more approachable by programmers used to more conventional languages. Its syntax is richer, making it more readable and enabling the compiler to give more helpful error messages. It has a cleaner semantics than most LISP systems eg the relationships between procedures and variables. For teaching or demonstration purposes a 'toy' LISP system is provided as a library package.

### Programmer Productivity vs Machine Efficiency

POPLOG could usefully be employed for any application where program development costs are significant or where the restrictions built in to more conventional languages introduce design difficulties. Although arithmetic operations are slower than in Fortran or Pascal, the system is being used for research involving number-crunching, namely image analysis and speech processing. For the design of very complex programs, the reduced program development time more than compensates for increased run time. As computing power becomes cheaper and software development costs rise, run-time efficiency becomes just one factor in a complex equation.

Where run-time efficiency is crucial, a 'system' dialect of POP-11 is used, for instance in the garbage collector, the editor, the compiler and most of the system procedures which are written in

POP-11 itself, as is the core of the PROLOG system.

POPLOG speeds up program development in a number of ways. First, it is a fully interactive system so that new commands can be given in the language at any time, procedures can be edited and re-compiled without a tedious re-linking process, the programmer can respond to the run-time behaviour by defining temporary procedures to interrogate data-structures, etc. Moreover, the automatic store management, including garbage collector, removes many programming problems. The 'macro' facility allows the programmer to extend the expressive power of the language to suit particular applications. The use of a pattern matcher, together with syntax for specifying structures by their format, allows programmers to think about structures in terms of how they look instead of in terms of low-level procedures for constructing or analysing them and this reduces programmer errors. The PROLOG system is a powerful tool for designing problem-solving programs which need to explore alternative combinations of strategies.

In addition to providing many powerful language constructs, POPLOG provides a development environment which can save a great deal of programmer time. The user-extendable screen editor, VED, allows several files to be worked on simultaneously and permits up to two of them to be visible on the screen. (Several customers report finding it the best editor they have used.) VED can be used for writing documentation as well as programs and can be asked to compile a marked portion of the file so that there is no delay caused by writing files and switching between editor and compiler. Besides, output from programs can be diverted into editor buffers and later written to disk.

The importance of good program development environments is increasingly widely recognized as the costs of software development spiral upwards. The price, in terms of lost efficiency, of AI systems is now less important because the cost of hardware has dropped and AI programmers have learnt how to make their user-friendly systems more efficient. For instance, POP-11 allows large programs and data-structures to be 'locked in' thereby minimizing time taken by garbage collections.

During program development POPLOG is often MORE machine efficient than most compiled languages because the compiler is very fast and need only re-compile altered portions of a program. With editor and compiler integrated into the run-time system, re-compiling and re-linking a modified procedure definition can take only a fraction of a second.

To illustrate the productivity claim: since much of the system is written in itself and can therefore be tested interactively during development, we were able to develop it in record time. The initial version of the compiler for the VAX took about three months, the first version of VED about 4 weeks. Both have been considerably extended since, yet it is only a year since work was begun and very little man-power has been available. How long did it take to produce a useable version of your favourite compiler?

The need for AI techniques in practical applications is being recognised increasingly, eg in improving the man-machine interface and many

Industrial or commercial programmers will need to learn to use AI systems. PROLOG provides an ideal environment for such learning, as well as research and development.

#### The Language

The core language, POP-11, is relatively conventional. It is based on familiar concepts such as sequential execution, variables, data-structures and procedures and the syntax has much in common with the ALGOL family. A programmer familiar with, say, PASCAL would not (at first) experience any culture shock. Several students familiar with PASCAL have been able to teach themselves to use POP-11 with relatively little help. It does not take them long to appreciate the considerably greater power of POP-11. Familiar programming concepts are extended by a variety of unusual features including a built in pattern matcher, extended control constructs, hash-coded association tables, UNIX-like stream I/O, the ability to build procedures at run time, macros for extending the syntax, a timed interrupt mechanism and the PROLOG subsystem.

AI languages give far more flexibility than conventional languages. Because POP data-structures can be type-free, it is possible to write programs with far greater generality than in typed languages. For instance, there is a set manipulation package in the library which can be used for sets of arbitrary objects. PASCAL would not allow a procedure to take as argument a predicate which could be applied to arbitrary objects: the type-restrictions provide a gain in efficiency and compile time consistency checking at the cost of intolerable limitations (and increased compilation time).

PROLOG is based on the use of predicate logic, which is given both the usual declarative interpretation and also a procedural interpretation. It is specially suitable for the design of programs which are given factual information from which inferences are to be made, for instance the design of data-bases where much of the information is implicit in general principles. PROLOG rules are compiled into POP-11 procedures for efficiency. These, in turn, compile to machine instructions.

Many (though not all) DEC-10 POP-2 programs will run with no alteration in POP-11 as the syntactic differences are handled by autoloadable library 'macros', i.e. procedures which run at compile time and re-arrange text before it gets to the compiler.

#### The Text Editor, VED

Good editing facilities are very important for programmers working on complex systems. VED is both powerful and easily user extendable since its procedures are written in POP-11 and can be extended using POP-11. It can also be called by users file-manipulating programs. Normally, the user communicates with the rest of the system through VED.

For ease of use, VED was designed for a terminal with a large number of labelled function-keys, such as the VISUAL-200. However, it is easily converted for other terminals. Library programs for the VT100 and Televideo (with VT-52 emulation) are provided.

VED has built in knowledge of some of the syntax of POP-11 so that it can be asked to compile, or to re-format, the 'current' procedure. It will usually find the beginning and end of the procedure automatically.

#### Debugging Tools

Explicit debugging tools are less necessary than with conventional programming languages, mainly because the compiler is part of the run time system, so that, for example it can be invoked at break-points. This allows the user to give any POP-11 command, not only to examine or change variables but also to edit and recompile procedures or to define new procedures to help explore complex data-structures during a break. Thus a debugging operation which would require a lengthy edit-compile-relink-restart process in a conventional language will be much faster in POP-11, saving programmer time and machine time.

During break points, produced by errors, planned pauses or the typing of an interrupt character, the procedure POPREADY enables the user to communicate with the system in the same language as is used for writing programs. So the language inherently provides very powerful debugging tools. In some cases POP enables execution to be restarted at an earlier point, after one or more procedures have been redefined or the environment altered. This uses the 'CHAINTO' mechanism.

#### Saved Images

It is possible to save the current state of one's computation in a disk file, and restart it later. This is useful during testing and may be used to avoid re-compiling frequently used packages. Saved images can allow intelligent learning programs to accumulate knowledge over several login sessions.

Compilation of frequently used utilities can be speeded up with the 'incremental' SAVE and RESTORE facility.

#### The Library

POP-11 has an 'auto-loading' mechanism that causes library files to be automatically compiled and included into any user program that references them. The user can alter the list of 'auto-load' directories to be searched. In addition there is a library of programs which are loaded by an explicit 'LIB' command. Teams of programmers can easily share libraries. VED can be instructed to alter the autoloadable library depending on what sort of file is being used, since the editor runs a user-definable procedure every time a file is set on the screen.

The SHOWLIB command makes it easy for users to examine library programs and to produce their own modified versions.

#### The Documentation System

There are several levels of documentation: 'help' files, 'teach' files, 'reference' files and 'manuals'. The screen editor greatly simplifies access to documentation for the on-line user. The editor command HELP FOR tells VED to read in the 'help file' for 'for'. This can remain visible when the user returns to his own file. Instead of typing a command, the user can merely press the 'help' key while the cursor is next to the word

'FOR'. If the information given is insufficient there will in some cases be a longer tutorial file which can be accessed using the command TEACH FOR. A third level of documentation, 'reference files', are provided for more sophisticated users, accessed by a command such as REF SYNTAX

A user who doesn't know the precise name of a help file, but thinks it may include, say, 'word', can type HELPFOR WORD and will be given a list of possibly relevant help files, eg

#### WORDS DATAWORD WORDSWITH

Because of liberal cross-referencing, finding a partly relevant file will normally quickly lead the user to the precise information required. HELP can be used as a tree-structured menu system. However, since the editor allows arbitrary switching between files there is no need to be restricted to a particular tree structure. While looking at file A, you can go up or down within it, or switch to a new file B, or return to where you were in a previous file C, saving your location in A. This overcomes the restrictions of tree-structured menus.

The files intended as off-line documentation can be accessed via the DOC editor command. The off-line documentation is not yet as complete as the on-line documentation. Off-line manuals and primers will be available later.

Dr A Sloman - Sussex University

#### 10. COMPUTER STATISTICS

IBM SYSTEMS 4/10/82 - 31/10/82

The following graphical presentation of performance statistics are now a regular feature.

