

No. 5/78  
23Nov78

UNIVERSITY OF QUEENSLAND

Prentice Computer Centre

MINI/MICRO NEWSLETTER

Contents

- 1.0 WORDS FROM THE GURUS
- 2.0 DIRECTOR'S COMMENTS
- 3.0 PDP11 SOFTWARE NEWS
- 4.0 RSX PERFORMANCE ANALYSIS
- 5.0 DATA ACQUISITION SYSTEMS FOR PDP11'S

authorisation: Director of the Centre



## 1.0 WORDS FROM THE GURUS

"Don't raise your voice and argue noisily. The quiet man who keeps his wits about him and watches his opportunity, generally has a six-to-four chance. Don't think it always incumbent on you to repeat the dreary formula, "Any complaints?" when visiting your men in their barrack-rooms or quarters. Such an inquiry in hospital, for instance, is obviously superfluous, and in married quarters it is discourteous."

Young Officer's Don't (1883)

## 2.0 DIRECTOR'S COMMENTS

In the recent allocation of funds for Central and Departmental computing equipment, provision was made for the purchase of interfaces for the connection of departmental minicomputers to the central facility. The method of connection depends to some extent on the amount of traffic to be transported and upon the level of facilities required.

John Barker, Systems Analyst on our staff, is currently in the U.S. and one of his briefs is to study this area and where possible obtain software from other installations. I would, therefore, wish to wait his return in December before finalizing our strategies.

My aim is to send a memorandum to each department (having a minicomputer) around January outlining the options and requesting details of your requirements. At this stage, however, you may wish to give some thought to your needs. Arthur Hartwig would be pleased to be of assistance.

Departments have received advice of their capital allocations for computing equipment in 1979. It is, we believe, in a Department's interest to consult the Centre and channel orders through us for initial computing equipment, add-ons and microprocessors. This will ensure that equipment is ordered under contracts (approved by the Legal Officer) which provide both technical and commercial protection for the University. As far as possible, we are anxious to rationalize equipment types (including plotters, digitisers, etc.) to keep the longer term costs of spares, maintenance and software support to a minimum.

As regards microprocessor development, the Computer Centre

has already some support facilities and has received a funds allocation to purchase a Tektronix 8001 Microprocessor Laboratory (or similar). This equipment will enable the Centre to provide an effective service to assist Departments in the development of microprocessor systems. It is most desirable that, before Departments purchase microprocessor systems, they should consult the Centre to ensure that their proposed purchases are compatible with the support facilities of the Centre. It should be kept in mind that the development costs of microprocessor systems can be high relative to their capital cost.

Some Departments have occasional need for magnetic tape on their systems but the purchase of magnetic tape units would not be justified. The Centre proposes to introduce a new service in 1979 to provide for the conversion of magnetic tapes to RL01 or RK05 disk packs.

As mentioned in a previous newsletter, the Prentice Computer Centre has acquired a licence on behalf of the University for UNIX software. The only CPU authorised to use it under our licence agreement at this stage is the Department of Computer Science. There are no great problems in extending the licence to other CPUs but certain formalities are required by Western Electric U.S (the owners of UNIX), otherwise our University licence may be prejudiced. Please arrange for your Departmental Head to write to me if you require a licence and distribution of UNIX software to your installation.

Director  
extension 2189

### 3.0 PDP11 SOFTWARE NEWS

#### 3.1 Corrections to Documentation for RSX-11M V3.1

The following corrections have been released by DEC: The numbers associated with the corrections are those assigned by Digital and are published to help us record those corrections of which users have been notified.

##### (7.1.1.1) LPS11 RELAY CALL

The LPS11 Relay call has wrong argument values.

The argument values as stated in the I/O Drivers Reference

Manual are incorrect. In paragraph 12.4.17 the values of IREL should read "ZERO FOR RELAY ONE, ONE FOR RELAY TWO)".

#### (7.1.1.2) ERROR IN K-SERIES SAMPLE PROGRAM

The sample program for the K-series lab peripherals in section 18.4.2 of the RSX-11M I/O Drivers Reference Manual has a minor error in it having to do with setting up the completion routine.

The two lines.

```
CALL ADSWP (IBUF,1024,-1,256,IPRSET
* AST,0,0,1)
```

should be replaced with:

```
CALL GETADR (TEMP,AST)
CALL ADWSP (IBUF,1024,-1,256,IPRSET,
* TEMP,0,0,1)
```

#### (7.1.1.3) ERROR IN K-SERIES SUPPORT ROUTINES SAMPLE PROGRAM

There is an error in the sample K-series program using completion routines. (I/O Drivers Manual)

In subroutine AST, insert the following line immediately before the line "DIMENSION BUF(1024,8), IBUF(40), IOSB(2)":

```
IMPLICIT INTEGER (A-Z)
```

#### (7.1.1.4) UNDOCUMENTED TKTN MESSAGE

When the system checkpoint file is full, the error message: "Checkpoint space allocation failure" occurs from TKTN. This message is not documented.

This message means that the executive has tried to checkpoint a task to the system checkpoint file, but there was not enough room in the file to contain the task image.

To solve the problem, use the "ACS" command to create another checkpoint file or to make the present one bigger.

#### (7.1.1.5) FORMATS GIVEN FOR IO.LDI, IO.LTI, IO.LTL, AND IO.LKE ARE WRONG

The formats given in the I/O Drivers Reference Manual for the following ICS/ICR-specific I/O functions are incorrect:

IO.LDI IO.LTI IO.LTY IO.LKE

Use the formats given below instead:

```

QIO$ IO.LDI,.....,<tname1,tname2,,[tevf],pn,csn>
QIO$ IO.LTI,.....,<tname1,tname2,,[tevf],cn[,ic]>
QIO$ IO.LTY,.....,<tname1,tname2[,,,tevf]>
QIO$ IO.LKE,.....,<tname1,tname2[,,,tevf]>

```

where tname1 is a word containing the first 3 characters of the taskname, tname2 is a word containing the second 3 characters of the taskname, and the rest of the parameters are as listed in the I/O Drivers Reference Manual.

Example of use:

```

.MCALL QIO$,QIO$C,QIO$$

TASKA: .RAD50 /TASKA/
QIOI:  QIO$  IO.LDI,.....,<^RTAS,<^RKB >,,32.,0,177777>
START:  QIO$C IO.LTI,.....,<^RTAS,^RKBB,,33.,0,500>
        QIO$$ #IO.LTY,.....,<TASKA,TASKA+2,,34.>
        MOV   #TASKA,RO
        QIO$$ #IO.LKE,.....,<(RO),2(RO),,35.>
        DIR$ #QIOI

.END

```

This program links TASKA to error and remote terminal interrupts, TASKA to digital interrupts, and TASKBB to counter interrupts. Note that if a taskname is less than 4 characters in a QIO\$ or QIO\$C call, the second word of the task name must be padded with spaces.

#### (7.1.1.6) READ\$ RETURNS EOF ONLY TO F.ERR

In section 3.15.1, the I/O Operations manual states, "If the virtual-block number references nonexistent data, an end-of-file (IE.EOF) error indication is returned to the I/O status block... and to offset location F.ERR of the associated FDB". This is not true. The error code is returned to F.ERR; it is not placed in the I/O status block.

#### (7.1.1.7) CLARIFICATION OF EOF (F.EFBK) HANDLING WITH READ\$/WRITE\$

There is no documentation in the I/O Operations Reference Manual about FCS handling of the F.FBY and F.EFBK fields in the FDB during READ/WRITE operations. It is not clear if the user must use these fields to determine the end-of-file

which would be set by FCS, or whether the user must set these fields explicitly.

FCS maintains both the F.FFBY and F.EFBK fields during READ\$ / WRITE\$. When a READ\$ is done, FCS uses these two fields to determine the location of the end of the file. If F.FFBY is nonzero, then the block pointed to by F.EFBK is the last block in the file and it will be delivered whole by FCS (with no indication other than the contents of F.FFBY that it is not filled). If F.FFBY is zero, then the EOF is at the end of the block before the one pointed to by F.EFBK, and FCS will return EOF when an attempt is made to read the block pointed to by F.EFBK.

On WRITE\$, FCS makes the same decision about EOF in order to determine whether the write will pass EOF. If it will, the file is extended to provide space for the data. In this case, WRITE\$ will also set F.EFBK to one more than the VBN of the last block in the file, and set F.FFBY to zero. This will cause the EOF to be reported in the appropriate place when the file is read with either GET\$ or READ\$.

In summary, it is NOT necessary for the user to maintain F.EFBK and F.FFBY in READ\$ / WRITE\$ mode. FCS will perform all necessary bookkeeping. If these fields are modified by the user, unpredictable results may occur.

#### (7.1.1.8) SPECIAL NOTE ON USE OF \$STMAP, \$IODON, and \$IOALT

The executive routines \$STMAP, \$IODON and \$IOALT push the address of routine \$DQUMR on to the stack before returning to the caller. This precludes the use of the stack for temporary data storage by drivers when calling these routines.

Note that the release notes mentioned this for \$STMAP, but it is otherwise undocumented.

#### (7.1.1.9) ERROR IN SAMPLE PROGRAMS

The sample programs of Section 18.4 of the I/O Drivers Reference Manual (AA-2539D-TC) each have the same bug. The line:

```
CALL RLSBUF(IBUF,IND,IBUFNO)
should be:
CALL RLSBUF(IBUF,IND,IBUFNO-1)
```

### 3.2 PROBLEMS IN RSX-11M V3.1

Some software restrictions have been published recently in the Software Dispatch. The following extracts are relevant to existing installations.

(5.1.6.4) DSC MAY NOT PROPERLY COPY CONTIGUOUS OR LARGE FILES

Problem 1: When DSC copies a disk, individual files on that disk which are defined as contiguous are actually non-contiguous on the output disk some of the time, even though merge mode was specified for the original copy from disk.

Problem 2: DSC can not copy large (multi-header) files correctly.

Problem 1 will be eliminated in a future release. The error in DSC can not be patched. Please note that a contiguous input file copied in non-merge mode will no longer be defined as a contiguous file on the output disk.

Problem 2 will also be eliminated in a future release. The errors which cause this problem are also non-patchable. This problem only arises in files which require more than one file header to map all of the blocks in the file. Note that a maximum of 26,112 virtual blocks can be mapped in a single file header.

(5.1.8.1) INI - ENTERING BAD BLOCKS MANUALLY

When bad blocks are entered manually, the information is not placed in BADBLK.SYS.

Whenever the MANUAL mode option of INI is used, the AUTO option should be specified. Example:

```
INI>DBO:TEST/BAD=[AUTO,MAN]
```

(5.1.12.1) REPROTECTING HIGHLY PROTECTED FILES

If a user has protected a file, so that he has no access to it, there is no way to change the protection on the file.

RESPONSE

A file which has been protected against all access may be reprotected by using the /PR:0 switch with PIP.

For example:

```
PIP>TEST.FTN/PR:0
```

will give Read, Write, Extend and Delete access to all users. The file may then be reprotected as desired.

#### (5.1.12.3) RESTRICTION WHEN MERGING FILES WITH PIP

Merging or appending files with different record attributes (macro listing and map files) does not always produce the desired results.

As pointed out, macro listings and task-builder maps cannot be merged. Macro listings have the FD.CR attribute with variable length records. The task-builder maps have zero carriage control (the carriage-return and line-feed are embedded in the text). FOR listings have zero carriage control and F4P listings have the FD.CR attribute.

#### (5.1.12.5) PIP MAY DISCARD NOSPANBLOCKS ATTRIBUTE

When using PIP to copy a file from disk to disk, PIP will fail to copy the FD.BLK (NOSPANBLOCKS) attribute.

The attribute may be preserved by use of the /SB switch. This will be considered for a future release of PIP.

#### (5.1.12.9) PIP /FO SWITCH FAILS IN MERGE MODE

When merging files, the /FO switch does not set the file owner properly.

This will remain a restriction in the current version of PIP.

Bryan Claire  
extension 3940

### 3.3 Recent Software Releases

The RSX-11M system monitoring program RMDEMO has been modified to run on ADM3A video terminals. Appropriate software is also available for administering logout disk block quotas. Enquiries concerning implementing these software enhancements should be directed to Bryan Claire on extension 3938.

#### 4.0 RSX PERFORMANCE ANALYSIS

Recently to assist Griffith University decide on equipment upgrades for their PDP11/34 a set of programs was produced to help discover the effect of extra memory on the elapsed time to perform certain tasks. The system may be very easily adapted to determine the effect of changing other system parameters or of a different job mix.

The package includes a data collection task, two tasks which send the time of day to the collection task, a task to place commands into a terminal input buffer (all written in MACRO-11) and a simple analysis program written in FORTRAN. The indirect command processor (part of the standard system software) is also heavily used.

A command is placed in the required number of terminal input buffers to have a command file processed at that terminal. The command file is processed a number of times and at the start and end of each pass the current time of day and terminal number is sent to the data collection task which calculates the elapsed time for each pass on each terminal and writes it to a file along with the terminal number.

A small very low priority task is fixed in core. It continuously increments a counter and after command files which have been initiated have finished it sends the value of its counter to the data collection task. Since this task runs when nothing else can the value of its counter provides a measure of the system idle time (the time when no task was able to run).

Some tests were performed to compare the effect of using the large task builder (BIGTKB) with using the fast task builder (FTB installed with a task increment of 8K words as recommended in the manual) and to see the effect of adding more memory. An RL01 was the system and swapping disk and an RK05 was used to store all the temporary output files, source files and command files. A FORTRAN program to tabulate the SINE of angles between 10 and 360 degrees in 10 degree increments was compiled, task built and run. The object file and task file produced was then deleted. There was a 5 second delay between issuing each command and the whole process was repeated four times for each terminal. The executive and related services occupied less than 24 Kw of memory so that the remaining memory was available for tasks run in response to the issued commands.

The results obtained are tabulated below where total elapsed time is the time between the time when the first terminal became active and the time when the last terminal completed its fourth pass.

80Kw total memory using BIGTKB

No. of active terminals	Average elapsed time per pass	Total elapsed time	% CPU idle time
1	1 min. 12	5 min. 11	89
2	1 min. 31	6 min. 30	76
4	4 min. 18	19 min. 12	64
6	5 min. 50	27 min. 38	62
8	5 min. 03	35 min. 36	64

Note that the system had become saturated at 6 active terminals. The reduction in average elapsed time per pass in going from 6 to 8 terminals is really due to the fact that the system was so busy with 6 active terminals that the 7th was unable to become active until the first had finished. Once the 7th and 8th terminals became active they completed much quicker than the first six due to the reduced number of terminals active. This is borne out by the increase in total elapsed time and CPU idle time in going from 6 to 8 terminals.

The next set of figures demonstrate a considerable improvement gained by using the fast task builder.

80KW total memory using FTB

1	58 sec.	4 min. 10	96
2	1 min. 12	5 min. 25	80
4	2 min. 36	12 min. 12	72
6	3 min. 40	18 min. 14	71

The next set of figures are for the same runs using BIGTKB but with 124Kw of memory on the system.

4	1 min. 54	8 min. 31	56
6	7 min. 12	30 min. 59	53
8	8 min. 05	42 min. 33	54

Comparison with the first set of figures shows a dramatic improvement for 4 active terminals but a considerable worsening of response time as more terminals become active. Adding more memory does not always improve system response time.

The next set of figures are for a 124Kw system but using FTB.

4	1 min. 42	8 min. 12	66
6	4 min. 33	19 min. 06	60
8	5 min. 30	25 min. 58	59

Again these figures demonstrate an improvement over those obtained using BIGTKB and an improvement over the 80 Kw system for 4 active terminals but a degradation for more than 4 active terminals. Why the degradation in response as a result of adding more memory?

It was proposed that this was due to the whole system becoming I/O bound. Swapping, task-builder temporary files, task builder overlays and searching the system library all caused I/O to the same disk. The more memory on the system the more copies of the task builder could initiate I/O requests and the more these could interfere with one another by causing seeks to distant parts of the disk. To test the I/O bound hypothesis commands were issued to direct swapping and task-builder temporary files to a second RL01. Task builder overlays and the system library were still read from the system disk. A single test with 8 active terminals and using BIGTKB yielded the following figures.

8	5 min. 54	27 min. 55	39
---	-----------	------------	----

These figures indicate an improvement over both previous sets obtained using BIGTKB tending to substantiate the I/O bound theory. Though the average time per run is worse than the figure for 8 terminals and 80 Kw memory the total elapsed time to do the same work is significantly reduced indicating a better response time at each terminal.

#### 4.1 Interpretation of Results

The results CANNOT be used to directly determine the number of terminals a particular RSX system will support. The job mix and what is regarded as acceptable response time determine the number of terminals which may be included in the configuration.

The effect of running a number of copies of the indirect command processor on the response time is not known. Since obtaining these results it was discovered that an executive "feature" prevented the indirect command processor from lowering its priority. This adds to the swapping load. I hope to run some tests with this feature removed and publish the results in the next newsletter. It must be remembered that a number of users using the text editor does not add much to the overall load since each copy of the editor is swapped out during terminal input (the time between the issuing of the prompt and the typing of the following carriage return). This is a considerable proportion of time.

#### 4.2 Conclusions

When performing program development it is quite obvious

sitting at a terminal that the task-building phase is the most time-consuming. The disk lights and noises from the disk indicate task building generates considerable I/O activity. The figures obtained indicate that four to six active copies of the task-builder using the system disk for swapping is as much I/O as a single disk can be expected to perform. In multi-user systems where more than 4 copies of the task-builder (or any other large I/O bound task) are active, worthwhile system performance improvement can be gained by directing swapping and task-builder temporary files to a disk which is otherwise little used. Note that though the controller treats an RK05F as TWO logical disks it is only ONE disk in that it has only one set of heads and seeks cannot be done on both logical disks simultaneously. Of course in many circumstances worthwhile improvements can be obtained through the use of FTB rather than TKB and without the cost or inconvenience of system reconfiguration. There are some circumstances though in which FTB can not be used. (See Task Builder Reference Manual, Appendix H).

#### 4.3 Acknowledgements

I wish to acknowledge the assistance of Bryan Claire and Tony Griffiths, DEA who coded some of the tasks and assisted with the design.

Arthur Hartwig  
extension 3021

#### 5.6 DATA AQUISITION SYSTEMS FOR PDP11'S.

In a number of laboratory situations there is a need for aquisition of data in a variety of forms. These forms typically are;

1. Analog voltages from strain gauges, thermocouples etc.
2. BCD outputs from DVMS etc.
3. Pulse trains for specialised devices.

The experimental site is often situated some distance from the computer site and this factor imposes some constraints on data aquisition.

These constraints are;

1. Noise induced on analog lines run over lengths greater than say three metres unless special techniques are used.
2. The necessity for the operator to control the data acquisition program and to control the experiment.
3. High costs of multi-line cabling linking experimental site and computer site.

Equipment which is purchased as add-in computer equipment (i.e. mounted in the computer backplane) is ideal for environments where the experimental site is situated in the same room. For other instances its use must be questioned if meaningful results are to be obtained.

In an effort to provide PDP11 users with a means of obtaining high precision data cheaply from remote experimental sites, the Prentice Computer Centre has developed a microprocessor based data acquisition system (MPDAS). Such a system would be portable unit which would be movable from site to site as necessity dictates. The only connection requirements would be a single standard high-speed (typically 1200 baud) communications line.

The MPDAS would typically consist of a base processor board, power supply, backplane (common to all systems) and other modules customised to the user's needs.

The base system provides the following;

1. Microprocessor system with
  1. 512 bytes of RAM
  2. up to 4k bytes of EPROM
  3. 22 bits of programmable parallel I/O.
2. Communications link to local terminal and PDP11.
3. Custom user modules which could typically provide the following;
  1. Analog to digital conversion e.g. 16 channel single ended or 8 channel differential, 12 bit resolution, etc.

2. Digital to analog conversion.
3. Extra digital parallel I/O lines for aquisition or control functions e.g. BCD input from DVMS.
4. Specialised interfaces for user equipment.
5. Extra RAM for high speed "burst" aquisition.
6. General Purpose Interface Bus support. (IEEE-488)

Software support supplied by the Prentice Computer Centre is proposed to cover the following areas;

1. An operating system for the microprocessor unit to cover startup initialisation and essential features such as driver modules and maintenance functions.
2. Modules to allow transparent communication with a PDP11 for experimental startup and general communications (such as result printout etc).
3. Modules for data transmission to a PDP11 with error checking and correction features.
4. Initialisation and control routines for handling parallel I/O devices, timers and other system devices.
5. Parameter setup modules to control data aquisition modes.

Additional software could be supplied to control other user specified interfaces. Facilities are also available for users to produce their own software and to have this software loaded into EPROM.

Communication rates between the MPDAS and the PDP11 are dependent on the baud rate, distance apart, and the environment through which the cables are run. RS-232-C is specified only over relatively short distances at high speed. Newer standards such as RS-422 and RS-423 allow significantly larger cable lengths to be used at higher baud rates. The later two interface standards however require special interfaces at the PDP11. The suggested baud rate for RS-232-C is 1200 baud, allowing a (14 bit) sample from any combination of sources to be made each 20mS (i.e. 50 samples per second) allowing for overheads in error correction and formatting. Higher speeds are quite possible depending on installation requirements mentioned previously.

Preliminary Hardware Specifications.

1. A microprocessor board consisting of
  1. An 8085 CPU
  2. 512 bytes of RAM
  3. Up to 4k bytes of EPROM
  4. A 14 bit programmable timer
  5. 22 bits of programmable I/O
  6. Interrupt capability
  7. MULTIBUS bus structure
  
2. An interface board consisting of
  1. Two USARTs with selectable baud rate
  2. RS-232-C or RS-422 interface
  3. Three 16 bit programmable timers (optional)
  4. Extra parallel I/O in blocks of 24 bits (optional)
  
3. Dedicated boards such as
  1. Analog to digital converters, typically 12 bit, 16 channel single-ended or 8 channel differential with optional programmable gain and/or digital to analog converters.
  2. Custom user interfaces and/or system resources such as memory.

Anybody interested in the use of such a system with a PDP-11, please contact the Prentice Computer Centre for further information.

Colin Lythall.  
extension 3938