DECUS

The DeVIAS Letter

Delaware Valley IAS Local User Group

April 1983

Issue No. 14

In This Issue

Contributions

The DeVIAS Letter, the newsletter of the Delaware Valley IAS User's Group, needs contributions from members in order to continue as an effective medium for exchange of information regarding IAS. All contributions should conform to the standards set by the Multi-Tasker, the IAS/RSX SIG Newsletter. These are:

All contributions should be camera-ready copy (i.e. sharp black type in a 160 x 240 mm area (8.5 x 11 inch paper with one inch margins)) and should not include photocopies. If you use RUNOFF to prepare your contributions, the following parameters have been found to be satisfactory:

```
.PAGESIZE 60,80
.LEFT MARGIN 0
.RIGHT MARGIN 72
.SPACING 1
```

These parameters assume output onto a line printer with a pitch of 10 characters per inch. Adjust them to maintain the same margins if another pitch is used.

Send contributions to:

Ontario Hydro
700 University Ave.
Toronto, Ontario
Canada, M5G 1X6

Attn: John W. Drummond
Mail Stop - M4E5

From the Editor

Here, at last, is the closest thing we are likely to get to an IAS Internals Manual. It is without a doubt more useful to those of us with source licences, but then that would be true regardless of the level of detail.

The Fall 1982 U.S. DECUS provided two papers from the IAS development group: one on the Node Pool Problem and a proposed solution and another on ACP's and their use in IAS. The IAS product panel declined to discuss the contents of IAS Version 3.2, but appear to be attempting to provide maximum benefits with minimal expenditure.

If any of you have suggestions, get them to Bob or myself and we'll pass them on to Tim Leisman, the IAS Product Manager. My own priorities include

1) PDP 11/24 support (apparently trivial)
2) Digital Storage Architecture Support (RA80, RA81, RA60, TU80)
3) Open-ended solution for SCOM depletion and memory deadlock problems.

In the longer term, other mechanisms such as autopatch or perhaps even access to RSX-11M distribution could be used to provide new releases of utilities such as PIP and BRU.

There appears to be some relief coming in the pricing of IAS DZ licences but no intention of including IAS under the General licence umbrella with RSTS and RSX-11M plus.

Keep the Faith

J.W. Drummond

Department of Radiation Therapy
University of Pennsylvania
Room 410
133 South 36th Street
Philadelphia, Pennsylvania 19104

Thanksgiving Day, 1982

Dear DeVIAS Member,

This issue is late and it is my fault. Mr. Drummond has had the body of Issue 13 ready for several months. I am sorry for the long delay.

For many of you, this will be the first issue since Issue 10. The problem arose during the transition activity at DECUS. The U.S. Chapter is no longer responsible for any mailings to Europe, Canada or Australia. The data bases that supported the mail labels were separated into the respective Chapters. For a while no one noticed that DeVIAS members were not getting the Letter. Julie Cibelli, who works at DECUS, noticed and called me for a complete copy of our mailing list. She said that she would fix it. The Letter is now the only publication of the U.S. Chapter that is so widely distributed. I have a few copies of Issues 11 and 12 for those of you who missed them and ask for them.

There have been some changes in the continuing saga of "Digital and the IAS Community". There will be a Version 3.2, for example. It was announced at the Philadelphia DeVIAS meeting a month ago. The five or six of us that showed up heard some other things too: The "Development Team" is now entirely in Maynard. Our friends in Reading, England have other jobs now and the entire effort is in the U.S. I find that comforting. There was a long delay in communicating problems to Maynard and then to Reading and back the same tortuous route. It is not that the people in Reading were in any way lacking, quite the contrary - they were outstanding, but the path via Maynard was too long. They built us a fine operating system but we should have been able to talk directly to them, not via SPR Administration or any other Eastern Massachusetts communication impediment.

Another point made at the Philadelphia meeting by Tim Leisman, the IAS Product Manager, and Bonnie Morrisey, from U.S. Area Software Product Services, was that support would change in some respects in June 1983. She sent me a copy of the details and it is enclosed in this issue. Further, IAS will have two solid representatives from the Development Team at DECUS in Anaheim. One will present a tutorial on F11ACP and the other will address the "Node Pool Problem." The quality of their presentations (and handouts that will appear in the Letter) are offered as testimony of their ability to SUPPORT IAS in the future. Mr. Leisman ducked the question: "How many people are in the Development Team", but assured the questioner that these two at least would allay our fears caused by the presentation in Atlanta.

On the whole, then, I was impressed by Digital's words. The fact that they still call it a "DEVELOPMENT" team is encouraging. I am anxiously waiting for Anaheim. I am convinced that Mr Leisman will do all he can to promote IAS and IAS Support. He said, "A first rate company doesn't dump people."

Thank you for your strong responce to my letter inviting you to rejoin DeVIAS. In some ways I was hoping that no one would answer and DeVIAS could be laid to rest. But, there are about one hundred of you that returned the form. Thank you (I think). Some of you even supported the DeVIAS Letter to the extent that you included checks for twenty dollars. I will return them, when I set the time, for DECUS will continue, for the immediate future, to publish us. It it very comforting to know that only one respondent said that he would not pay for the Letter. Thank you for all the nice things that the rest of you said.

If you did not receive an invitation to renew your membership, please notify me. I did fail to send them to people who had only recently joined.

Another small detail was taken care of at the meeting. By acclamation our new bylaws were passed. DECUS has taken to renewing the license of each LUG every year. I have been living on waivers for a while, trying to get the required documents together. The bylaws, which have never existed before, are required. Our bylaws are due to the effort of Bob Stodola, our Librarian; Thank you Bob. So, our new "Operating Procedures" are also enclosed in this issue. I would like to hear any comments on them you would care to commit to paper.

I wish you all a merry Christmas, happy Chanukah, and a prosperous New Year. And, I thank you for your support during this year.

Sincerely,
Robert F. Curley

OPERATING PROCEDURES

Article I
Name

1.0 The name of the organization is the Delaware Valley IAS Local Users' Group (DeVIAS).

Article II
Purpose

2.0 DeVIAS is established under the bylaws of the DECUS/U.S. Chapter to:

1. Provide a forum for users of the IAS operating system to exchange ideas, programs, and any other items of common interest.

2. Provide feedback to Digital Equipment Corporation (DEC) on all matters concerning the IAS operating system, related software products, services, policies, and all DEC manufactured computers, peripheral equipment, and other hardware products.

Article III
Membership

3.0 Membership requirements:

1. Any person using or interested in the IAS operating system or its related products, equipment or services who is in the Delaware Valley or any other area without an "IAS Only" Local Users' Group is qualified to be a member.

2. Any person qualified to be a member will be accepted as a member upon submitting a completed and signed membership application to the chairman.

3.1 Rights of members:

1. Members shall have the right to vote for all DeVIAS Elected Officers.

2. Ten or more members of DeVIAS may, by written petition, bring a motion before a meeting of the LUG Steering Committee.

Article IV
Steering Committee

4.0 General

1. DeVIAS shall be administered by the Steering Committee.

2. The Steering Committee shall consist of four officers and up to two at-large members.

3. Any member of DeVIAS may be on the Steering Committee and the Steering Committee shall be composed solely of members.

4. The Chairman may act independently on all matters, and shall inform and consult with the Steering Committee as (s)he sees fit. A majority vote of the remaining members shall be required to override decisions of the chairman.

4.1 Steering Committee Officers

1. The Steering Committee Officers shall serve until resignation, or until removed by the Chairman or by vote of no confidence by members.

2. The officers are the Chairman, the Newsletter Editor, the Program Chairman, and the Program Librarian.

4.2 At-large Members

1. The chairman may appoint up to two At-large Members of the Steering Committee.

4.3 Duties of the Chairman

1. The Chairman runs the show. Due to the wide geographic distribution of its members, frequent meetings of the Steering Committee or Officer elections are impractical. Therefore, the Chairman shall discharge all duties normally associated with the Chairman as well as those of the Steering Committee. The Chairman is subject to the review of the Steering Committee, or recall by vote of no confidence of the members.

4.4 Duties of the Newsletter Editor

1. The Newletter Editor shall edit and publish the 'DeVIAS Letter', the LUG newsletter.

2. In the event that the position of Chairman becomes vacant, the Newletter Editor shall temporarily assume all duties of the Chairman except that of Steering Committee appointments until a permanent Chairman is found.

4.5 Duties of the Program Chairman

1. The Program Chairman is responsible for the planning and scheduling of meetings.

2. In the event that both the position of Chairman and Newsletter Editor become vacant, the Program Chairman shall temporarily assume all duties of the Chairman except that of Steering Committee appointments until a permanent Chairman is found.

4.6 Duties of the Tape Librarian

1. The Tape Librarian shall maintain copies of such non-proprietary programs deemed of interest to DeVIAS members, and furnish copies of these to DeVIAS members on request.

2. In the event that the positions of Chairman, Newsletter Editor, and the Program Chairman become vacant, the Tape Librarian shall temporarily assume all duties of the Chairman except that of Steering Committee appointments until a permanent Chairman is found.

4.7 Vacancy in Office

1. Should the Chairman vacate his(her) office by resignation, disability, or inelisibility, a new Chairman shall be appointed by a majority vote of the remaining officers.

2. Should any other officer vacate his(her) office by resignation, disability, or inelisibility, the Chairman shall appoint a replacement.

Article V
Elections

5.0 Removal of Officers

In accordance with Article III, the Steering Committee will accept any motion to remove an officer of DeVIAS. The motion will be presented in the next Newsletter along with

the comments of the remaining Steering Committee members and a request that members file a vote on the motion within 30 days. Should a majority of respondents comprising at least 1/4th of the membership at the time of the Newsletters distribution agree to the removal, the officer is removed, and must be replaced by election by the members, as described below.

5.1 Nominations

Should an officer be removed, or all three Steering Committee officer positions become simultaneously vacant, nominations for that position will be accepted by the Newsletter Editor, or the person designated to function in that capacity. The nominees will be contacted, and shall accept by filing a brief statement in their behalf to be published in the next Newsletter. All members may return the ballot published in that copy of the Newsletter. The nominee receiving the most votes will be elected and take office immediately.

Article VI
Meetings

6.0 General meetings

Meetings shall be scheduled approximately six times per year. Two of these meetings shall be at the Spring and Fall DECUS U.S. Chapter symposiums to allow geographically distant members to attend.

6.1 Steering Committee meetings

The Steering Committee shall meet by phone prior to each general meeting, or at the Chairman's request.

Article VII
Amendments

7.0 Amendments to these operating procedures shall be made in the same manner as removal of officers above.

A.R.A.P.
AERONAUTICAL RESEARCH ASSOCIATES of PRINCETON, INC.
50 WASHINGTON ROAD, P.O. BOX 2229, PRINCETON, N.J. 08540 . . . (609) 452-2950

September, 15, 1982

Joe Volonakis
Zamil Soule Steel Building Co., Ltd.
P.O. Box 270
Dhahran Airport
Saudi Arabia

Dear Mr. Volonakis:

In the August 1982 issue of The DeVIAS Letter you requested the changes necessary to run SPY on IAS V3.1. The module SPYTI2.MAC needs to be modified, no other changes are necessary. I have enclosed a DIFFERENCES listing between the V3.0 (SPYTI2.MAC) and V3.1 (S31TI2.MAC) version of this module. Make the necessary source code changes, assemble and link.

Frank Borger of the Michael Reese Medical Center has noticed a problem with SPY in a heavy swapping/shuffling environment. It appears the CPU time is not in any of the places DEC says it should be. If you experience these symptoms, you may want to contact Frank to see if he has a work-around. His address is:

Frank R. Borger
Instrumentation Division Head
Department of Medical Physics
Michael Reese Medical Center
29th Street and Ellis Avenue
Chicago, IL 60616

I have also enclosed the V3.1 modification to another A.R.A.P. utility, DSM (Disk Storage Monitor), for future reference. The change in this case involves the patch to the PDS login module.

Please feel free to contact me if I can be of any further assistance.

Sincerely,

John D. Leonard
Manager, Computer Facilities

JDL/oh

Enclosure: as stated

cc: The DeVIAS Letter

```
**********
File DBA1:[360,245]S31TI2.MAC;1
    8   ;++2    05/28/81 LEN    - MODIFIED FOR IAS VERSION 3.1. ACCOUNTING INFO
    9   ;                         NO LONGER ALWAYS STORED IN ATL. REFER TO RELEASE
   10   ;                         NOTES FOR MORE DESCRIPTION.
   11   ;
******
File DBA1:[360,245]SPYTI2.MAC;1
    8   ;
**********
**********
File DBA1:[360,245]S31TI2.MAC;1
   55           BNE     5$              ;;NO
   56           JMP     RETURN          ;;BR IF SO
   57   5$:
   58           MOV     (R1),R1         ;;GET A NODE
******
File DBA1:[360,245]SPYTI2.MAC;1
   52           BEQ     RETURN          ;;BR IF SO
   53           MOV     (R1),R1         ;;GET A NODE
**********
**********
File DBA1:[360,245]S31TI2.MAC;1
  133   ;
  134   ;       FOR IAS VERSION 3.1 THE ACCOUNTING WORDS ARE NOT LONGER STORED
  135   ;       IN A.TAC AND A.TAC+2. REFER TO VS 3.1 RELEASE NOTES FOR AN
  136   ;       EXPLANATION OF WHERE THE ACCOUNTING INFORMATION IS AT A GIVEN
  137   ;       TASK STATE.
  138   ;
  139   ;++2    ADD     A.TAC+2(R1),10(R3)      ;;LOW WORD OF CPU TIME
  140   ;++2    ADC     12(R3)                  ;;CARRY TO HIGH ORDER WORD
  141   ;++2    ADD     A.TAC(R1),12(R3)        ;;HIGH ORDER WORD
  142   ;++2    TST     A.HA(R1)        ;;HAS TASK BEEN LOADED?
  143   ;++2    BEQ     30$             ;;IF NOT, FLAG SWAPPED OUT
  144   ;++2    BIT     #AT.TR,A.TST(R1);;TASK RESIDENT?
  145   ;++2    BNE     30$             ;;NO, FLAG SWAPPED OUT
  146   ;++2    BIT     #AF.CP,A.TF(R1) ;;IS TASK CHECKPOINTED?
  147   ;++2    BEQ     40$             ;;BR IF NOT
  148   ;++2 30$:       BIS     #'* ,16(R3)     ;;FLAG SWAPPED OUT
  149   ;++2 40$:
  150   ;++2    BR      ATL
  151   ;
  152   ;
  153           MOVB    A.TAI(R1),R2    ;;GET ACCOUNTING STATE
  154           JMP     @DSPTCH(R2)     ;;DISPATCH FOR PROPER HANDLING
  155   ;
  156   DSPTCH: .WORD   30$             ;;TASK INITIALIZATION
  157           .WORD   40$             ;;TASK SWAPPED OUT
  158           .WORD   50$             ;;TASK IN MEMORY
  159           .WORD   60$             ;;TASK EXITING
  160   ;
  161   30$:    CLR     10(R3)          ;;NO ACCOUNTING INFORMATION AVAILABLE YET
  162           CLR     12(R3)
  163           BR      70$
  164   ;
  165   40$:    MOV     A.JN(R1),R2     ;;SWAPPED, INFORMATION IN JOB NODE
  166           ADD     J.SX+2(R2),10(R3)       ;;LOW ORDER WORD
  167           ADC     12(R3)                  ;;CARRY
  168           ADD     J.SX(R2),12(R3)         ;;HIGH ORDER WORD
```



```
  169           BIS     #'* ,16(R3)     ;;INDICATE SWAPPED OUT
  170           BR      70$
  171   ;
  172   50$:    MOV     #77406,-(SP)    ;;IN MEMORY, INFORMATION IN TASK HEADER!
  173           MOV     A.HA(R1),-(SP)  ;;REMAP APR3 TO THE APPROPRIATE HEADER
  174           CALL    @#..SPD3        ;;MAP TO IT
  175           ADD     60000+H.TAC+2,10(R3)    ;;LOW ORDER WORD
  176           ADC     12(R3)                  ;;CARRY
  177           ADD     60000+H.TAC,12(R3)      ;;HIGH ORDER WORD
  178           CALL    @#..SPD3        ;;MAP APR3 BACK
  179           CMP     (SP)+,(SP)+     ;;CLEAN UP THE STACK
  180           BR      70$
  181   ;
  182   60$:    ADD     E.TAC+2(R1),10(R3)      ;;EXITING, INFO IN RE-USED ATL NODE
  183           ADC     12(R3)                  ;;CARRY
  184           ADD     E.TAC(R1),12(R3)        ;;HIGH ORDER WORD
  185   ;
  186   70$:    JMP     ATL
  187   ;
******
File DBA1:[360,245]SPYTI2.MAC;1
  128           ADD     A.TAC+2(R1),10(R3)      ;;LOW WORD OF CPU TIME
  129           ADC     12(R3)                  ;;CARRY TO HIGH ORDER WORD
  130           ADD     A.TAC(R1),12(R3)        ;;HIGH ORDER WORD
  131           TST     A.HA(R1)        ;;HAS TASK BEEN LOADED?
  132           BEQ     30$             ;;IF NOT, FLAG SWAPPED OUT
  133           BIT     #AT.TR,A.TST(R1);;TASK RESIDENT?
  134           BNE     30$             ;;NO, FLAG SWAPPED OUT
  135           BIT     #AF.CP,A.TF(R1) ;;IS TASK CHECKPOINTED?
  136           BEQ     40$             ;;BR IF NOT
  137   30$:    BIS     #'* ,16(R3)     ;;FLAG SWAPPED OUT
  138   40$:
  139           BR      ATL
  140   ;
  141   ;
**********

Number of difference sections found: 3
Number of difference records found: 60

DIFFERENCES /MERGED=1/OUTPUT=DBA1:[LENJX.DIF;1-
    DBA1:[360,245]S31TI2.MAC;1-
    DBA1:[360,245]SPYTI2.MAC;1
```

```
        .TITLE  LOGI
        .IDENT  /V03.1/
;
;       ARAP SPECIFIC PATCH.
;
;       MODIFIES LOGIN PROCEDURE SO THAT ALL LOGIN'S RUN ...NOT TO CHECK
;       DISK STORAGE USEAGE.  ALL NOTICES ARE FROM LB0:[1,1]NOTICE.TXT
;
;       MODIFIED 5/29/81 BY J. LEONARD TO ADJUST FOR IAS VERSION 3.1
;
        .PSECT  LOGI,RO,I
;
LOGI=.
;
.=LOGI+2702
        NOP             ; FORCES /NONOTICE LOGINS TO RUN ...NOT ALSO
;
;       PATCH LGINOT TO CALL ...NOT AND CHECK NOTICE FLAG. IF
;       NOTICE THEN SEND ...NOT TI:=LB0:[1,1]NOTICE.TXT; IF NO NOTICE THEN
;       SEND NOT TI:=LB0:[1,1]BATNOTICE.TXT.
;
.=LOGI+224
LGI.NO  =       001             ; NOTICE FLAG
        BIT     #LGI.NO,QUFLG
;
.=LOGI+240
        MOV     2(R3),R2        ; ADDRESS OF NF1: OR BF1: ('LB0:').
        MOVB    #1,(R2)+        ; HEADER BYTE OF "1
        MOVB    USRUIC,(R2)+    ; MOVE UIC TO COMMAND LINE
        MOVB    USRUIC+1,(R2)+
        MOVB    #1,(R2)+        ; TRAILER BYTE OF "1
        NOP
        NOP
        NOP
        NOP
        NOP
        NOP
        NOP
        NOP
        NOP
        NOP
        NOP
        NOP
        NOP
        NOP
        NOP
;
;       CHANGE PIP TO NOT
;
.=LOGI+330
        .ASCII  /NO/
        .ASCII  /T /
;
;
        .END
```



```
!
! 11/27/79      - APPLY ARAP PATCH TO PDS LOGIN PROCEDURE LOGI.OBJ
! 05/29/81      - MODIFIED FOR IAS VERSION 3.1 DISTRIBUTION
!
!
!       PDSODL.ODL MUST BE EDITED TO REPLACE THE NORMAL LOGIN MODULE
!       WITH THE PATCHED LOGIN MODULE.
!
!       IN [11,100]PDSODL.ODL CHANGE -
!
!
!       01170:  .FCTR   LOGI-[11,100]PDS/LB:LOGI
!                       TO
!       01170:  .FCTR   LOGI-[11,100]ARAPLOGI
!
EDI [11,100]PDSODL.ODL
MCR
ON SEVERE CONTINUE
PIP [11,100]ARAPLOGI.OBJ;*/DE/LD
ON SEVERE STOP
MAC [11,100]ARAPLOGI.POB=[11,100]V31LOGI.PAT
LBR [11,100]ARAPLOGI;1=[11,100]PDS/EX:LOGI
INS [11,1]PAT
PAT [11,100]ARAPLOGI;2=[11,100]ARAPLOGI;1/CS:076113,ARAPLOGI.POB/CS:026161
!
TKB @[11,100]PDSTKB
!
REM ...PAT
DCL
```

IAS Support

U.S. Area Software Product Services is pleased to announce the consolidation of telephone and on-site support for IAS and all layered products, including DECnet/IAS and DBMS-11/IAS, at the Sudbury Customer Services Support Center. These services will be available through June 1983.

Support provided by the Sudbury Support Center includes:

- Installation
- On-site and telephone remedial services during warranty
- Telephone remedial services for customers having BASIC service and DECsupport contracts
- On-site remedial and preventive maintenance support for DECsupport customers
- Consulting services

Since July 1, 1982, telephone support has been available to all IAS warranty and SPS customers from 8:00 A.M. to 5:00 P.M. local time, nationwide, using a toll free number (800-343-5734). Massachusetts customers call collect. All Colorado CSC registration information has been automatically transferred to the Sudbury CSC.

Effective July 1, 1983 - SPS will offer Self-Maintenance Service. The Self-Maintenance Service being offered for IAS will consist of the following:

- Software problem reporting service - individual response. All responses to be published.
- IAS Software Dispatch
- Autopatch service
- Software product updates

SPR:
IAS 3.1 FMT 01.11 14-APR-82
Problem:
Online Disk-Formatter FMT causes system loop

When FMT is used to format a RP03 disk, the system runs into a high-priority loop. Only contr.C starting MCR reactivates the system. DEMO showes no activity (Null-task active), even the system clock is stopped. This problem seems to occur when any other task is doing IO via F11ACP and/or DP-handler.

Concerned tasks:

DP....	06	GEN	248
TT....	V03.00	GEN	248
F11ACP	D0322	GEN	220
...MCR	V42	GEN	230
...FMT	01.11	GEN	54

ANSWER:
Up to now no answer (24-sep-82)!

SPR:
IAS 3.1 RUN V004A 4-MAY-82
Problem:
Lower case letters.
Can RUN be changed to accept also lower case letters to run task.

Answer:
PROBLEM:
Character case considerations on RUN command.
RESPONSE:
Thank you for your SPR.
Users of timesharing systems will find that PDS handles all case conversions they can issue commands in upper, lower or mixed cases. However, some of the older system tasks perform their own command string parsing and often only check for uppercase characters. It would be quite an effort to convert all of these and at this late stage in the product life of IAS it is very unlikely that the cost could be justified.

PS: This is in my opinion a very unsattisfying answer !

New SPR:
IAS 3.1 EXEC 22-SEP-82

Problem:
Data Parity Error on Swapping disk moves taskstatus to IR4.

An occasional data parity error occuring in the swap file (ERRLOG: Hard Error) causes a task status of IR4. The task is then blocking the system, because no abort is possible. Manually clearing the I/O count (ATL + 12) will free the memory but not the used nodes, used temporary files (i.e with F4P) are not deleted. There is no way to exit the task in a normal way, in case of such a possible data error.

SPR:
IAS 3.1 AUTOPATCH "B" 8-JUN-82

Problem 1:
AUTOPATCH does not return a corrected file to the System-disk, when used with a separate work disk.

The file [11,15]EXEC.ODL is changed with EXECODL.COR and is used to link the new EXEC, but is only on the work disk.
The file EXEC.ODL on the real SYSTEM disk is left unchanged, this causes APR problems as soon as the EXEC has to be linked again with following exec-patches.

Problem 2:
Missing Message files on the work disk.

The Autopatch procedure should copy [1,2]message files to the work disk (actual SY:). In case of an error there is no useful error message.
Example: missing WK: disk, LBR produces a fatal error 31 ...

ANSWER:
PROBLEM:
The problems with the Autopatch Kit "B"
RESPONSE:
Thank you for your SPR in which you outlined two problems with the Autopatch Kit "B".
Problem (I) Autopatch does not return a corrected file to the System disk, when used with a separate Work disk.
Please find enclosed a preview of an article which addresses this problem.
Problem (II) Missing message files on the work disk.
This problem will be considered for any future IAS V3.1 Autopatch Releases. We would point out, however, that there are space restrictions involved with the work disk in the Autopatch Process, particularly when the disk is an RK05. To copy message files to the work disk from the system disk would reduce the amount of space available for Autopatch files from the tape. This may result, for example, in the process becoming much slower, and consequently this point will also have to be considered.

SPR:
IAS 3.1 F4POTS V3.0 14-APR-82
K. Centmayer Munich
Inst. f. Datenverarbeitung
Technische Universitaet Muenchen
Franz-Josef-Str.38, 8000 Muenchen 40

11/45 820 120k MT 9 RP 03

Problem:
F4POTS does not always handle Floating-Point Interrupts.

When the FP Unit encounters an Underflow the Value should be set to Zero, this is not done in every case, instead a large (positive exponent) value is returned.
This seems to bee a timing problem, because system activity (bus activity) increases the error rate.
The same promlem exists with FP overflow (Error 72 turned off).

The system hardware is o.k. , same problem at other hardware.
This problem exists only under IAS not under RSX 11M.
This was tested on a system normaly running 11M, same error.

Same problem with FORTRAN 77 (V4.0).

ANSWER:
PROBLEM:
Floating point exception error.
RESPONSE:
Thank you for your SPR. The problem you have reported is due, under certain circumstances, to the Floating point exception service being delivered to the wrong task.
Please find enclosed a preview of an article which will correct this fault.

PS:
Article is in Software Dispatch Aug.1982 Seq. 2.1.1.7 (is o.k.).

SPR:
IAS 3.1 BRU 1.01 19-APR-82

Problem:
BRU Errors.

BRU reports a Select-error on MT: and switches tape off-line, but realy expects only the continuation tape; please provide a better message.
To copy one file with /VER four times a new reel must be mounted.

BRU reports END OF TAPE 1 ON MT0: but continues to read (or something else) on the tape.

BRU reports an error in [001,001]file, indeed the file is in UIC 1,2 but UIC 1,1 was an empty UIC.

After copying a single file BRU with /VER reports 2 files and twice the number of blocks.
BRU reports tape-error, but not always, which file is realy bad.

The second atempt to copy a bad file brings a warning FILE NOT SUPERSEDED, and endless error messages (I/O -1) in the verify pass.

BRU produces mysterious messages: Fatal I/O error code -16 (nothing has been changed !). Fatal Device not in system DP2.

BRU produces files which are empty or nonsense.

ANSWER:
PROBLEM
STATEMENT
The user reported several BRU errors.
RESPONSE
Thank you for bringing these problems to our attention. However, in order to solve the problems we definitely need more information. We believe that most of the reported problems are fixed in the Autopatch "C" for IAS.
If you encounter these problems again, please resubmit the SPR enclosing the console listing, a map of BRU running on your system and any other information you think will help to solve the problems.
The article with all new BRU patches which were included in the Autopatch "C" for IAS is attached for your convenience to this SPR response.

PS: Autopatch "C" has a lot of BRU-patches , not tested up to now.

SPR:
IAS 3.1 EXEC 3-MAY-82

Problem:
Interaction of SWA , UTL , SAV et al.

A system cannot be saved with the Swap-file on another disk. (without the /NOIN switch). But a system can be saved with the sheduler enabled. The swap-file can be removed with the sheduler enabled, without an error message. But without a swap file starting of TS-programms is not possible, there is simply no reaction, and no message, why there is no reaction. Now a SAV with the sheduler still enabled, what is normaly possible, is with no, or a very small swap-file not possible, because the DMO will not start, and that is a fatal error.

ANSWER:
PROBLEM:
Interaction of SWA, UTL and SAV.
RESPONSE:
Thank you for your SPR.
In order that the swap file can be saved in the system it must reside on SY0: (the system device) or on a dedicated volume (i.e. a non-removable medium such as DS0:).
As regards being able to delete the swap file while the scheduler is enabled, enclosed is a preview of a forthcoming publication article which addresses the problem.

PS: There is a patch for ...SWA: A swap file cannot be deleted with the sheduler enabled. Not published up to now.

SPR:
IAS 3.1 EXEC 23-APR-82

Problem:
Pseudo Devices OV: and PI:

There are two devices, which can be found in the list of the logical units of a task (e.g. TKB) OV0: and PI0:. These devices are not in the list of the symbolic devices (SYS /DEV) and are not generated in a System-generation (the line ;DEV=PI is commented out in SYSGEN.CMD). These devices cannot be REDirected, but can be REAsigned (with unknown effect).
Please provide some documentation concerning OV: and PI:.

ANSWER:
PROBLEM:
Information required on Pseudo Devices.
RESPONSE:
Thank you for your SPR.
Pseudo Device PI is the name of a handler task that contains the Task Control Primitives. They are used by CLIs to initiate and monitor their sub-tasks and can be accessed by users of the Task Control Services (refer to the "Writing Command Language Interpreters" manual for further information).
IAS can be generated in three different types of system but some component software (notably TKB) is required to be common to them all and so must modify its action to suit the type it is running on. So it will try to queue an I/O request to PI0:, upon failure (as in your case) it knows it is running on a Real-Time or Multi-User system and so can take appropriate action.
Pseudo Device OV is referenced by the task builder when it is building an overlayed task. The LUN assigned to this device is used to access overlay segments. It should always be left assigned to the disk containing the load image, i.e. its initial value.
It is possible to REASSIGN both these devices for a task (with suitable privilege) but in doing so (particularly for a privileged task) you can seriously degrade system performance.
SPR:
IAS 3.1 PIP DI332 3-MAY-82

Problem:
The switch /FI in PIP to access a file via file ID number does not work always.

With the /FI switch only a file copy is possible.
Listing (/LI and /FU) is not possible. BAD DIRECTORY FILE ...
Renaming (/RE) produces strange errors. (see attached listing.)
Deleting (/DE) only partially deletes a file. "

Especially listing would be very useful to find a file which is reported in error by other utilities (e.g. BRU).

ANSWER:
PROBLEM
STATEMENT
The switch /FI in PIP to access a file ID number does not always work. Specifying the /FI, /LI, and/or /FU switches together do not list the directory information for the file. Instead, PIP issues the error message "BAD DIRECTORY FILE".
RESPONSE
The only function of the /FI switch is to list the contents of directory files (UFDs), for example [0,0]005222.DIR (UFD [5,222]). The UFD to be listed may be the default directory or one specified in the command line as [g,m]. It is also possible to specify the UFD to be listed by supplying its File-ID using the /FI switch.
Under no circumstances can the PIP /LI switch be used to list the directory information of a given file when only the File-ID of the file is known. The /FI switch checks the file header for the owner and UFD of the file. It cannot get this information from any of the separate files in a directory, only from the directory file itself ([0,0]005222.DIR).
The only way to gain any information about the owner (and therefore possibly the UFD of such a file) is to use the DMP utility to examine the file header with the /HD switch. Note that it is generally impossible to always determine the UFD of a file from the file's header, because information concerning renames is not tracked there.
We realize that the Utilities Manual is also a little misleading when it states that "to list a directory file whose identification number is 301,27...". But it is correct, as it means that the file ID that you specify must be the file ID of a directory file (i.e. [0,0]005222.DIR, UFD [5,222]).

SPR:
IAS 3.1 DSC X0036 24-MAY-82

Problem:
DSC terminates with memory protect violation

Trying to copy a disk to another disk (both foreign mounted) results in an exit of DSC.

```
DSC>DP2:/VE=DP3:
DSC -- 84 INPUT DISK NOT BOOTABLE

DSC -- *WARNING* 56 OUTPOT DISK DP2: IS NOT BOOTABLE

TASK "...DSC" TERMINATED
MEMORY PROTECT VIOLATION
  PC=063702
  PS=174000
  R0=063713
  R1=000015
  R2=000000
  R3=000002
  R4=063616
  R5=063646
  SP=000776
```

ANSWER:
PROBLEM
STATEMENT
If the V3.1 version of DSC detects a bad file header while copying to a disk, it may trap as a result of clearing the bit in the bitmap which corresponds to the bad file header. The first of the following patches eliminates that possibility. If a primary header of a file on an input disk is found to be bad, that entire file will not be copied. DSC may also trap in this case unless the second patch is applied.
RESPONSE
In this version of on-line DSC, the Index File bitmap of the input disk is copied directly to the output media on the assumption that the output bitmap will be an exact copy. If an invalid header is detected on the input disk and the output is a disk, that file is not copied and the corresponding bit in the bitmap is cleared. Under the same condition but with tape output, the file is also by-passed but it is impossible to correct the bitmap. Later, when a disk is restored from that tape, there is no indication to DSC that an error was detected during creation of the tape. Thus the new output disk created from the tape has a bit in its bitmap corresponding to a non-existent file. VFY will detect and report the invalid bit(s). The best way to clean up the disk is to copy it directly to another disk so that the invalid bits may be cleared. Version 3.0 of IAS handled this area differently but in a much slower fashion.
Two problem areas have been discovered in the V3.1 version which may be corrected by the following patches to module DWTID and module DFIND. The following patch DWTID eliminates the cause of a trap, which may occur during a disk to disk copy, because of the clearing of a bit in the bitmap. The stand-alone DSC s cannot be patched. The patch to DFIND eliminates the cause of a trap when trying to access the next valid header on the input disk after the detection of a bad header. This trap is not dependent on the type of output medium.
Use the following procedure to apply the patches.
PS: Two patches for DSC, up to now unpublished.

30 Aug 82

Robert F. Curley
University of Pennsylvania
School of Medicine
Department of Radiation Therapy
3400 Spruce Street
Philadelphia, PA 19104

Bob:

Sorry for the delay. I have improved the notes for the talk to include what I would like to have done if there had been more time for preparation and presentation. Everything that was in the talk as I gave it is included. I also edited the document that Richard DeMorgan prepared to correct a few minor errors. Both are on the tape, which is in DOS format, 800 BPI.

I am now part of the VMS support group. As things stand now, this means that I won't be going to Disneyland this December. Perhaps some future DECUS.

I also want to thank you and Ray French for the help that you gave in Atlanta. The success of the Q and A was due in large part to this help. Thank you.

Sincerely,

Rod Shepardson

CONTENTS

1.0 ABBREVIATIONS

The followings abbreviations are used consistently both in IAS Executive listings and in this document:

ACP	Ancilliary Control Process
ADB	Attachment Descriptor Block
APR	Active Page Register
ASQ	AST queue
ASR	Active Segment Register (same as APR)
AST	Asynchronous System Trap
ATL	Active Task List
CDA	Crash Dump Analyser
CIT	Command Interpreter Table
CKQ	Clock Queue
CLI	Command Language Interpreter
DEQUE	Double-ended Queue
DIC	Directive Identification Code
DLT	Device Load Table
DPB	Directive Parameter Block
DVT	Device Table
FCB	File Control Block
FTL	Fixed Task List
GCD	Global Common Directory
IOSB	Input/output Status Block
IRQ	Input/output Request Queue
ISR	Interrupt Service Routine
JNP	Job Node Pool
LBN	Logical Block Number
LRG	Load Region (task state)
LUT	Logical Unit Table

MCR	Monitor Console Routine
MFD	Master File Directory
MRL	Memory Required for Load
MUL	Memory Usage List
OTS	Object Time System
PAR	Page Address Register
PDR	Page Descriptor Register
PLAS	Program Logical Address Space
PUD	Physical Unit Directory
RDB	Region Descriptor Block
RDL	Region Descriptor List (same as GCD)
RRQ	Receive by Reference Queue
SFL	Swap File List
SGA	Sharable Global Area
SRQ	Send/receive Queue
SST	Synchronous System Trap
STD	System Task Directory
STL	Spawned Task List
TCP	Timesharing Control Primitives
TCS	Timesharing Control Services (macros)
TNP	Terminal Node Pool
TPD	Task Partition Directory
UFD	User File Directory
UIC	User Identification Code
UIT	Unit Information Table
UJN	User Job Node
UMR	Unibus Mapping Register
UTL	User Task List
UTN	User Terminal Node

VCB	Volume Control Block
WDB	Window Descriptor Block.

2.0 EXECUTIVE STRUCTURE

The resident executive consists of two images that exist in real memory. They are both constructed as tasks (by the Task Builder.) One image is a bootstrap, XXXXXBOOT.TSK, and always resides at real memory location zero. The other is the executive, EXEC.TSK. The overlay description is as follows:

base	top	
000000	025413	EXEC
140000	157777	SCOMM
040000	057523	DIR
040000	057463	ASX

INTERRUPTS } EM02, EM04, EM06,
EXCEPTIONS } EM08

POWER UP/DOWN — EM03

ASTs — EM07

DIRECTIVES — EM10 + DM...

MEMORY MANAGEMENT — ESTPAR, ESMEM, TSMEM

I/O — ESEXIO, ESSFL, ESSWAP

ATL SCANNING SCHEDULING — EM05, TSSHED

PRIMARY AREAS OF THE IAS EXEC

3.0 EXEC MODULE SUMMARIES

3.1 EM00

Macros for:

- interrupt vector format
- set/put a word from/into previous address space
- crash system.

Glossary.

Symbolic definitions for:

- bit patterns in program status word
- external page labels
- PDP-11/70 registers
- system trace (T-bit) debugging aid conditionals
- indices to SST vectors
- termination notice codes
- event flag bit masks
- common dequeue node words
- task partition directory
- memory usage list
- global common directory
- physical unit directory
- system task directory
- active task list
- timesharing ATL linkage
- task header offsets
- region and window descriptor blocks
- E.xxx offsets
- swap file list
- user task list
- I/O request queue
- allocation and deallocation
- clock queue
- AST queue
- send/receive queue
- send/receive by reference queue
- spawn task list
- MCR command buffer
- task states
- system event recognition flag
- UMR support.

3.2 EM01

KVZERO trap vectors (kernel virtual zero).

BOTKSP Kernel stack area.

.SG.OT Executive impure area.

3.3 EM02

Return to interrupted program:

- .INTXT normal ISR exit
- EXINTX, ..ERTZ EXEC ISR exit
- ENDDIR process end of directive.

3.4 EM03

Power fail and recovery service:

- PWRXDN power down
- PWR.UP power up.

3.5 EM04

Service for:

- TRAP04 traps at location 4
- SFAULT segment faults
- TRPTRP TRAP instructions
- EMTTRP EMT instructions
- BPTRP T-bit traps and BPT instructions
- IOTTRP IOT instructions
- RESERV reserved instructions
- FEXTRP floating point exception interrupts
- SSTCOM SSTs.

3.6 EM06

CKI Clock interrupt service routine.

CTREC Clock tick recognition.

3.7 EM07

.DAST0, .DAST1 AST declaration.

3.8 EM08

FEXREC Service for floating point exception interrupts.

3.9 EM09

This module was split into the modules ESASX, ESDIR, ESEXIO, ESMEM, ESMISC, ESREQS, ESSFL and ESSWAP.

3.10 EM10

DIRECT System directive dispatch.

DIRDPT Unprivileged directive dispatch table.

PDIDPT Privileged directive dispatch table.

.DRSRN System directive return.

3.11 EMNUL

NULTSK The null task.

3.12 ESTPAR

Support for IAS type partitions:

.MULFR	free memory
.TACTK	call .ACTK for a timesharing partition
.IASFR	free memory in an IAS partition
.GATSP	get memory for a task segment
.IASFM	get contiguous space in an IAS partition
.IASFC	find/free block of memory contiguous to allocated memory
HOLEFD	find first hole large enough
MEMOVE	shuffle occupied memory
SWMVCK	see if memory can be shuffled
SWSTCT	relocate task root when shuffled
SWDROP	move block of memory
FREESP	free space.

3.13 ESASX

ATL scan routines:

.AIWAK	wake up handlers on task load
.CLNGA	clean up task's global areas
.FLRRQ	check for sends by reference on task exit.

3.14 ESEXIO

Executive I/O routines:

.IONOD	select and clear I/O node from pool
.RDTSK	read task root segment
.WTTSK	write task root segment
.RDRGN	read region
.WTRGN	write region.

3.15 ESMEM

Memory manipulation routines:

.ACTK	allocate memory to task
.FMEM	locate contiguous block of memory in partition
.FCMEM	locate free memory contiguous with allocated block
.FREM	free allocated memory
.FRESG	free contiguous segment in partition.

3.16 ESMISC

Miscellaneous routines:

.CEFN, .CEFN1	validate and convert event flag
.CKDEL	delete node from clock queue
.CKINS	insert node in clock queue
.CLMEM	clear memory
.CREQS, TREQS	request tasks for executive
.CRGCD	create GCD node for region
.CTIT	convert time interval to ticks
.DLRG	delete region
.EXCU	clean up for exiting task
.FLRCQ	flush send/receive queues
.FSTD	search STD
.HCKSM	compute header checksum
.LUNPT	convert LUN to LUT address
.MPNOD	fill in memory parity error log node
.NDSCH	search ATL for task
.PPOOL	select node from pool
.RELES	release regions on task exit
.PGNSW	remove region/SGA from memory
.RLRG	release region
.RMCR	return MCR command line node to pool
.SPDR	set PDR contents for task
.SRCH	search list for task
.STSPN	make STL node into ATL node on task exit
.TSKRS	resume task
.TSKUS	unstop task.

3.17 ESREQS

.REQS attempt to make a task active.

3.18 ESSFL

SFL manipulation routines:

.SWALL allocate swap file space
.SWDAL deallocate segment of swap file
.SWLBN fill in request node for swap I/O request
.SWTRN translate swap file block number.

3.19 ESSWAP

Swap/checkpointing control routines:

.SWAP make space available
SWGCD remove unaccessed SGAs from memory
SWSTOP remove stopped tasks from memory
SWTASK remove other tasks from memory
SWATSC scan ATL for swapping
SWAPIT try to remove task from memory.

3.20 TSMEM

Memory allocation routines for timesharing tasks:

.TSWAP obtain space for a task
.LVRES reset UTL level head
SCUTL scan UTL for tasks to swap
SWSWAP find tasks to stop
.TSTOP find stopped timesharing tasks
.SWTTK swap out selected task.

3.21 TSDATA

Data for timesharing scheduler.

3.22 MP45

Memory parity trap handling routines for PDP-11/45:

PARERR memory parity handler
.PARCK null job parity checking.

3.23 MP60

Memory parity trap handling routines for PDP-11/44 and PDP-11/60:

PARERR memory parity handler.
.PARCK null job parity checking.

3.24 MP70

Memory parity trap handling routines for PDP-11/70:

PARERR memory parity handler
.PARCK null job parity checking.

3.25 MPNONE

Memory parity trap handling routines for systems without parity memory:

PARERR memory parity handler
.PARCK null job parity checking.

3.26 NTRACE

Dummy system trace debugging aid.

3.27 TRACE

System trace debugging aid.

3.28 ODT

System debugger.

4.0 DIR MODULE SUMMARIES

4.1 ESPLAS

PLAS subroutines:

.ATRG	attach region (with protection check and RDB)
.ATRG1	attach region (unchecked)
.CADRI	convert ADB address to window identifier
.CKRAC	check access rights to region
.CWIHA	convert window identifier to header address
.ELAW	eliminate address window
.MAPW	map address window
.SRRGN	search GCD for region
.UMAPW	unmap address window
.VRDB	validate RDB
.VWDB	validate WDB
.WNIO	check for I/O in progress through window.

4.2 ESDIR

Directive subroutines:

.CEFNG	check user-specified event flag
.CKTI	check TI indicator for validity
.CRIAD	convert region identifier to ADB address
.FEPPU	get partition, priority and UIC
.RQRSM	request/resume/unstop receiver task
.SNDCK	check DPB size and receiver task name
.SRTPD	search for specified partition
.SRNEW	check if task is on the new task list.

4.3 DMABO

.D.ABO abort task directive.

4.4 DMALP

.D.ALP alter priority directive.

4.5 DMASS

.D.ASS assign LUN directive.

4.6 DMAST

Specify AST service directives:

.D.RAS	specify receive AST
.D.PUT	specify power fail AST
.D.SRA	specify receive-by-reference AST
.D.FET	specify floating point exception AST.

4.7 DMATX

.D.ATX AST service exit.

4.8 DMCMT

.D.CMT cancel mark time directive.

4.9 DMCSR

.D.CSR cancel scheduled requests directive.

4.10 DMDCP

Checkpointing directives:

.D.DCP	disable checkpointing directive
.D.ECP	enable checkpointing directive.

4.11 DMDST

Disable and enable directives:

.D.DST	disable task directive
.D.ENT	enable task directive.

4.12 DMEXT

Exit directives:

.D.EXT	exit directive
.D.EXS	exit with status
.D.EIF	exit if.

4.13 DMFIX

Fix and unfix directives:

.D.FIX fix directive
.D.UNF unfix directive.

4.14 DMGCL

.D.GCL set command line.

4.15 DMGCP

.D.GCP set common block parameters directive.

4.16 DMGCX

.D.GCX set mapping context.

4.17 DMGLI

.D.GLI set LUN information directive.

4.18 DMGMP

.D.GMP set partition parameters directive.

4.19 DMGPP

.D.GPP set task parameters directive.

4.20 DMGSS

.D.GSS set sense switches directive.

4.21 DMGTP

.D.GTP set time parameters directive.

4.22 DMIAR

Inhibit and enable AST recognition directives:

.D.IAR inhibit AST recognition directive
.D.EAR enable AST recognition directive.

4.23 DMMAP

PLAS window directives:

.D.CRW create address window directive
.D.ELW eliminate address window directive
.D.MAP map address window directive
.D.UNM unmap address window directive.

4.24 DMMKT

.D.MKT mark time directive.

4.25 DMPDIR

Privileged directives:

.P.RLR release region directive
.P.GSW get swap space directive
.P.GRD get RDL address directive
.P.PSM claim privileged task semaphore directive
.P.VSM release privileged task semaphore directive
.P.ATK activate tasks directive.

4.26 DMQIO

.D.QIO QIO directive.

4.27 DMREG

PLAS region directives:

.D.CRR create region directive
.D.ATR attach region directive
.D.DTR detach region directive.

4.28 DMREQ

Task request directives:

.D.EXE execute task directive
.D.REQ request task directive
.P.SPW spawn MCR task privileged directive.

4.29 DMRRF

.D.RRF receive by reference directive.

4.30 DMSAR

Send and receive directives:

.D.SEN send data directive
.D.SAR send and request or resume directive
.D.REC receive data directive
.D.ROE receive or exit directive
.D.ROS receive or suspend directive
.D.ROT receive data and stop directive.

4.31 DMSCH

Scheduling directives:

.D.SCH schedule directive
.D.RUN run directive
.D.SYN synchronise directive.

4.32 DMSDV

SST table directives:

.D.SDV specify SST vector table for debugging aid
.D.STV specify SST vector table for task.

4.33 DMSED

Significant event directives:

.D.CEF clear event flag directive
.D.SEF set event flag directive
.D.DSE declare significant event directive
.D.REF read event flag directive
.D.RAF read all event flags directive
.D.WFS wait for single event flag directive
.D.SFS stop for single event flag directive
.D.WFL wait for logical or of event flags directive
.D.SFL stop for logical or of event flags directive
.D.WSE wait for next significant event directive.

4.34 DMSRF

Send by reference directives

.D.SRF send by reference
.D.SRR send by reference and resume or request

4.35 DMSUS

Suspend and resume directives

.D.STP stop directive
.D.SUS suspend directive
.D.RUS resume or unstop directive
.D.RES resume directive
.D.UST unstop directive

4.36 DMXTK

Extend task directive

.D.XTK Extend task directive

5.0 ASX MODULE SUMMARIES

5.1 EM05

ATL scanning:

ASXE1 scan ATL from top
ASXE2 dispatch per active task status.

5.2 TSSHED

Timesharing scheduler:

TSSTS1 entry for dummy ATL node in state TS1
TSSTS2 entry for dummy ATL node in state TS2
MOVUJN promote UJNs to level 1 after completed TTY reads
DEMOTE demote a task one level
TSTAT task accounting, periodic task promotion, periodic scheduling of batch level and statistics gathering
SWPCLT process swap complete event
GETNEW get new jobs
UNSHED remove currently scheduled task from ATL
ATLDSP ATL status dispatch table
ASTDSP sub-status dispatch table after checking for an AST
UJNDSP dispatch table for UJN status settings
UJNDS2 dispatch table for possible task loads
SCHED scheduler scan of UTL
ASTCHK check if AST has been declared for task
SDCON continue task
SDSUP service RSX suspended status
SDSUS suspend task
SDNEW set up new task
CKSIZE calculate swap size
GETATL get and initialize ATL node
SDMRR scheduling service for LRS state
SDSFC scheduling service for SFC state
SDWAT0, SDWAT1, SDWAT2, SDWAT3, SDWAT4 service RSX "waitfor" state
SDIR1 service ATL IR1 status
SDIR4 service ATL IR4 status
SDSWP service ATL status of recoed request succeeded
SDEXIT clean up after task terminates
STDRTN return STD node to pool
SDSTEX terminate task for scheduler
SDETCP inform TCP that task is exiting
SDABT task to be aborted
SDRES runnable task found
SDRUN run task
SDJER, SDAER fatal error found while scheduling
SDQANT calculate task quantum
SWMKRM control task loading/unloading
SWSPCL claim SGAs for task
SWPFAL swap failure
SWLOAD swapping task load complete
CLRACC clear task tick count

GETSWP find swap space for task
RELSWP release swap space
SWSETK set up header for new task
.TMOVE move deque node
.AFAC calculate allocation factor
QUETCP declare event to TCP.

6.0 EXECUTIVE HARDWARE OPERATION

6.1 Processor Priorities

The following processor priority levels are used by IAS:

Level 0: task execution. If a segment fault occurs at this priority, a system trap is caused if the task is set up to service it. Otherwise the task is aborted.

Level 1: the servicing of TRAP-type instructions (i.e. EMT, IOT, TRAP). This includes system directives (EMT 377) and the causing of an SST if any other trap type instruction is detected and the task is set up to service it. Note, however, that a BPT trap executes at priority 1 or the priority of the interrupted process, whichever is the higher.

Level 2: the recognition of 'system events', i.e. significant event declarations, clock ticks, power failure recoveries and scheduler operation. These events are only serviced when returning to task execution, i.e. from an interrupt or a directive.

Level 3: the execution of routines which cannot be interrupted by significant event or clock tick recognition, but can be by peripheral device interrupts. The system trace (debugging routine also runs at this level.

Levels 4 - 7: peripheral interrupt service routines and short uninterruptable sequences.

Note that there are no "software interrupts" between priority levels 0 - 3, and therefore no implied precedence.

6.2 Execution Modes

All tasks, both normal and privileged run in user mode (privileged tasks may access system areas via PAR mapping.) All interrupt and TRAP-type instruction service code runs in kernel mode.

6.3 Register Usage

The system uses only one set of the general registers (R0 - R5), two stack pointers (kernel and user). (There is, of course, only one PC.)

In all processor status words, bit 11 is set. This is ignored by the PDP-11/40, and indicates register set 1 on the PDP-11/45 and PDP-11/70. This is used because although PS bit 11 can be set in kernel or user mode by an RTI or RTT instruction, user mode programs cannot clear it.

6.4 APR Mapping

6.4.1 Kernel APR Mapping -

APR0	000000 - 017777		executive
APR1	020000 - 037777		executive
APR2	040000 - 057777		executive (dynamically mapped to directives or ATL scan)
APR3	060000 - 077777		utility (Map to task headers, Map ISR's for devices, etc.)
APR4	100000 - 117777	)	
APR5	120000 - 137777	)	tables, lists, pool and system
APR6	140000 - 157777	)	routines
APR7	160000 - 177777		external page.

6.4.2 User Privileged Task APR Mapping -

APR0	000000 - 017777		task
APR1	020000 - 037777		handler library (or task)
APR2	040000 - 057777		handler library (or task)
APR3	060000 - 077777		
APR4	100000 - 117777	)	
APR5	120000 - 137777	)	tables, lists, pool and system
APR6	140000 - 157777	)	routines
APR7	160000 - 177777		external page.

By definition, privileged tasks have the same mapping for APR's 4, 5, 6, an 7 as the Executive.

7.0 EXECUTIVE DATA STRUCTURES

7.1 Double-ended Queues (Deques)

Many of the system internal structures take the form of double-ended queues. These have the advantage that the queue can be scanned forwards or backwards and it is easy (i.e. requires few instructions) to insert or remove an entry. The list head consists of two words which are linked to the entries (the first word to the first entry, the second to the last. Thus the first entry's backward pointer and the last entry's forward pointer point to the first word of the list head.)

The first few words of each deque entry are in a standard format:

offset literal	value	function
N.FP	0	forward pointer
N.BP	2	backward pointer
N.AW	4	node accounting word
N.TI	6	terminal PUD address
N.PR (byte)	8	priority.

7.2 Fixed-length Tables

All other structures take the form of fixed length tables. The size of each entry and the number of entries are known at system generation (SYSGEN).

7.3 The System Communication Area (SCOM)

SCOM consists of a number of subroutines, variables, fixed tables and lists, with the remaining space being available for constructing nodes.

Particularly important variables are:

.CRJOB	ATL for current task in scheduler slot
.CRTSK	ATL node for current task
.SERFG	system event recognition flag. (low byte is a set of bit flags. high byte is count of clock ticks to process.)
.SDFLG	scheduler flag word

The particular fields of .SDFLG are

SD.SS	sub-scheduling
SD.LD	a task can be loaded
SD.MR	currently scheduled task in in LRG or MRR state
SD.BG	batch scheduled
SD.JB	timesharing job scheduled
SD.SP	swapping
SD.ND	sub-schedule quantum allocated
SD.RT	swapping out for realtime task

SD.RP	swap for realtime task pending
SD.AA	.ACTK scan aborted
SD.TS	swapping task segment
SD.TL	loading task segment
SD.MT	timesharing mark time due
SD.SW	timesharing swap complete
SD.HP	high priority schedule required
SD.FS	scheduling because nothing active.

7.3.1 Asynchronous System Trap Queue (ASQ) -

The ASQ for each task is a deque consisting of one node for each AST to be executed for the task. The list head is kept in the ATL entry for the task. ASQs are created when an AST occurs and another AST is being processed by the task. The ASTs are fed to the task in the order in which they occur.

7.3.2 Active Task List (ATL) -

The ATL is a priority-ordered deque of ATL nodes. Tasks which have entries in the ATL are either memory resident, a request for their loading has been queued or are stopped (not requiring memory). The ATL entry for a task contains the characteristics of the task.

7.3.3 Clock Queue (CKQ) -

The clock queue is a deque consisting of one node for each operation scheduled to be performed at some future time. A schedule delta-time in the first node (if any) is decremented at each clock tick until the operation becomes due, at which time it is performed. Subsequent nodes contain schedule delta-times relative to the previous node's schedule.

7.3.4 Fixed Task List (FTL) -

This is a deque of nodes for each inactive task that has been fixed in memory. The entries are the same as ATL entries, and when the task is made active, the FTL entry is chained into the ATL.

7.3.5 Global Common Directory (GCD) -

The entries in this deque contain the information required to control SGAs created by INSTALL, and regions created dynamically by the create region directive (CRRG$). It also contains entries for task pure areas.

There are five types of entry:

Dynamically created regions:
these are created by the CRRG$ directive. Initially their contents are undefined. They are subsequently moved to and from the swap file.

Installed libraries:
these are pure, and are thus never written out of memory, merely discarded. They are loaded from the image file from which they were installed.

Pure areas of installed tasks:
as installed libraries, but anonymous. They are created by INSTALL when a task has a pure area.

Installed common areas:
These are swapped to and from the image file on the disk from which they were installed.

Installed regions:
These are initially loaded from the image file from which they were installed. Subsequently they are swapped to the swap file, so that the original task image file is unchanged.

7.3.6 Input-output Request Queue (IRQ) -

Each physical device (all units) has its own IRQ. The IRQ is a priority ordered deque of I/O request nodes with its list head in the header of the handler servicing the device.

I/O request nodes are created primarily by QIO directives, however the executive also creates I/O requests to load task images, record task images (checkpointing) and to run down I/O on an exitted task.

7.3.7 MCR Command Buffer (MCR) -

This deque exists only for compatibility with earlier versions of IAS and RSX-11D. It may be used to pass a command line to a task, although the correct way to do this is via the SPWN$ directive.

7.3.8 Memory Usage List (MUL) -

These deques contain one entry for each allocated segment of memory in a timesharing-type partition. It is primarily used when shuffling memory, so that the occupant of each part of memory can readily be identified.

7.3.9 Physical Unit Directory (PUD) -

This is a fixed list of entries describing each physical device unit in a system. The directory is created by the system configuration routine (SGEN1).

7.3.10 Send/Receive-by-Reference Queue (RRQ) -

The RRQ for a task is a deque containing nodes for each block of data sent (by a send/receive-by-reference directive) to the task whose RRQ list head is in the STD entry for the task.

7.3.11 Swap File List (SFL) -

The SFL is a deque whose entries contain information about the swap files available to the system. It is used by the swap file allocation/deallocation routines in conjunction with the swap file bitmap. It is also used when translating a swap file block number into a PUD address and disk LBN. The entries are in ascending order of swap files.

7.3.12 Send/Receive Queue (SRQ) -

The SRQ for a task is a deque containing nodes for each block of data sent (either by a send or send-and-request directive) to the task whose SRQ list head is in the STD entry for the task.

7.3.13 System Task Directory (STD) -

This is a memory resident directory of all tasks which have been installed into a system. The directory consists of two parts:

(1) a fixed size area (the "alpha table") of one word for each task that may be installed at any time. It takes the form of an alphabetically ordered contiguous list of pointers to STD entries to facilitate search for the STD entry by task name.

(2) The STD entry proper.

Having a task installed enables it to be located quickly without the overhead of going through the MFD and UFD. Instead, a binary chop search is performed on the alpha table, the STD entry located, and the logical block number is used to locate the task image.

7.3.14 Spawn Task List (STL) -

This deque contains one node for each spawned task (i.e. tasks initiated by the SPWN$ directive). In addition, if a command line was issued with the directive, the node contains the command line until it is picked up by the GMCR$ directive.

A spawned task has a pointer in its header to it's STL node, so that there is no need to search the STL to find the relevant node. The purpose of the STL is to let the executive to find all tasks spawned by another task when it exits, so that the linkages can be undone.

7.3.15 Task Partition Directory (TPD) -

This is a fixed list of entries describing each partition in a system (with the exception of the system bootstrap memory). The directory is created by the system configuration routine (SYSGEN) and entries fall into three categories: system-controlled partitions, user-controlled partitions and timesharing partitions.

Each partition has a unique six character partition name.

7.3.16 User Task List (UTL) -

This list is a deque of entries used by the scheduler to find which task to run. It is divided into a number of levels (usually four) which determine the priority of the tasks. Each entry in the deque contains the list head of a deque of Job nodes which belong to that level.

The scheduler can promote or demote tasks between levels on the basis of their activity history by unlinking nodes from one level and relinking them into another. Jobs in the level 1 UTL entry get highest priority service from the scheduler. The maximum number of levels is specified at SYSGEN.

The first three levels are interactive, and the scheduler arranges tasks as follows:

level 1: terminal interactive tasks

level 2: input-output-bound tasks

level 3: compute-bound tasks.

Level 4, if specified at system generation time, is used for batch jobs.

7.3.17 Unibus Mapping Register Request Block (UMR) -

This block is present in each handler that calls the UMR allocation routine.

7.4 The IAS Common Area (IASCOM)

7.4.1 Command Interpreter Table (CIT) -

The CIT contains an entry for each CLI in the system. The maximum number of CLIs which can operate concurrently is determined at SYSGEN.

7.4.2 Device Load Table (DLT) -

The DLT contains one node for each device mounted in the system for timesharing users. It is used by the Timesharing Control Primitives (TCP) for device management.

7.4.3 Device Table (DVT) -

The DVT supplements information contained in the PUD. The contents are information of use to timesharing users.

7.4.4 Job Node Pool (JNP) -

This is a pool of currently unused Job nodes. The number of nodes is specified at SYSGEN.

7.4.5 New Task List -

This list contains the partial ATL nodes for newly created tasks. They are held in this list until the scheduler puts them in the UTL.

7.4.6 Terminal Node Pool (TNP) -

This is a pool of currently unused terminal nodes. The number of nodes is specified at SYSGEN.

7.4.7 User Job Node (UJN) -

A UJN exists for every task under the control of TCP. It contains all the information necessary for resource management of the task. A Job node is obtained from the JNP when a task is initiated and returned when it terminates.

A list head for UJN's belonging to a particular terminal is held in the UTN. The UJN also contains a pointer to the ATL entry for the task if one exists.

7.4.8 User Terminal Node (UTN) -

A UTN is allocated during system start up for every device that is a timesharing terminal. The UTNs serviced by a particular CLI are chained together. The UTN contains the timesharing device characteristics and information about the current state of the terminal. UTNs are obtained from the TNP.

8.0 MEMORY PARTITIONS

Partitions are used to allocate memory for task execution. Partitions are named contiguous areas of physical memory allocated at SYSGEN. There are three types of partitions:

8.1 User-controlled Partitions

This type of partition can only contain one task, SGA or dynamic region at one time. They are intended for the execution of realtime user tasks that are resident for long periods of time.

8.2 System-controlled Partitions

These can contain one or more tasks at the same time. A task can only be loaded into a partition if there a free hole large enough as "shuffling" is not permitted in this type of partition. In general, this type of partition is used when shuffling is impossible because tasks require to be aware of each other's physical location.

8.3 Timesharing Partitions

This is similar to a system-controlled partition except that shuffling is performed if there is not a sufficiently large enough hole. Tasks under the control of the IAS scheduler always reside in timesharing partitions.

9.0 TASK SCHEDULING

All tasks have a priority in the range 1 - 250, a higher priority denoting a more urgent task. Each active task has an ATL entry, ordered in decreasing priority. User tasks run at priorities in the range 2 - 99, timesharing tasks at priority 100. Higher priorities are reserved for realtime and system tasks. The ATL is only scanned when a significant event occurs (i.e. the completion of an input-output request, a task exit, the occurrence of a situation declared explicitly by a task (send data, alter priority, receive by reference or declare significant event), the execution of an illegal instruction, the operation of the IAS scheduler or the processing of a clock tick.

9.1 Checkpointing

If a task can be checkpointed (decided by the setting of the /CP switch in task building), it may be written to the checkpoint file freeing memory for a (higher priority) realtime task. This process is purely priority driven.

9.2 Swapping

Swapping is controlled by the IAS processor, and unlike checkpointing, is not priority driven.

9.3 Timesharing Control Primitives

An IAS system may or may not have TCP. TCP is a privilesed task (named PI....) that runs at priority 221. It is a pseudo-device handler and is communicated with by QIO directives (or more indirectly by TCS macros.) TCP communicates with system services through SCOM and IASCOM. However, certain system services (such as the terminal handler) also issue QIOs to it.

Jobs running under TCP have UJNs. The IAS scheduler knows whether there is a UJN because the A.JN field contains its address, or zero if there is not one. Further consideration of the operation of the IAS scheduler will assume that TCP is not present.

9.4 ATL Scanning And The IAS Scheduler

ATL scanning is performed by two routines ASXE1 and ASXE2 in module EM05. ASXE1 scans the ATL downward when it is called by 'common return to interrupted program', or from ATL scan task status service, whenever a significant event declaration is to be effected. ASXE2 is called from the common exit for directives (EMT 377) and either returns control to the task or scans through lower priority tasks.

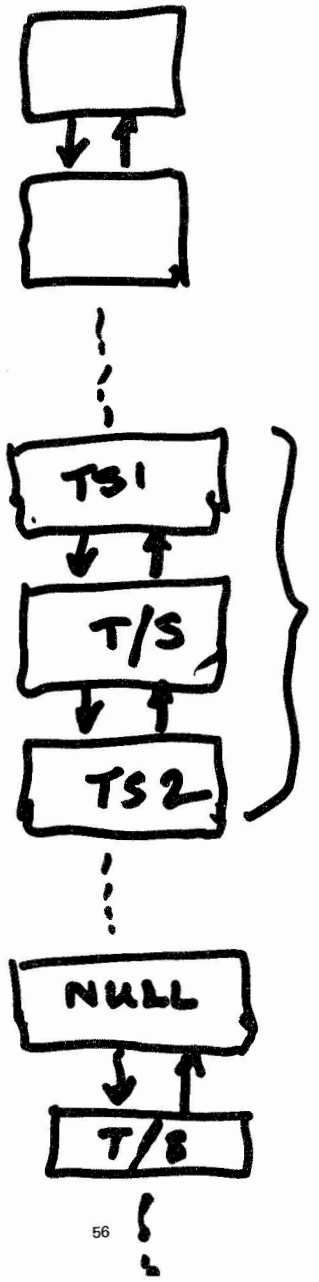
When the timesharing scheduler is installed, there are two special nodes in the ATL, distinguished by having the task status byte (A.TS) set to TS1 or TS2. ASEX2 uses the task status to jump through a table ASXDT (set up by the TS macro in EM00) to TSSTS1 or TSSTS2 respectively, in TSSHED. The nodes with statuses TS1 and TS2 bracket the node corresponding to the timesharing task currently selected to run by the scheduler. All other runnable timesharing jobs are placed in the ATL at positions corresponding to priority 1, but above the null job, which also has priority 1.

9.4.1 Task States -

The followings are task states according to the task status byte A.TS in the ATL node:

LRP load request pending: memory has been found and allocated for the task. The I/O request(s) to load the task root segment is now queued.

LRQ load request queued: the I/O request(s) to load the task have been queued and may be in progress.



LRS load request succeeded: all I/O load requests have been successfully completed.

LRF load request failed: one of the I/O load requests completed unsuccessfully.

LRG waiting for region to load: the task is waiting for a region to be loaded.

RUN runnable: the task is runnable but not necessarily executing.

AST AST queued: an AST has been queued for the task.

RLA reloading for AST checking: an AST has occurred for the task when it is checkpointed or swapped out. It's header must be examined to see if the task is already processing an AST.

SUS suspended.

WND suspended waiting for nodes: the task is in a wait state until sufficient nodes are available to complete a directive.

WSM waiting for privileged task semaphore: the task has attempted to claim a semaphore which is already claimed.

STP stopped for STOP$ directive.

ST0 - ST4 stoped for event flags: the respective ranges are 1 - 16, 17 - 32, 33 - 48, 49 - 64 and 1 - 64.

WF0 - WF4 waiting for event flags: the ranges are as above.

EXT exiting/exited.

IR1 I/O rundown is to be started.

IR2 I/O rundown in is in progress on a unit.

IR3 I/O rundown is complete on a unit.

IR4 I/O rundown is complete (successful or otherwise.)

TFF terminated for execution fault.

TNR termination notice is requested.

STN suspended for termination notice: the special task .TKTN. is run to provide the termination notice.

SFC suspended for checkpointing: the task is being swapped out of memory.

RRQ record request queued: the I/O request to write out the task has been queued.

RRS record request succeeded: the I/O request to write out the task has been completed successfully.

RRF record request failed: the I/O request to write out the task has failed.

PAR parity error: a memory parity error has occurred and .TKTN. is not in the system. The task is indefinitely suspended.

TSE timesharing task has exited: the scheduler can now clean up and use the ATL node to return exit status to the task's requestor.

TS1 timesharing scheduler special node 1.

TS2 timesharing scheduler special node 2.

MRL waiting for memory.

MRE waiting for memory for EXEC$. If none can be allocated at the first attempt, the task will not run.

MRR waiting for memory for region.

WDI waiting for directive to complete: the task issued a directive resulting in another task being requested, and the request operation is not yet completed.

DIF directive failed: the task was in state WDI and the directive failed.

IDL idle: special state used for the null task.

WAC waiting for accounts write: the task is built with RSX-11D style accounting and the system is running with accounting included. The task has exited and is waiting for the accounting information to be written. The accounts logger ACCLOG will set status to EX1.

EX1 exit complete after accounting.

MEX marked for extension.

In addition, timesharing tasks have a status byte (A.TST) accorded as follows:

RUN runnable.

RSD to be suspended.

SUS suspended.

ABT to be aborted.

NEW new to scheduler.

EXT exited (not yet processed by TCP).

LOD to be loaded.

CON to be continued.

NW2 new after install.

EXX exiting, TCP QIO pending.

FIN exited and processsed by TCP (UJN released).

9.4.2 Starting A Task -

A task may be started in a number of ways. For the purposes of this description it is assumed that a CLI has issued a RUN directive and that the task is to be a timesharing task on a multi-user system, i.e. without TCP. The process is different for non-timesharing and TCP-controlled task.

The notation xxxxxx/yyyyyy means routine xxxxxx in module yyyyyy.

1. DIRECT/EM10 dispatches to .D.RUN/DMSCH.

2. .D.RUN calls TE.COM/DMSCH to scan the STD and check the partition parameters. It calls .PPOOL/ESMISC to pick a node from the pool, setting the ATL address entry to zero, and .CKINS/ESMISC to enter it in the clock queue (it will be at the front as the delta time interval is zero.)

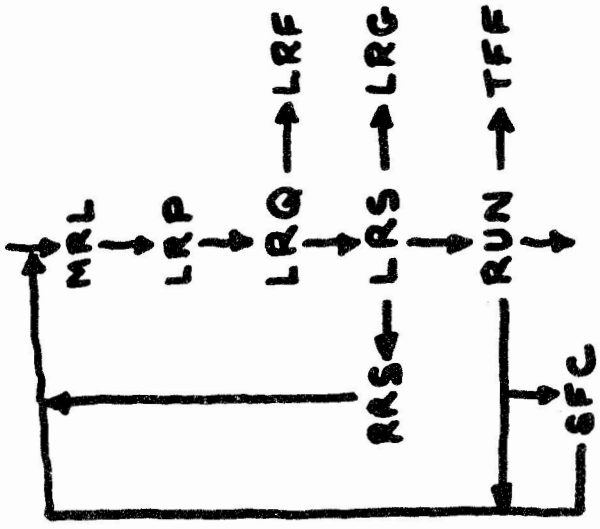
3. When a clock tick occurs, CTREC/EM06 is called to recognise it. In the process of examining the clock queue, it calls .REQS/ESREQS which sets up an ATL node with task status MRL (waiting for memory), adds it to the new task list and declares a significant event.

4. When ATL scanning occurs, because of the significant event, TSSTS2/TSSHED is eventually reached (after TSSTS1). It calls GETNEW/TSSHED to get new jobs from the new task list and insert them in the UTL at the level appropriate to each task.

5. When scheduling is required, SCHED/TSSHED scans the UTL. Table UJNDSP is used indexed by the timesharing task status byte. If the value is JS.NEW, indicating a new task, control is passed to SDNEW/TSSHED. This calls the routine GETATL/TSSHED which links the ATL node into the ATL after the null job. The timesharing status byte in the ATL is set to LOD (to be loaded). SDNEW also calls GETSWP/TSSHED to check for swap space. Control now passes to SDRES/TSSHED (scheduling runnable task found) and results in attempts to find store for the task, and if successful, the process of loading the task is started. This results in the process passing through the states LRP, LRQ, LRS, LRG (and possibly LRF).

   When a load request succeeds, the ATL scan passes control to ASXLRS/EM05 where the header is checked, the APRs are set up, LUN

STARTING A TASK



assignments set up etc. Finally, the RUN state is entered, and the ATL is rescanned from the top.

9.4.3 Running A Task -

When a task reaches RUN status, it will execute if it is the highest priority task in RUN state in the ATL until its time quantum expires or some internal or external event occurs to switch it into another state. These states are described below:

1. A task can get checkpointed or swapped out even if it is potentially runnable. In this case it will go into into state SFC, but when checkpointing is complete, the state will still be RUN so that the ATL scan can reload it if there is enough free memory.

2. There are a number of states, SUS, WND, WSM, STP, ST0 - ST4, WF0 - WF4 and MEX, where a task cannot run until some action occurs, causing it to go back into the RUN state. In these states, the task can, of course, be checkpointed or swapped.

3. When an AST occurs for a task, it can be in memory, in which case an AST node is linked onto its AST queue. If it is not in memory, it passes into state RLA, as the task must be reloaded to get the AST queue head. (N.B. It would be desirable to have the AST queue head in the ATL node, but this would mean making such nodes 8 words longer.)

9.4.4 Exiting A Task -

When a task is to exit (either voluntarily, forced or because of a load request failure), it goes through the following states:

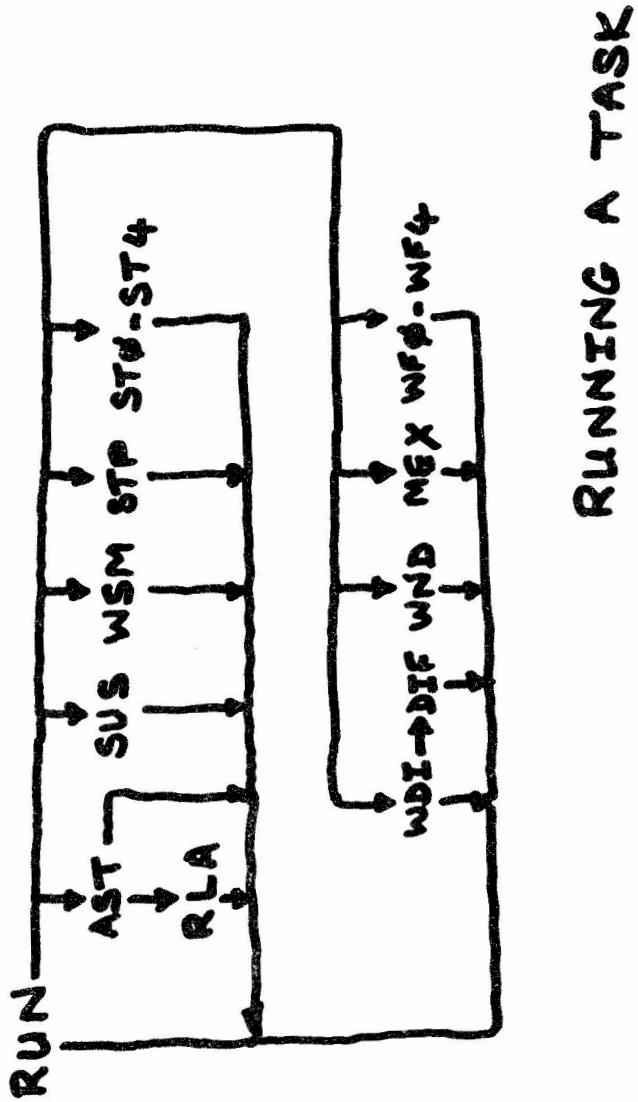
1. State EXT is entered; ASXEXT/EM05 clears any privileged task semaphores in use and looks to see if there are any I/O requests pending. If so, state IR1 is entered. Otherwise, if a termination notice is required, state TNR is entered. If not, store is freed, various nodes flushed from queues (e.g. ASTs), the command line node, if any, is deallocated, and if the task has been spawned, its requestor is notified, and the ATL node returned to the pool.

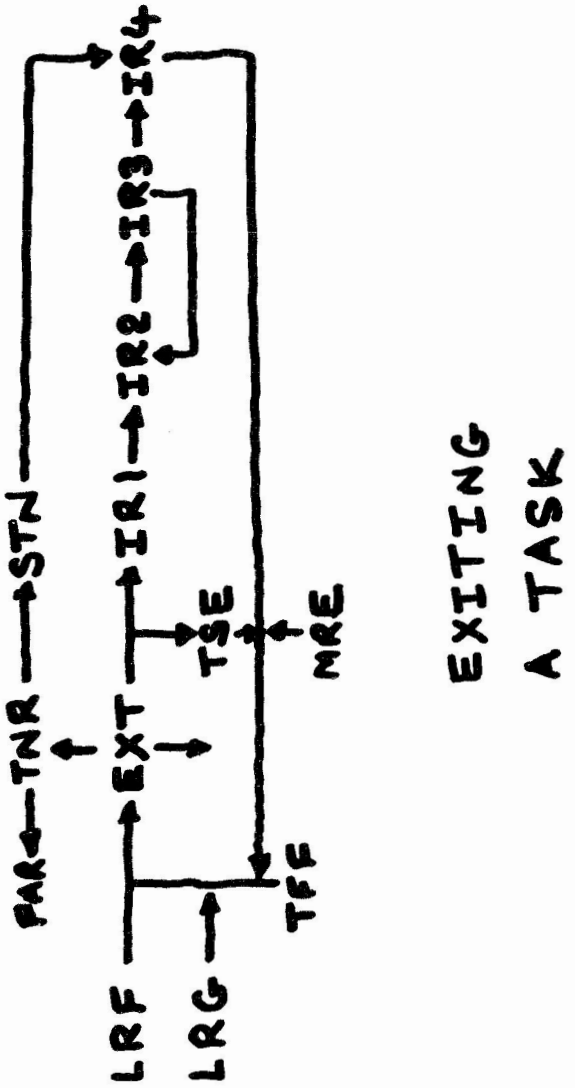
2. State IR1 is processed by ASXIR1/EM05. It queues the first I/O rundown request and proceeds to state IR2.

3. State IR2 results in no action by the ATL scan.

4. State IR3 is processed by ASXIR3/EM05. If there are any requests to be queued on another unit, this is done and state IR2 is re-entered. Otherwise, state IR4 is entered.

5. State IR4 serves as a trap for a failed I/O rundown request and is also entered from IR3. A transition is made back to state EXT.





EXITING A TASK

6. State TNR is entered from state EXT if a termination notice is required. The STD is scanned for the task .TKTN. If it is not found, state IR4 is entered unless there was a parity error, in which case state PAR is entered. If .TKTN. is installed, a check is made to see if the task exiting is .TKTN. If not, the task is set in state STN and .TKTN. requested by a call to .REQS/ESREQS. Otherwise, the ATL is searched for tasks in state STN, and they are set in state IR4.

END of [DEMOR.IASJIAS.MEM
REV: 0

PDS>

DECUS

DIGITAL EQUIPMENT COMPUTER USERS SOCIETY
ONE IRON WAY, MRO2-1/C11
MARLBORO, MASSACHUSETTS 01752

BULK RATE
U.S. POSTAGE
PAID
PERMIT NO. 129
NORTHBORO, MA
01532

ASSOCIATE
[redacted]

MOVING OR REPLACING A DELEGATE?

Please notify us immediately to guarantee continuing receipt of DECUS literature. Allow up to six weeks for change to take effect.

( ) Change of Address
( ) Delegate Replacement

DECUS Membership No.: _____

Name: _____

Company: _____

Address: _____

_____

State/Country: _____

Zip/Postal Code: _____

Mail to: DECUS - ATT: Membership
One Iron Way, MRO2-1/C11
Marlboro, Massachusetts 01752 USA

Affix mailing label here. If label is not available, print old address here. Include name of installation, company, university, etc.