

```

Status : NOT (Ready);
DO WHILE (Status <> Ready);
    CALL OptTable_DisksReady (First$D$Num, Qty, Status$Ptr);
    IF (Status <> Ready)
        THEN
            DO;
                /* Allow other tasks to continue */
                CALL RQ$SEND$CONTROL (@Exception); /* Release OptTable Resource */
                CALL RQ$SLEEP (Time$Len, @Exception);
                CALL RQ$RECEIVE$CONTROL (OptTable$Region, @Exception);
            END;
            END; /* While - Files are open for use */
        END;
END MarksOpen$And$Wait;

/*PROCEDURE */ MarksAll$Open: Procedure;
/* Mark all Disks of the application as open - Files are already open */

/*BEGIN*/
IF ( NOT(Rqds$Appl.All$Reqd) )
    THEN Status : Cannot$Load$All;
ELSE
    DO;
        Qty : Rqds$Appl.Qty; /* Set the output for the number of disks */
        CALL OptTable_Search$Disk (Rqds$Disk, First$D$Num$Ptr, Status$Ptr);
        CALL MarksOpen$And$Wait (Qty);
    END;
    CALL RQ$SEND$CONTROL (@Exception); /* Release OptTable Resource */
END MarksAll$Open;

/*PROCEDURE */ MarksSingle$Open: Procedure;

/* Mark Req'd Disk Open in the OptTable - Attach first & open file if necessary */

/*BEGIN*/
/* Set the output for the number of disks */
Qty : 1;
/* Ensure that the disk does exist */
IF ( (Rqds$Disk - Rqds$Appl) > Rqds$Appl.Qty)
    THEN
        DO;
            Status : Disk$Not$Exist;
            CALL RQ$SEND$CONTROL (@Exception); /* Release OptTable */
        END;
    ELSE /* Disk does exist */
        DO;
            CALL OptTable_Search$Disk (Rqds$Disk, First$D$Num$Ptr, Status$Ptr);
            IF (Status : Found)
                THEN /* Disk has already been used */
                    DO;
                        CALL MarksOpen$And$Wait (1);
                        CALL RQ$SEND$CONTROL (@Exception); /* Release OptTable Resource */
                    END;
                ELSE /* Disk has not yet been used */
                    DO;
                        /* Create Entry in OptTable Resource & Mark it as open */

```

```

CALL OptTable_AttachDisks (Rqst$Disk, 1, Opt$App$Num, First$D$NumPtr, Status$Ptr);
IF (Status = Successful)
  THEN
    DO;
      CALL OptTable_MarkOpen (First$D$Num, 1, Read$Write, Status$Ptr);
      IF (Status <> Successful)
        THEN /* Restore OptTable */
          CALL OptTable_RemoveDisks (First$D$Num, 1);
    END;
  CALL RG$SEND$CONTROL (@Exception); /* Release OptTable Resource */

  /* Open the files on the Hard Disk */
  IF (Status = Successful)
    THEN
      DO;
        /* Attach & Open files on the Hard Disk */
        CALL RG$RECEIVE$CONTROL (HD$Attach$Region, @Exception);
        CALL HD$Attach_Open$Create$File ((Rqst$Disk MOD 16), Req$App$App$Connection,
          Read$Write, @Opt$Disk$Map(First$D$Num), Open$Connection, @Exception);
        IF (Exception <> $NOX)
          THEN Status = Disk$Attach$Error;
        CALL RG$SEND$CONTROL (@Exception); /* Release HD$Attach Resource */

        /* If an error occurred then restore the OptTable Resource */
        IF (Status <> Successful)
          THEN
            DO;
              CALL RG$RECEIVE$CONTROL (OptTable$Region, @Exception);
              CALL OptTable_MarkClosed (First$D$Num);
              CALL RG$SEND$CONTROL (@Exception);
            END;
          END;
        END; /* Disk Entry Created */
      END; /* Disk required is a valid disk */
    END Mark$Single$Open;

/*PROCEDURE */ Get$News$App: Procedure;

/* Reads the application's parameters from the Common$Info Resource, Determines
  access & obtains a token for the application's directory file */

/*BEGIN*/
  CALL Common$Info Read (Rqst$App, Status$Ptr);
  IF (Status = Successful) THEN
    IF ( ( (Rqst$Disk - Rqst$App) > Req$App$Qty) AND (NOT Whole$App$Rqst) )
      THEN Status = Disk$Not$Exist;
    ELSE
      DO;
        CALL Query$Access (@News$App$Access, News$App$Load$Type, News$App$OS$Num);
        IF (Status = Granted)
          THEN /* Can attach the Applications Directory */
            DO;
              CALL Common$Info_Attach (Status$Ptr);
              IF (Status <> Successful)
                THEN CALL RG$DELETE$CONNECTION (News$App$App$Connection,
                  Not$Mailbox, @Exception);
            END;
          END;
        END; /* Access determined & directory attached */

```

END GetNewsAppl:

/*PROCEDURE */ LoadAllDisks: Procedure;

/* Each disk of the new application is attached to the OptTable. The required disk is marked as open & the files for the disks are opened */

/*BEGIN*/

/* Set the output for the number of disks */

IF (WholeApp=Rqst)

THEN Qty = ReqdsAppl.Qty;

ELSE Qty = 1;

CALL OptTable_AttachDisks (RqstDisk, ReqdsAppl.Qty, OptAppNum, FirstDiskNumPtr, StatusPtr);

IF (Status = Successful)

THEN

DO;

/* Mark all disks of the application or just the required disk as open */

IF (WholeApp=Rqst)

THEN CALL OptTable_MarkOpen (FirstDiskNum, ReqdsAppl.Qty, ReadWrite, StatusPtr);

ELSE CALL OptTable_MarkOpen ((FirstDiskNum + (RqstDisk MOD 16) - 1), 1, ReadWrite, StatusPtr);

IF (Status <> Successful)

THEN /* Restore OptTable */

CALL OptTable_removeDisks (FirstDiskNum, ReqdsAppl.Qty);

ELSE

DO;

/* Open or create files on the Hard Disk */

CALL RqstSEND\$CONTROL (@Exception); /* Release OptTable Resource */

CALL RqstSEND\$CONTROL (@Exception); /* Release CommonsInfo Resource */

CALL RqstRECEIVE\$CONTROL (HD\$Attach\$Region, @Exception);

DO Count = 0 TO (ReqdsAppl.Qty - 1);

CALL HD\$Attach_Open\$Create\$File ((RqstDisk + Count), ReqdsAppl.App\$Connection, ReadWrite, @OptDisk\$Map(FirstDiskNum + Count).Open\$Connection, @Exception);

END; /* Do Loop */

CALL RqstSEND\$CONTROL (@Exception); /* Release HD\$Attach Resource */

END;

END; /* Else */

/* Status <> Successful

THEN /* Restore OptTable */

CALL OptTable_RemoveAppl (OptAppNum);

END LoadAllDisks;

/*PROCEDURE */ LoadSingleDisk: Procedure;

/* Attach the disk to the OptTable (create an entry), mark as open & open the file associated with the disk */

/*BEGIN*/

/* Set the output for the number of disks */

Qty = 1;

CALL OptTable_AttachDisks (RqstDisk, 1, OptAppNum, FirstDiskNumPtr, StatusPtr);

IF (Status = Successful)

THEN

DO;

```

/* Mark disk entry as open */
CALL Op$Table_Mark$Open (First$D$Num, 1, Read$Write, Status$Ptr);
IF (Status <> Successful)
  THEN /* Restore Op$Table */
    CALL Op$Table_Remove$Disks (First$D$Num, 1);
  ELSE
    DO;
      /* Open or create file on the Hard Disk */
      CALL RGS$SEND$CONTROL (@Exception); /* Release Op$Table Resource */
      CALL RGS$END$CONTROL (@Exception); /* Release Common$Info Resource */
      CALL RGS$RECEIVE$CONTROL (HD$Attach$Region, @Exception);
      CALL HD$Attach_Open$Create$File ( (Rqst$Disk MOD 16), Req$App$App$Connection,
        Read$Write, @Op$Disk$Map(First$D$Num).Open$Connection, @Exception);
      CALL RGS$SEN$CONTROL (@Exception); /* Release HD$Attach Resource */
    END;
  END; /* Then */
IF (Status <> Successful)
  THEN /* Restore Op$Table */
    CALL Op$Table_Remove$App$ (Op$App$Num);
  END Load$Single$Disk;

```

```

/* BEGIN Common$Disks$Open */

```

```

/* If Rqst$Disk is an application number then all disks of the application are required,
   else only the single disk requested is required */

```

```

Rqst$App$ : (Rqst$Disk AND 0Fh); /* Drop the last 4 bits */

```

```

IF (Rqst$App$ = Word$App$)

```

```

  THEN Status = App$Not$Exist; /* Nothing further will be attempted */

```

```

  ELSE

```

```

DO; /* Rqst$App$ <> Word$App$ */

```

```

IF (Rqst$Disk = Rqst$App$)

```

```

  THEN

```

```

    DO;

```

```

      Whole$App$Rqst = True;

```

```

      Rqst$Disk = Rqst$App$ + 1; /* First Disk of App */

```

```

    END;

```

```

RESTART;

```

```

/* Is the Application already in the Op$Table */

```

```

CALL RGS$RECEIVE$CONTROL (Op$Table$Region, @Exception);

```

```

CALL Op$Table_Search$App$ ( Rqst$App$, @Op$App$Num, Status$Ptr);

```

```

IF (Status = Found)

```

```

  THEN /* Case Where Application is already in use, and some files available. */

```

```

    DO;

```

```

      /* Set up pointer to the relevant application entry */

```

```

      Req$Op$App$Ptr = @Op$App$Map(Op$App$Num);

```

```

      CALL Query$Access (@Req$App$.Access, Req$App$.Load$Type, Req$App$.QS$Num);

```

```

      IF (Status <> Granted)

```

```

        THEN CALL RGS$SEND$CONTROL (@Exception); /* Release Op$Table Resource */

```

```

      ELSE IF (Whole$App$Rqst)

```

```

        THEN CALL Mark$All$Open;

```

```

        ELSE CALL Mark$Single$Open;

```

```

      END; /* Case where Application is already available */

```

```

ELSE /* Application is not yet in use - Must load */
DO;
CALL RM$SEND$CONTROL (@$Exception); /* Release OptTable Resource */
CALL RM$RECEIVE$CONTROL (Common$Info$Region, @$Exception);
CALL Get$New$Appl;
IF (Status <> Successful)
THEN CALL RM$SEND$CONTROL (@$Exception); /* Release Common$Info Resource */

ELSE /* Continue as long as successful */
DO;
CALL RM$RECEIVE$CONTROL (OptTable$Region, @$Exception);

/* Ensure that while the OptTable resource was released, that another
task did not attach the application. If this occurred, then the
procedure must be restarted since control is in the wrong option */

CALL OptTable_Search$Appl (Req$Appl, @$Appl$Num, Status$Ptr);
IF (Status = Found)
THEN
DO;
CALL RM$DELETE$CONNECTION (New$Appl.App$Connection, No$Mailbox, @$Exception);
CALL RM$SEND$CONTROL (@$Exception); /* Release OptTable Resource */
CALL RM$SEND$CONTROL (@$Exception); /* Release Common$Info Resource */
GOTO RESTART;
END; /* Restart from outermost level */

/* ELSE the above did not occur - Continue & Attach the Application to the OptTable */

CALL OptTable_Attach$Appl (Req$Appl, @$Appl$Num, Status$Ptr);
IF (Status <> Successful)
THEN /* No more Application entries are available in the OptTable */
CALL RM$DELETE$CONNECTION (Req$Appl.App$Connection, No$Mailbox, @$Exception);
ELSE
DO;
Req$Opt$Appl$Ptr = @$Appl$Map(Opt$Appl$Num);
IF (Req$Appl.All$Reqd Ok Who$Appl$Reqst)
THEN CALL Load$All$Disk;
ELSE CALL Load$Single$Disk;
END; /* Else */

/* If errors have occurred then resources are still being accessed */
IF (Status <> Successful)
THEN
DO;
CALL RM$SEND$CONTROL (@$Exception); /* Release OptTable Resource */
CALL RM$SEND$CONTROL (@$Exception); /* Release Common$Info Resource */
END;
END; /* Else Continue */

END; /* Case where Application is not immediately available & has to be loaded */

END; /* Req$Appl <> No$Appl */

END Common$Disk$Open;

/* END COMMON$DISK$OPEN PROCESS */

/* ----- */

```

REFERENCES

1. De Souza, E.A. *EDULAN: A Local Area Network for an Educational Environment*, MSc Dissertation, University of the Witwatersrand, Johannesburg, S.A., 1987.
2. Schutz, R.A. *An Investigation into the Automatic Analysis of Pascal Programs*, MSc Project Report, University of the Witwatersrand, Johannesburg, S.A., 1987.
3. Svobodova, L. "File Servers for Network-based Distributed Systems", *ACM Computing Surveys*, Vol 16, No.4, Dec 1984, pp353-398.
4. Hurst, M. "Sharing data between Microcomputers on a local Net", *Data Communications*, Sept 1984, pp139-148.
5. Miller S. and Collins M. "Toward a Reference Model of Mass Storage Systems", *IEEE Computer*, July 1985, pp9-22.
6. Birrell A.D. and Needham R.M. "A Universal File Server", *IEEE Trans. on Software Engineering*, Vol SE-6, No.5, Sept 1980, pp450-453.
7. Hudson, C. "A Microprocessor File Server", *Software and Microsystems*, Vol 2, No.1, Feb 1983, pp9-13.
8. Rolander T., Baird R. and Wharton J. "Network Software borrows design from Microcomputer Operating System", *Data Communications*, Dec 1982, pp123-133.
9. Kolod, M. "IBM PC Disk Performance and the Interleave Factor", *Byte*, Fall 1985, pp283-290.
10. Shiell J. and Markoff J. "Adding a Hard Disk to your PC AT", *Byte*, Fall 1985, pp159-172.
11. Calingaert, P. *Operating System Elements: A user Perspective*, Prentice-Hall, 1982.
12. Walker, B.J. *Computer Security and Protection Measures*, Dawden, Hutchinson and Ross Inc., Pennsylvania, 1977.

References

13. Haskett, J.A. "Pass-Algorithms: A user validation scheme based on knowledge of secret algorithms", *Communications of the ACM*, Vol 27, No. 8, Aug 1984.
14. Jones, R. "Disc Cache for PCs", *Data Processing*, Vol 28, No.2, March 1986, pp79-81.
15. McKeon, B. "An Algorithm for Disk Caching with Limited Memory", *Byte*, Sept 1985, ppi29-138.
16. Walker, A.J. *Structured Information Processing System Design (Rev 4.0)*, Dept. of Elec. Eng., University of the Witwatersrand, Johannesburg, S.A., 1986.
17. Booch, G. "Object-Oriented Development", *IEEE Transactions on Software Engineering*, Vol SE-12, No. 2, Feb 1986, pp211-221.
18. Bishop, J.M. *Data Abstraction in Programming Languages*, Computer Science Dept., University of the Witwatersrand, Johannesburg, S.A., 1984.
19. Tucker, S. "iRMX86 - A Realtime operating system for advanced 16-bit Microcomputers", *Microprocessors and Microsystems*, Vol 7, No. 8, Oct 1983.
20. *Introduction to the iRMX 86 Operating System*, Intel Corporation, Santa Clara, California, 1982.
21. *iRMX 86 Nucleus Reference Manual*, Intel Corporation, Santa Clara, California, 1982.
22. *PL/M-86 User's Guide*, Intel Corporation, Santa Clara, California, 1982.
23. Voydock V. and Kent S., "Security Mechanisms in High-Level Network Protocols", *ACM Computing Surveys*, Vol 15, No. 2, 1983, pp135-171.
24. Voydock V. and Kent S., "Security in High-Level Network Protocols", *IEEE Communications Magazine*, Vol 23, No. 7, July 1985, ppi2-24.

References

25. Fidlow D. and Waterhouse P. "A Comprehensive approach to Network Security", *Data Communications*, April 1985, pp195-213.
26. *LAN Software Report*, Novell Inc., Provo, Utah, USA, 1985.
27. *LAN Operating System Report*, Novell Inc, Provo, Utah, USA, 1986.

Author Hildyard Mark William

Name of thesis Design Of A Disk-based File Server Involving The Mapping Of "pseudo-disks". 1989

PUBLISHER:

University of the Witwatersrand, Johannesburg

©2013

LEGAL NOTICES:

Copyright Notice: All materials on the University of the Witwatersrand, Johannesburg Library website are protected by South African copyright law and may not be distributed, transmitted, displayed, or otherwise published in any format, without the prior written permission of the copyright owner.

Disclaimer and Terms of Use: Provided that you maintain all copyright and other notices contained therein, you may download material (one machine readable copy and one print copy per page) for your personal and/or educational non-commercial use only.

The University of the Witwatersrand, Johannesburg, is not responsible for any errors or omissions and excludes any and all liability for any errors in or omissions from the information on the Library website.