AAEC/E458

AUSTRALIAN ATOMIC ENERGY COMMISSION

RESEARCH ESTABLISHMENT

LUCAS HEIGHTS

AN ARCHIVAL AND AUTOMATIC RETRIEVAL SYSTEM
FOR IBM360 DISK DATA SETS

by

G.W. COX
D.J. RICHARDSON

October 1978

AUSTRALIAN ATOMIC ENERGY COMMISSION
RESEARCH ESTABLISHMENT
LUCAS HEIGHTS

AN ARCHIVAL AND AUTOMATIC RETRIEVAL SYSTEM FOR IBM360 DISK DATA SETS

by

G.W. COX
D.J. RICHARDSON

ABSTRACT

A system is described which allows infrequently used disk data sets to be archived on magnetic tape and restored to disk automatically when required.

The system currently operates on an IBM360 computer under Release 21.7 of operating system MVT.

CONTENTS

1. INTRODUCTION

Modern computer operating systems allow much flexibility in the storage and retrieval of data. The choice of storage medium depends on many factors : user preference, the volume of data to be processed, the mode of data access, and the dictates of equipment and operating system environment. A small deck of FORTRAN cards may satisfy the occasional computer user, whilst magnetic tape may be essential for the user with large reference banks of data. Disk or other similar direct access storage must be available for real-time, interactive data access.

Direct access storage limitations present all computer installations with the problem of deciding what data shall occupy prime on-line direct access storage space, and how long such data shall be allowed to remain there. Various schemes have been devised to allow the transfer of infrequently used direct access data to archival storage such as magnetic tape; but these schemes are usually cumbersome and slow, and require special operator assistance to retrieve data.

A direct access housekeeping system has been developed by one of us [Cox 1978]. This has enabled on-line disk storage space to be used much more effectively by allowing the migration and tidying, at theoretical maximum speed, of data stored on groups of disks according to such installation-determined parameters as frequency of data use and amount of unused storage space available on each disk. This system has been used as a basis for the automatic data archiving and retrieval scheme currently in use on an IBM360/65 computer under Release 21.7 of IBM's MVT (Multiprogramming with a Variable number of Tasks) operating system.

2. DESIGN OF THE ARCHIVAL AND RETRIEVAL SYSTEM

The utility AEMOVE [Cox 1978] is currently used at the Australian Atomic Energy Commission's Research Establishment to maintain disk data sets and to shift infrequently used data sets to magnetic tape. All direct access user data are accessed through the system catalog, their precise locations being unknown to the user.

To design an archival and retrieval system for disk data sets, a means is needed to distinguish between user data normally resident on magnetic tape, and data that has been archived on magnetic tape and which needs restoration to disk before use. Magnetic tape volume serial names beginning with # are chosen for the storage of archived data sets, and a reference to a data set is deemed to be a reference to an archived data set if

- the data set is referred to through the system catalog,
- the data set is catalogued on a magnetic tape volume, and
- the catalogued volume serial name or names begin with #.

Explicit references to data stored on magnetic tape volumes, including # volumes, proceed normally. The archiving process itself uses such explicit references.

In the design of the retrieval component of the system, it was required that

- data set retrieval be 'transparent' to the computer user,
- no special operator action be necessary, and
- modifications to the standard computer operating system be minimal.

Transparency to computer users implies that retrieval of data sets shall not in any way affect their jobs or job-step specifications. The operating system must, however, allocate at least one magnetic tape device for retrieval use. Use of an operating system (OS) internally generated dummy job and job-step appeared to us to be a way out of this allocation dilemma, as the IBM360 MVT OS already uses this concept to allow the mounting and use of disk packs for subsidiary catalog searching during device allocation. The disk drives allocated for this purpose are, moreover, released for further allocation immediately the catalog search has been completed. Investigation showed that magnetic tape drives could similarly be allocated for system use in dummy jobs and job-steps during normal job and job-step device allocation, and this approach was used in devising the data set retrieval system. The requirement that no special operator action be needed was thus satisfied, as normal device mount messages would be issued to the operator to mount the necessary magnetic tapes.

Modifications to MVT components were confined to the device allocation and allocation interface modules. Additional modules were written to retrieve the actual data set.

3. IMPLEMENTATION OF THE ARCHIVAL AND RETRIEVAL SYSTEM

During the device allocation phase of each job-step, the module IEFW21SD issues a LOCATE macro instruction for every data set whose job control language location is not explicitly given, to ascertain through the system catalog the volume serial name and device type on which the data set resides. The actual existence of the data set on the located

volume is not relevant at this time. The return code from LOCATE is used to decide what further action is required. This depends on whether the system catalog contains a reference, or possible indirect reference, to the required data set. A return code of 4 indicates that the mounting of another control volume is required to continue the catalog search. The catalog specifies the required volume and device type. The operating system then sets up a dummy job and job-step to arrange and wait for the mounting of the specified control volume. When this has occurred, the module IEFW21SD is recalled and the same LOCATE macro instruction re-issued. This recursive procedure is continued until the required data set catalog entry is either located or found to be absent.

A modification is included immediately after the LOCATE macro instruction to test for successfully located entries catalogued on magnetic tape volumes whose serial names begin with #. If so, a new volume serial name, #INITX is synthesised in core, where X denotes the EBCDIC value of the job-step initiator's key. This synthetic name is assigned to ensure the later allocation of one and only one magnetic tape volume for each concurrent task, independent of the number of data set retrievals needed for each such task. The LOCATE return code is then changed to 4 to force a mount request - without waiting - for this #INITX control volume. The generality of the allocation modules allows this to be done, as no check is made that the requested control volume is actually a direct access device.

As there is no system wait for the actual mounting of allocated magnetic tape volumes, allocation attempts to proceed to the immediate winding up of the dummy job and job-step set up for the mounting of the control volume. This would normally involve the use of the de-allocation module IEFCVOL2. Modifications have been made to rename this module AEFCVOL2 and to insert a new module IEFCVOL2 for the saving of the necessary catalog information and for determining the direct access devices needed for the receipt of the recovered archived data sets. Control is then passed from the new IEFCVOL2 to AEFCVOL2. The task completion code word in the initiator's task control block is used to accumulate a chain of data set recovery information blocks, should one or more archived data set recoveries be needed in any one job-step.

For later receipt of each retrieved data set the inserted module IEFCVOL2 chooses a direct access device which

- is of type STORAGE,
- does not already contain the desired data set, and
- has the greatest free space.

If no device is able to satisfy these requirements, the job-fail bit is set.

Before passing control to IEFCVOL2, if a direct access device has been selected, the data set to be restored is re-catalogued as a multi-volume, mixed device type data set by inserting as the first catalogued volume, the selected direct access device serial name and type, followed by the original # tape catalog entry. This enables a later re-catalog to a single direct access device if retrieval is successful, or if unsuccessful the restoration of the original magnetic tape catalog entry. Mixed device type catalog entries may be used in this context, as the operating system makes no use of them for any other purpose.

After a successful LOCATE, a corresponding modification to the module IEFW21SD tests for multi-volume, mixed type catalog entry information and resets this information in core to appear as a single-volume direct access device. Device allocation then assumes that the required data set will reside on this direct access volume, although it is not yet present.

Further modifications, made at a point in the module IEFSD062 after all step device allocations have been completed, LINK to a new module AECTDSNQ and then ATTACH a new module AECTDISK.

AECTDSNQ performs necessary enqueuing safety functions, including the conversion of any DISP=SHR shared data set enqueueing for retrieval data sets to DISP=OLD exclusive enqueueing. This avoids the possibility of competing tasks each trying to restore a shared data set.

AECTDISK performs the actual restoration of archived data sets, using the information addressed by the TCBCMP field of the initiator's task control block. Restorations are sorted to ensure minimal tape movement. Before the first restoration, the mounting of the #INITX control volume is simulated by clearing the NOT READY device status bit in the allocated tape unit control block (UCB). This avoids intervention-required messages for the non-existent #INITX volume when archive tapes need to be mounted. The CSECT AETD within AECTDISK performs necessary housekeeping functions, including the scheduling of multiple data set restorations and the allocation of skeleton data sets for data set

retrieval. The CSECTs, HSC and IGG019WS, contain subsets of the high speed data copying facilities of AEMOVE [Cox 1978] for copying from tape to disk. The first record of each archived data set contains data set decriptions used by AETD in direct access space allocations. Depending on the success or otherwise of each data set restoration, the data set is re-catalogued on either disk or tape. AECTDISK is ATTACHed as a job-step task, and is ATTACHed recursively for error recovery in the event of any ABEND (system error) during data set restorations.

Appendix A details the SYSGEN procedure for installing the archival and retrieval system. Appendix B is an actual job listing for archival and retrieval installation.

4. CONCLUSIONS

The archival and retrieval system outlined provides a simple and effective means of minimising unwanted data set occupancy of expensive, on-line computer disk storage. The system is efficient and should prove useful to any IBM360 or IBM370 installation which uses the MVT operating system.

5. ACKNOWLEDGEMENT

The authors wish to acknowledge the work done by Mr R.P. Backstrom [Backstrom 1977], in providing an interactive computing utility for the easy listing and recovery of archived data set information.

6. REFERENCES

Cox, G.W. [1978] - AEMOVE : A utility for the maintenance of direct access storage space. (Proposed AAEC/E report.)

Backstrom, R.P. [1977] - $.LIB : An interactive archive data set recovery program for the IBM360 Computer (unpublished paper).

APPENDIX A

INSTALLATION OF THE ARCHIVAL AND RETRIEVAL SYSTEM

All source data necessary for the installation of the archival and retrieval system are contained within the partitioned data set AMC.GENS. This is available for distribution in unloaded form on 800 BPI magnetic tape and is suitable for recovery by the IBM utility IEHMOVE. The tape volume serial is AMCGEN.

The data set AMC.GENS contains the following members: AECTDISK, AECTDSNQ, AEFSD162, AEFVMLS1, AETD, ASMLKED, HSC, IEFCVOL2, IEFSD162, IEFVMLS1, IGG019WS, REGS and REL.

AEFSD162 and AEFVMLS1 contain the necessary changes to the IBM modules IEFSD162 and IEFVMLS1 in IBM utility IEBUPDTE format. REGS and REL contain assembly macros for use when assembling HSC. ASMLKED contains the job-steps necessary to install this system : a listing of ASMLKED is given in Appendix B. The remaining members have been discussed earlier in this report.

APPENDIX B

ARCHIVAL AND RETRIEVAL INSTALLATION JOB LISTING

```
//*JOBPARM      L=17                                                                    00000010
//* DATA SET ARCHIVAL AND RETRIEVAL ASSEMBLIES AND LINK EDITS                          00000020
//*                                                                                    00000030
//* REQUIRES PRE-ALLOCATED AND CATALOGED DATA SETS -                                   00000040
//*    AMC.GENO        SPACE=(400,(4,1,1)) RECFM FB LRECL 80                           00000050
//*    AMC.LINKLIB     AS FOR SYS1.LINKLIB, PLUS 3 ENTRIES AND 10000 BYTES             00000060
//*                                                                                    00000070
//UPDGENSM PROC DSN='AMC.GENS',OBJDSN='AMC.GENO'                                       00000080
//UPD     EXEC  PGM=IEBUPDTE                                                         00000090
//SYSPRINT DD   DUMMY                                                                00000100
//SYSUT1   DD   DSN=&DSN.(&NAME),DISP=SHR                                            00000110
//SYSUT2   DD   DSN=&&OBJ,DISP=(,PASS),SPACE=(3200,(60,40)),                         00000120
//              DCB=(RECFM=FB,BLKSIZE=3200,LRECL=80),UNIT=SYSDA                      00000130
//SYSIN    DD   DSN=&DSN.(&SYSIN),DISP=SHR                                           00000140
//*                                                                                  00000150
//ASM     EXEC  PGM=IEUASM,PARM='DECK,NOLOAD',COND=(0,NE,UPD)                        00000160
//SYSPRINT DD   SYSOUT=A,                                                            00000170
//              DCB=(BLKSIZE=1210,LRECL=121,RECFM=FBSM)                              00000180
//SYSLIB   DD   DSN=SYS1.MACLIB,DISP=SHR,                                            00000190
//              DCB=(RECFM=FB,BLKSIZE=7200,LRECL=80)                                 00000200
//         DD   DSN=SYS1.MODGEN,DISP=SHR,                                            00000210
//              DCB=(RECFM=FB,BLKSIZE=7200,LRECL=80)                                 00000220
//         DD   DSN=SYS1.PVTMACS,DISP=SHR,                                           00000230
//              DCB=(RECFM=FB,BLKSIZE=7200,LRECL=80)                                 00000240
//         DD   DSN=SYS1.TSOMAC,DISP=SHR,                                            00000250
//              DCB=(RECFM=FB,BLKSIZE=7200,LRECL=80)                                 00000260
//         DD   DSN=&DSN,DISP=SHR,                                                   00000270
//              DCB=(RECFM=FB,BLKSIZE=7200,LRECL=80)                                 00000280
//SYSUT1   DD   DSN=&&UT1,UNIT=SYSDA,SPACE=(1700,(50,100))                           00000290
//SYSUT2   DD   DSN=&&UT2,UNIT=(SYSDA,SEP=SYSUT1),                                   00000300
//              SPACE=(1700,(50,100))                                                00000310
//SYSUT3   DD   DSN=&&UT3,UNIT=(SYSDA,SEP=(SYSUT1,SYSUT2)),                         00000320
//              SPACE=(1700,(50,100))                                                00000330
//SYSPUNCH DD   DSN=&OBJDSN.(&NAME),DISP=OLD                                         00000340
//SYSIN    DD   DSN=&&OBJ,DISP=(OLD,DELETE)                                          00000350
//       PEND                                                                        00000360
//*                                                                                  00000370
//GENASM PROC DSN='AMC.GENS',OBJDSN='AMC.GENO'                                       00000380
//ASM     EXEC  PGM=IEUASM,PARM='DECK,NOLOAD',REGION=120K                            00000390
//SYSPRINT DD   SYSOUT=A,                                                            00000400
//              DCB=(BLKSIZE=1210,LRECL=121,RECFM=FBSM)                              00000410
//SYSLIB   DD   DSN=SYS1.MACLIB,DISP=SHR,                                            00000420
//              DCB=(RECFM=FB,BLKSIZE=7200,LRECL=80)                                 00000430
//         DD   DSN=&DSN,DISP=SHR,                                                   00000440
//              DCB=(RECFM=FB,BLKSIZE=7200,LRECL=80)                                 00000450
//SYSUT1   DD   DSN=&&UT1,UNIT=SYSDA,SPACE=(1700,(50,100))                           00000460
//SYSUT2   DD   DSN=&&UT2,UNIT=(SYSDA,SEP=SYSUT1),                                   00000470
//              SPACE=(1700,(50,100))                                                00000480
//SYSUT3   DD   DSN=&&UT3,UNIT=(SYSDA,SEP=(SYSUT1,SYSUT2)),                         00000490
//              SPACE=(1700,(50,100))                                                00000500
//SYSPUNCH DD   DSN=&OBJDSN.(&NAME),DISP=OLD                                         00000510
//SYSIN    DD   DSN=&DSN.(&NAME),DISP=SHR                                            00000520
//       PEND                                                                        00000530
//*                                                                                  00000540
//ASM1    EXEC  UPDGENSM,SYSIN=AEFVML51,NAME=IEFVML51                                00000550
//ASM2    EXEC  UPDGENSM,SYSIN=AEFSD162,NAME=IEFSD162                                00000560
//ASM3    EXEC  GENASM,NAME=IEFCVOL2                                                 00000570
//ASM4    EXEC  GENASM,NAME=AECTDSND                                                 00000580
//ASM5    EXEC  GENASM,NAME=AECTDISK                                                 00000590
//ASM6    EXEC  GENASM,NAME=AETD                                                     00000600
//ASM7    EXEC  GENASM,NAME=HSC                                                      00000610
//ASM8    EXEC  GENASM,NAME=IGG019WS                                                 00000620
//*                                                                                  00000630
```

APPENDIX B (Continued)

```
//LKD1    EXEC  PGM=IEWL,COND=(0,LT),PARM='NCAL,LIST,MAP,LET,REFR,REUS'  00000640
//SYSPRINT DD   SYSOUT=A                                                00000650
//SYSLMOD  DD   DSN=AMC.LINKLIB,DISP=OLD                                00000660
//REF1     DD   DSN=AMC.GEN0,DISP=SHR                                   00000670
//REF2     DD   DSN=SYS1.LINKLIB,DISP=SHR                               00000680
//SYSUT1   DD   UNIT=SYSDA,SPACE=(1024,(5,10))                          00000690
//SYSLIN   DD   *                                                       00000700
         INCLUDE REF1(IEFSD162)                                        00000710
         INCLUDE REF2(IEFSD062)                                        00000720
         ALIAS   IEFV4221                                              00000730
         ENTRY   IEFSD062                                              00000740
         NAME    IEFSD062(R)                                           00000750
/*                                                                     00000760
//LKD2    EXEC  PGM=IEWL,COND=(0,LT),PARM='NCAL,LIST,MAP,LET'          00000770
//SYSPRINT DD   SYSOUT=A                                                00000780
//SYSLMOD  DD   DSN=AMC.LINKLIB,DISP=OLD                                00000790
//REF1     DD   DSN=AMC.GEN0,DISP=SHR                                   00000800
//REF2     DD   DSN=SYS1.LINKLIB,DISP=SHR                               00000810
//SYSUT1   DD   UNIT=SYSDA,SPACE=(1024,(5,10))                          00000820
//SYSLIN   DD   *                                                       00000830
         CHANGE  IEFCVOL2(AEFCVOL2)                                    00000840
         INCLUDE REF2(IEFMCVOL)                                        00000850
         ALIAS   IEFCVOL1,AEFCVOL2,IEFCVOL3                            00000860
         ENTRY   IEFCVOL1                                              00000870
         NAME    IEFMCVOL(R)                                           00000880
         CHANGE  IEFSD059(IEFVM2)          SYSGEN DEPENDENT            00000890
         INCLUDE REF1(IEFVMLS1)                                        00000900
         INCLUDE REF2(IEFW21SD)                                        00000910
         ALIAS   IEFVM1,IEFXA,IEFVMCVL                                 00000920
         ENTRY   IEFW21SD                                              00000930
         NAME    IEFW21SD(R)                                           00000940
         INCLUDE REF1(IEFCVOL2)                                        00000950
         ENTRY   IEFCVOL2                                              00000960
         NAME    IEFCVOL2(R)                                           00000970
         INCLUDE REF1(AECTDSNQ)                                        00000980
         ENTRY   AECTDSNQ                                              00000990
         NAME    AECTDSNQ(R)                                           00001000
         INCLUDE REF1(AECTDISK)                                        00001010
         INCLUDE REF1(AETD,HSC,IGG019WS)                               00001020
         ENTRY   AECTDISK                                              00001030
         NAME    AECTDISK(R)                                           00001040
/*                                                                     00001050
//*                                                                    00001060
//CPY     EXEC  PGM=IEBCOPY,COND=(0,LT)                                00001070
//SYSPRINT DD   DUMMY,DCB=BLKSIZE=121                                   00001080
//SYSUT1   DD   DSN=SYS1.LINKLIB,DISP=SHR                               00001090
//SYSUT2   DD   DSN=AMC.LINKLIB,DISP=OLD                                00001100
//SYSIN    DD   DUMMY                                                   00001110

END OF FILE. (TTR=000706)
```