# NOS/BE TO NOS USAGE MIGRATION

## CONTENT

- BRIEF HISTORY AND ARCHITECTURE OF NOS
- SYSTEM ACCESS
- PERMANENT FILES
- MAGNETIC TAPES
- SOURCE STORAGE, USAGE AND MAINTENANCE
- OBJECT STORAGE, USAGE AND MAINTENANCE
- INTERACTIVE, REMOTE BATCH AND TELEPROCESSING USAGE
- SOURCE EDITORS
- PLANNING FOR THE TRANSITION

## WHY ME?

- CYBER EXPERIENCE STARTED ON SCOPE 3.1.2 IN 1969.

- RECENT CONVERSION EXPERIENCE.

  PERSONALLY CONVERTED IN 1978.

  EMPATHY FOR NOS/BE MIGRATION CONCERNS.

- PRODUCT SET FIELD SUPPORT

  EXPOSURE TO NOS, SCOPE 2 AND NOS/BE USER PROBLEMS.

- USE BOTH SYSTEMS DAILY.

## NOS/BE TO NOS USAGE MIGRATION

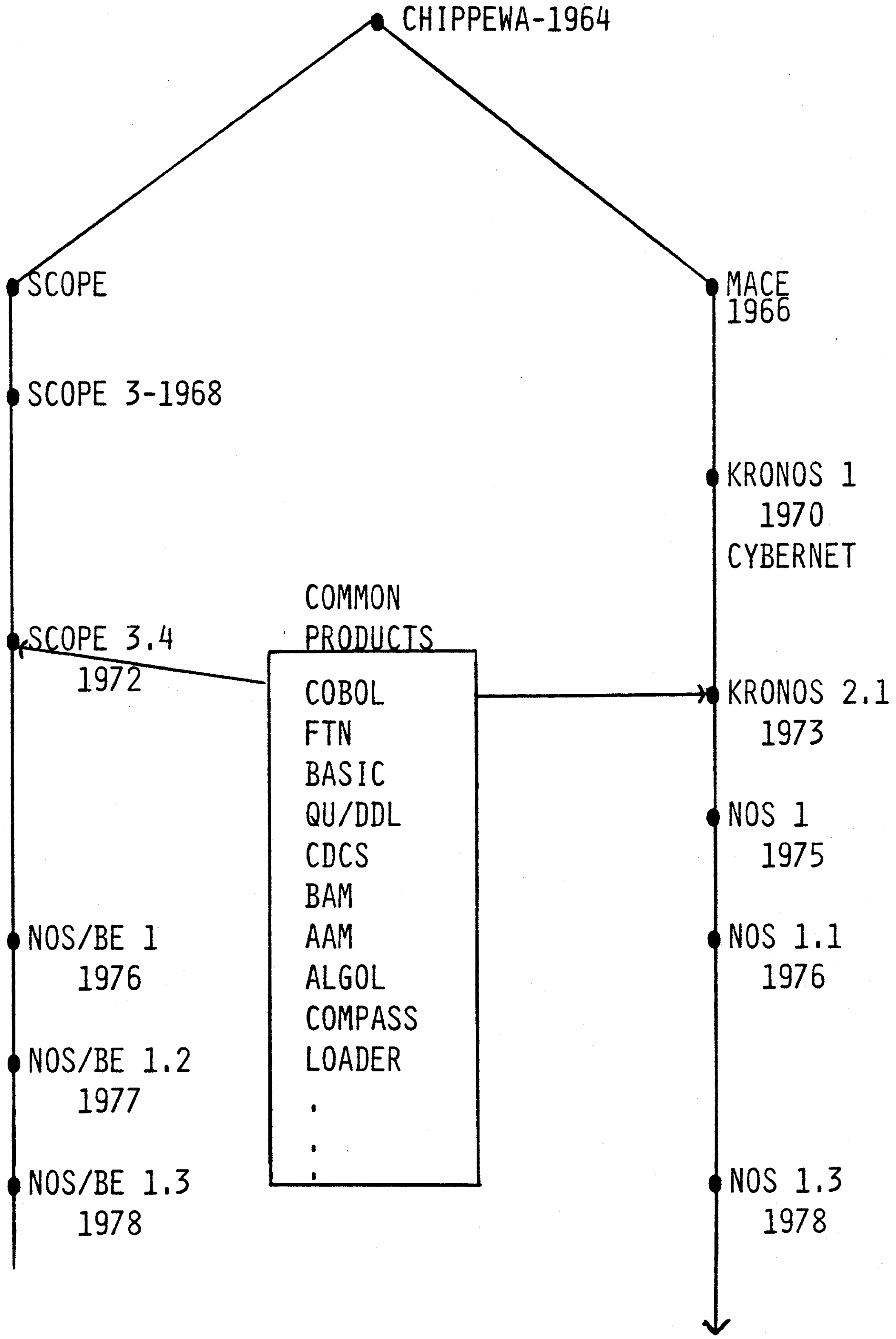
- SYSTEM USAGE - 98% OF PEOPLE AT A SITE.

  THE SUBJECT OF THIS SESSION.

- SYSTEMS ANALYST - 1% OF PEOPLE AT A SITE.

- OPERATOR TRAINING - 1% OF PEOPLE AT A SITE.

- CONVERSION AIDS


CHIPPEWA-1964
SCOPE
SCOPE 3-1968
SCOPE 3.4
1972
NOS/BE 1
1976
NOS/BE 1.2
1977
NOS/BE 1.3
1978
MACE
1966
KRONOS 1
1970
CYBERNET
KRONOS 2.1
1973
NOS 1
1975
NOS 1.1
1976
NOS 1.3
1978
COMMON
PRODUCTS
COBOL
FTN
BASIC
QU/DDL
CDCS
BAM
AAM
ALGOL
COMPASS
LOADER
.
.
.

COMPUTER IS USED FOR-

- AUTOMOTIVE
- BANKING
- CONSTRUCTION
- EDUCATION
- PETROLEUM
- WEATHER
- ETC.

COMPUTER IS NOT USED FOR -

- CPU MTR
- FANCY OPERATOR DISPLAYS
- BUILDING SYSTEMS
- ETC.

CDC COMPUTERS ARE FOR SOLUTIONS NOT MACHINE CYCLES.

APPLICATIONS REQUIRE -

- COMMON PRODUCTS - COBOL, FTN, CRM, ETC.
- VEHICLE TO RUN COMMON PRODUCTS
- "SERVICE" FACILITIES

  FILE RETENTION
  ACCOUNTING
  ETC.

## NOS CHARACTERISTICS

- SYSTEM CONTROL THRU VALIDATION

  AVOID WASTE OF RESOURCES.
  PROTECTION OF USER'S RESOURCES.

- SECURITY - ACCESS TO PFS, ACCOUNTING, ETC., IS THRU A USER CONTROLLED PASSWORD.

- EVERYTHING BELONGS TO A 'USER'.

- RELIABILITY - CURRENT FAILURE RATE IS LESS THAN 1 PER MONTH. RESULT OF MAINTAINABILITY IMPROVEMENTS THROUGHOUT THE ENTIRE PSR LEVEL 400 SERIES.

- SIMPLICITY - MORE COMMANDS; FEWER PARAMETERS. VERY FEW 'INPUT' DIRECTIVES.

- RESOURCE OWNERSHIP - THERE IS A 1 TO 1 RELATIONSHIP BETWEEN PEOPLE AND USER NUMBERS.

- OPERATOR DOES NOT MAKE USER DECISIONS.

- MANY WAYS TO DO A TASK.

  CHOOSE ONE - I'LL HELP.

## SYSTEM ACCESS

### USER STATEMENT

- Common and required for all access

  BATCH
  INTERACTIVE
  REMOTE BATCH
  TELEPROCESSING
  TRANSACTION

- Password protection controlled by USER

- Permanent File Access

- Resource Limits

- Billing (Accounting)

- Secondary USER statements can be entered in the middle of a session (site must choose this option). This feature is primarily for permanent file access.

# FAMILIES AND USER INDICES

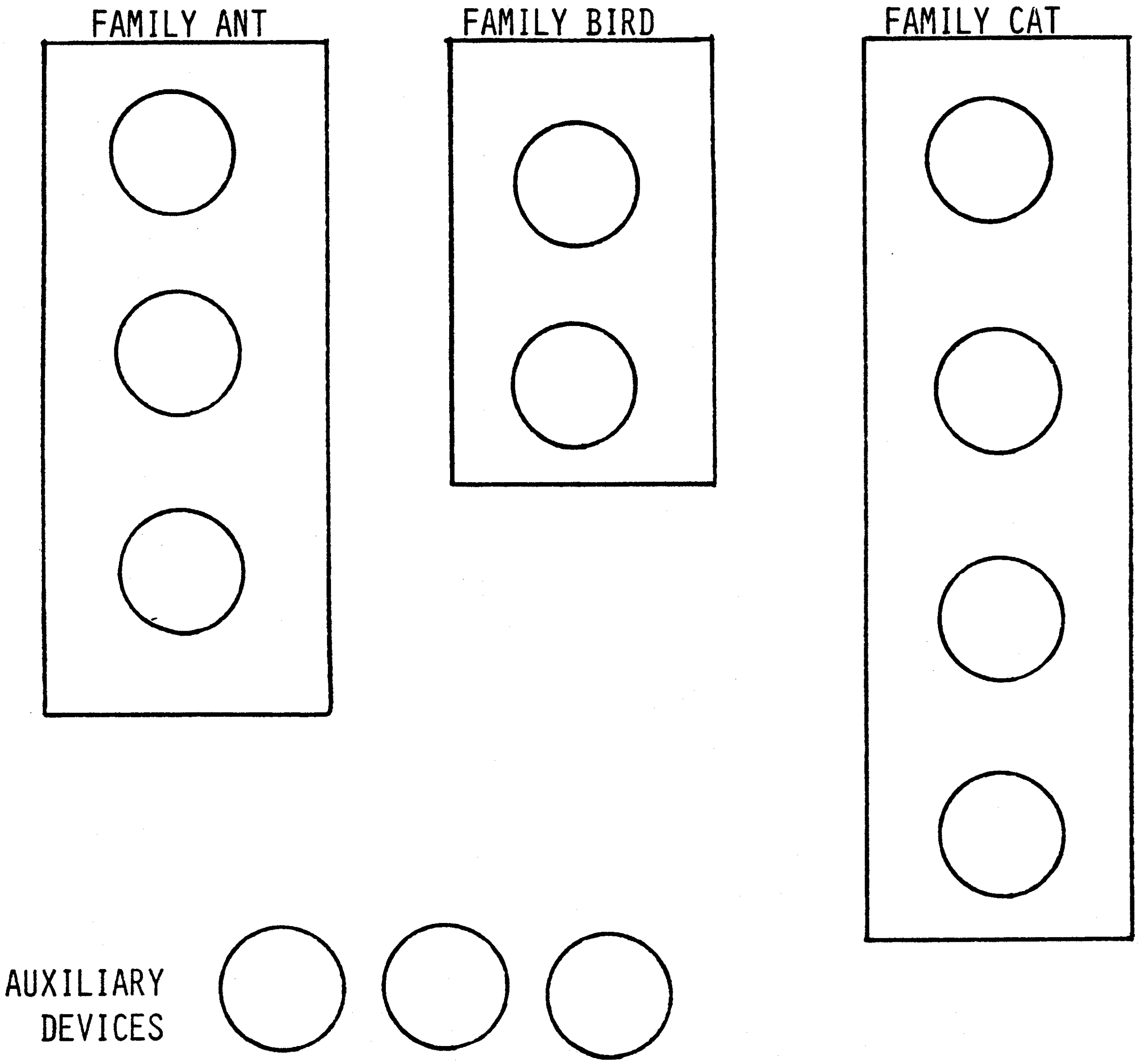
- FAMILY IS THE TECHNIQUE TO GROUP MEMBERS OF USER COMMUNITIES TOGETHER.

  CONTAINS THE VALIDATION FILE FOR USERS IN THAT FAMILY.

  PROVIDES A BACKUP FACILITY SINCE IT IS VERY EASY TO MOVE USERS FROM A DOWN MACHINE.

- USER INDEX IS THE TECHNIQUE TO SPREAD THE USERS IN A FAMILY OVER THE DISKS IN A FAMILY. THE DISK THAT A USER INDEX IS 'ASSIGNED' CONTAINS THAT USER'S -

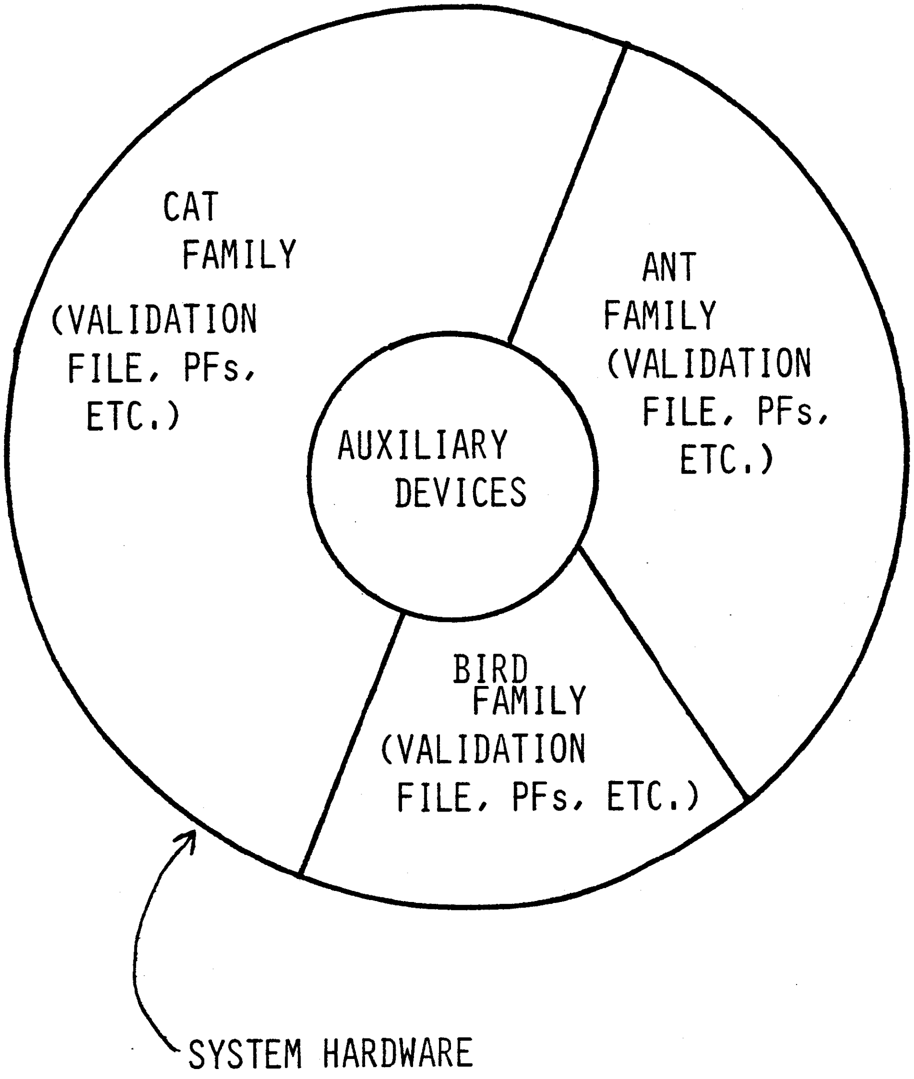
  PF CATALOG<br>
  INDIRECT PFs<br>
  SOME DIRECT PFs



EACH 'DEVICE' WITHIN A FAMILY OR AUXILIARY DEVICE MAY BE 1 TO 8 DISK SPINDLES.
EACH AUXILIARY DEVICE MAY BE DECLARED

-PRIVATE - ONLY VALIDATED USER CAN CREATE FILES.
ALTERNATE USERS CAN BE PERMITTED TO ACCESS FILES

-PUBLIC - ANY USER CAN CREATE OR ACCESS FILES


CAT FAMILY (VALIDATION FILE, PFs, ETC.)
ANT FAMILY (VALIDATION FILE, PFs, ETC.)
AUXILIARY DEVICES
BIRD FAMILY (VALIDATION FILE, PFs, ETC.)
SYSTEM HARDWARE

- HARDWARE
- FAMILY
- USER INDEX
- FILES

- AUXILIARY DEVICES

## FILE STRUCTURE

Essentially the same.

NOS/BE

| | EOR | EOF | EOI |
|---|---|---|---|
| CARDS | 7/8/9 | IMPOSSIBLE | 6/7/8/9 |
| DISK | SHORT PRU + LEVEL=0 | ZERO LENGTH PRU + LEVEL = 17 | END OF RBR CHAIN |
| TAPE (SI) | The same as disk. | The same as disk. | DOUBLE TAPEMARKS |

NOS

| | EOR | EOF | EOI |
|---|---|---|---|
| CARDS | 7/8/9 | 6/7/9 | 6/7/8/9 |
| DISK | SHORT PRU + LINK TO NEXT PRU. | ZERO LENGTH PRU + LINK TO NEXT PRU | ZERO LENGTH PRU<br>NO LINK |
| TAPE (SI/I) | SHORT PRU + LEVEL = 0 | ZERO LENGTH PRU + LEVEL = 17 | DOUBLE TAPEMARKS |

- There are no random bit problems on NOS. Any file (random or sequential) may be copied with the COPYEI directive

## BATCH ACCESS

```
FOX,T40.
USER,JFOX,WEIRD [,family].
CHARGE,MYDEPT,MYPROJECT.
:
```

- JOB STATEMENT - Only 3 parameters!

  T - Time limit in CP seconds for each job step.

  CM - Specifies MAXFL. Not recommended. MAXFL must satisfy -

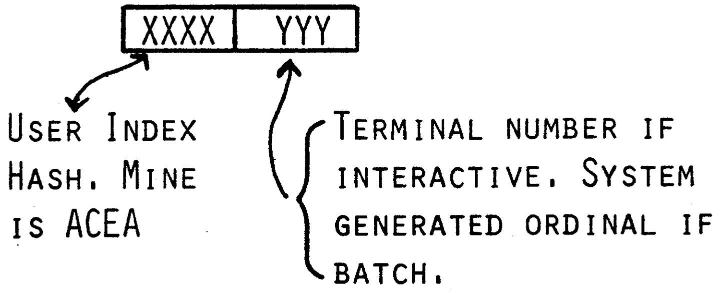
  RFL ≤ MFL ≤ MAXFL ≤ Validated maximum.

  EC - Extended Central Storage

In NOS many of the NOS/BE job parameters are specified by control statements within the job when needed. There are generally more control statements on NOS with fewer parameters.

## BATCH ACCESS - CONTINUED

- JOB NAME IS UNIQUE AND CONSISTS OF TWO PARTS-



USER INDEX HASH. MINE IS ACEA

TERMINAL NUMBER IF INTERACTIVE. SYSTEM GENERATED ORDINAL IF BATCH.

- FAMILY IS VALIDATION FILE TO CHECK FOR USER NAME.

- USER NAME (JFOX) IS THE OPERATIONS SUPPLIED NAME IN THE VALIDATION FILE. THIS IS ASSOCIATED WITH A USER INDEX THAT DETERMINES WHERE MY PF CATALOG AND FILES RESIDE.

- PASSWORD (WEIRD) IS SUPPLIED, CHANGED AND MAINTAINED BY ME (THE USER). I CAN CHANGE IT WHEN APPROPRIATE.

- DEPARTMENT (MYDEPT). EACH USER CAN BE VALIDATED TO CHARGE TO CERTAIN DEPARTMENTS AND PROJECTS WITHIN DEPARTMENTS.

# BATCH ACCESS

## SPECIAL CONSIDERATIONS

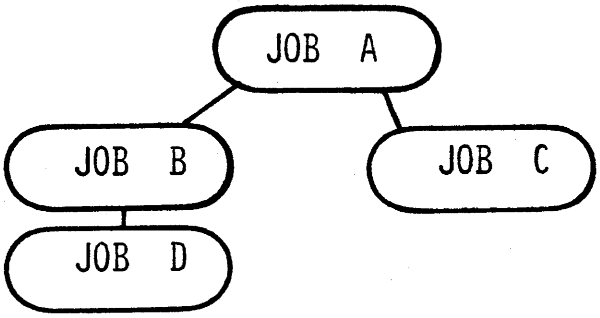
- RERUN<br>NORERUN - USER CAN EXPLICITLY TURN RERUN FLAG ON OR OFF AS DESIRED IN JOB. NO STRANGE PF TRICKS LIKE ON NOS/BE.

- EXIT<br>NOEXIT<br>ONEXIT - NO EXIT(S) BUT FACILITY TO TURN EXIT GLOBALLY ON OR OFF

# BATCH ACCESS

## SPECIAL CONSIDERATIONS

- JOB NAME - CODE IS AVAILABLE VIA THE UNDERGROUND. 'JOB ADMINISTRATION' ADDRESSES THE ENTIRE PROBLEM AND IS SCHEDULED FOR A FUTURE RELEASE.



- JOB DEPENDENCIES - NO JOB STATEMENT PARAMETER. CAN USE ANY OF THE FOLLOWING AND AVOID WASTED NOS/BE FNT ENTRIES AND READING JOBS IN BACKWARD.

  - ROUTE OR LDI COMBINED WITH CCL IS A VERY POWERFUL COMBINATION.

  - PF INTERLOCKS COMBINED WITH ROLLOUT.

  - SPECIAL PRODUCTS LIBRARY.

    WAIT, RESUME, LISTDJ, DROP

# PERMANENT FILES

## CHARACTERISTICS

- OWNERSHIP
- TYPES - DIRECT AND INDIRECT
- CATEGORIES - PUBLIC, SEMI-PRIVATE, PRIVATE
- ACCESS MODES
- COMMANDS

# PERMANENT FILES

## CHARACTERISTICS

- FILES BELONG TO SPECIFIC USERS.

  NOS IS SET UP TO KEEP USERS AWAY FROM EACH OTHER. IF YOU WANT TO SHARE FILES YOU MUST MAKE THEM PUBLIC OR PERMIT SPECIFIC USERS TO ACCESS THE FILE(S).

- TWO TYPES OF FILES - DIRECT AND INDIRECT

  WHEN AN INDIRECT FILE IS ASSIGNED TO A JOB YOU GET A COPY IN SCRATCH SPACE. WHEN A DIRECT FILE IS ASSIGNED TO A JOB YOU GET A REAL FILE.

  DIRECT FILES TAKE UP 'MORE' DISK SPACE- THE OLD RECORD BLOCK PROBLEM. INDIRECT PERMANENT FILE SPACE IS MAINTAINED IN INCREMENTS OF PRUs BY NOS. DIRECT FILES ARE IDEAL FOR LARGE DATA FILES WHILE INDIRECT FILES ARE IDEAL FOR SMALL (LESS THAN A LOGICAL TRACK) DATA FILES.

## PERMANENT FILES CHARACTERISTICS - CONTINUED

- THREE CATEGORIES

PUBLIC - ANYONE WHO KNOWS USER-NAME,
FILE NAME AND PASSWORD IF
SPECIFIED CAN ACCESS.

SEMI-PRIVATE - SAME AS PUBLIC EXCEPT
THAT A RECORD IS KEPT OF WHO
AND WHEN THE FILE WAS ACCESSED.

PRIVATE - ONLY YOU CAN ACCESS OR SPECIFIC
USERS PERMITED FOR ONLY SPECIFIC
ACCESS MODE.

## PERMANENT FILE CHARACTERISTICS - continued

- Eight Access MODEs. Each mode is 'higher' than previous.

| | |
|---|---|
| N | - NULL |
| E | - EXECUTE ONLY |
| R | - READ ONLY |
| RA | - READ and/or APPEND |
| RM | - READ and/or MODIFY |
| A | - APPEND |
| M | - MODIFY |
| W | - WRITE |

## PERMANENT FILE CHARACTERISTICS - CONTINUED

- COMMANDS

| | | |
|---|---|---|
| DIRECT AND INDIRECT | CATLIST | - SORTED LIST OF PFs AND MISC. INFO. LIKE AUDIT EXCEPT MUCH MORE USER FRIENDLY! |
| | PURGE | - REMOVE A FILE FROM PF CATALOG. |
| | PERMIT | - ALLOW OTHER USERS TO ACCESS. |
| | CHANGE | - CHANGE PERMISSIONS. |
| | PURGALL | - PURGE ALL FILES! |
| DIRECT ONLY | DEFINE | - ESTABLISH A DIRECT FILE. |
| | ATTACH | - ACCESS A DIRECT FILE. |
| INDIRECT ONLY | SAVE | - ESTABLISH AN INDIRECT FILE. |
| | GET | - ACCESS AN INDIRECT FILE. |
| | REPLACE | - REPLACE AN EXISTING INDIRECT FILE. |
| | APPEND | - ADD A LOCAL FILE TO THE END OF AN INDIRECT FILE. |

# DIRECT FILES

## CREATION

```
DEFINE,[LFN=]PFN[/CT=CATEGORY,M=MODE,
                    PW=PASSWORD].
```

PLACE WHERE YOU WOULD NORMALLY INSERT THE NOS/BE 'REQUEST,LFN,*PF' STATEMENT.

'CATEGORY' MAY BE

P  - PRIVATE<br>
S  - SEMI-PRIVATE<br>
PU - PUBLIC

## DIRECT FILES - CONTINUED

THE MEANING OF THE 'MODE' PARAMETER DEPENDS UPON THE CATEGORY OF THE FILE -

PRIVATE - MODE HAS NO MEANING SINCE OWNER CAN ACCESS AT ANY LEVEL AND EACH ALTERNATE USER MUST EXPLICITLY BE PERMITTED TO ACCESS THE FILE.

PUBLIC AND SEMI-PRIVATE - MODE IS THE MAXIMUM ACCESS MODE FOR ALL ALTERNATE USERS EXCEPT THOSE WHO HAVE BEEN EXPLICITLY PERMITTED BY THE OWNER. AN EXPLICIT PERMIT TAKES PRECEDENCE OVER MODE ON THE DEFINE FOR PERMITTED USERS.

## DIRECT FILES - CONTINUED

### ACCESS

```
ATTACH,[LFN=]PFN[/UN=USER-NAME,M=MODE,
                   PW=PASSWORD].
```

'USER-NAME' IS OPTIONAL IF PFN IS YOUR FILE.

'MODE' IS THE ACCESS MODE DESIRED FOR THIS ACCESS. IF AN ALTERNATE USER ATTEMPTS ACCESS OF A PERMITED PRIVATE FILE AND EXCEEDS THE PERMITED MODE, THE FOLLOWING MESSAGE IS DISPLAYED -

```
PFN NOT FOUND
```

THE SAME THING HAPPENS FOR PUBLIC AND SEMI-PRIVATE FILES WHEN THE AUTHORIZED MODE IS EXCEEDED.

DEFAULT 'MODE' IS READ IF M IS NOT SPECIFIED.

# INDIRECT FILES

## CREATION

```
SAVE,[LFN=]PFN[/CT=CATEGORY,M=MODE,
                PW=PASSWORD].
```

COPIES LFN TO AREA NOW CALLED PFN.

```
REPLACE,[LFN=]PFN[/UN=USER-NAME,PW=PASSWORD].
```

REPLACE DOES AN IMPLICIT PURGE AND SAVE.
IT IS VERY DIFFICULT TO GO BACK TO NOS/BE
AND HASSLE CYCLES AFTER USING THIS COMMAND.

## INDIRECT FILES - CONTINUED

### ACCESS

```
GET,[LFN=]PFN[/UN=USER-NAME,PW=PASSWORD].
```

COPIES PFN TO LFN.

### EXTENSION

```
APPEND,PFN,LFN1,LFN2...[/UN=USER-NAME,
                      PW=PASSWORD].
```

COPIES LFN1, LFN2, ETC., TO THE END OF INFORMATION OF PFN.

## PERMANENT FILES

### SPECIAL CONSIDERATIONS

- NO REQUEST,LFN,*PF
- SEQUENTIAL REWRITE IS AUTOMATIC
- OVERWRITE IS AUTOMATIC
- EXTEND IS ALWAYS AUTOMATIC ON DIRECT FILES
- NO PFD OVERFLOW PROBLEMS
- ALTER

  TO TRUNCATE FILE USE WRITER,LFN.

  TO 'REDEFINE' FILE JUST REWIND AND START WRITING.
- CATLIST IS 'TERMINAL-FRIENDLY'

  FN<br>
  UNIVERSAL CHARACTER<br>
  LO<br>
  LENGTH STATEMENT

## PERMANENT FILES

### SPECIAL CONSIDERATIONS - CONTINUED

- REPLACE IS VERY EASY TO LOVE.
- RANDOM FILES CAN BE COPIED DISK TO DISK (NO RANDOM BIT PROBLEM) AND BE USED. RANDOM FILES CAN ALSO BE COPIED TO TAPE FOR BACKUP PURPOSES AND COPIED BACK TO DISK FOR USE.
- 40 CHARACTER PFN - BITE THE BULLET
- USER DUMP/LOAD
  - IF THE RANDOM BIT IS THE PROBLEM ON NOS/BE; THERE IS NO PROBLEM ON NOS. USE COPYEI.
- ARCHIVING

  SPL - SPECIAL PRODUCTS LIBRARY.
  FUTURE SYSTEM RELEASE (1ST HALF 80).
- CYCLES - MUST BE NICE TO HAVE THAT MUCH DISK SPACE...
  - REWIND THEN WRITE
  - REPLACE
  - USE CCL PROC

```
.PROC,NEWDEF,LOCAL,PERM.

 PURGE,PERM→1/NA.                      Purge oldest

 CHANGE,PERM→1 = PERM→2/NA.            2 to oldest

 CHANGE,PERM→2 = PERM→3/NA.            3 to 2

 DEFINE,LOCAL = PERM→3.                New to 3
```

```
 NEWDEF,PUPPY,DOG.
```

NOTES -

```
         CHANGE,NEW=OLD/NA.

         DOG1 = OLDEST
         DOG2 =
         DOG3 = NEWEST
```

## PERMANENT FILES

### SPECIAL CONSIDERATIONS - CONTINUED

- VERY LARGE FILES - USE POSITIVE CONTROL (KNOW WHERE THE FILE IS LOCATED!!).
  - USE AUXILARY MULTI-SPINDLE LOGICAL DEVICE PRIVATE PACKS.

    PACKNAM
  - USE MULTI-SPINDLE DEVICE FAMILIES.

  EITHER WILL RESULT IN BETTER ACCESS.

- MACROS - WATCH OUT FOR PF FET - CRM FIT CONFLICT.

# MAGNETIC TAPE

- RESERVATION
- ASSIGNMENT
- RELEASE
- DIFFERENCES

## MAGNETIC TAPE

- RESERVATION

IF ONLY ONE DEVICE (7 TRACK DRIVE OR 9 TRACK DRIVE OR DISK DRIVE) IS REQUIRED NO RESERVATION IS REQUIRED.

IF MORE THAN ONE DEVICE IS REQUIRED THEN A RESOURC STATEMENT MUST BE ENTERED IN THE JOB STREAM BEFORE THE FIRST DEVICE IS ASSIGNED.

```
RESOURC(MT=2,NT=1)
```

THIS STATEMENT INDICATES THAT THE JOB NEEDS TO USE TWO 7 TRACK DRIVES AND ONE 9 TRACK DRIVE AT THE SAME TIME. THE NOS/BE USER WOULD HAVE ENTERED -

```
...MT2,NT1...
```

ON THE JOB STATEMENT.

THE COUNTS MAY BE LOWERED BY ANOTHER RESOURC STATEMENT.

## MAGNETIC TAPE

- ASSIGNMENT

Use the LABEL statement. It does everything. Most of the parameters are identical to NOS/BEs LABEL parameters.

```
LABEL,lfn,
  D = den,        den may be HI, HY, HD, GE, etc.
  F = fmt,        Data format - I,S,L,SI
                  (SI is a NOS/BE System
                  Internal).
 LB = type,       Label type may be -
                  KL - NOS labeled (ANSI)
                  KU - Unlabeled
                  NS - Non-standard
VSN = vsn,
 PO = option,     Processing option A,B,F,R,W,etc.
  R,              Read Existing label
 W)               Write new label
```

There are other parameters listed in the NOS R.M. Notice that there is no RING/NORING parameter. This is accomplished by the PO=W and PO=R options. PO also has many other options (parity error processing, etc.).

## MAGNETIC TAPE

THE VSN PARAMETER APPEARS INNOCENT BUT THERE IS ONE SITUATION TO BE AWARE OF -

NOS/BE IMPLEMENTED AN OLD ANSI LABEL STANDARD WHILE NOS IMPLEMENTED THE NEW ANSI STANDARD. THE VSN IS LEFT JUSTIFIED WITH BLANK FILL ON NOS WHILE IT IS RIGHT JUSTIFIED WITH ZERO FILL ON NOS/BE. TO READ A NOS/BE TAPE ON NOS, SPECIFY THE VSN AS A SIX CHARACTER FIELD WITH LEADING ZEROS.

VSN=4567 ON NOS/BE WOULD BE
VSN=004567 ON NOS

- RELEASE

RETURN AND UNLOAD OPERATE THE SAME ON NOS AND NOS/BE.

## SOURCE PROGRAM LIBRARIES

- UPDATE - It is the same on both systems. You know how to use it. Continue to use it. PLs can be COPYEIed since there is no random bit problem. Future development is scheduled for UPDATE. e.g. 8 bit support.

- MODIFY - Has similar features as UPDATE. Cannot be used when the PL resides on tape. Operating system software is maintained by MODIFY.

- XEDIT - An editor that works both in batch and interactive. More later in this presentation.

## OBJECT MODULE 'LIBRARIES'

- STORAGE

  SEQUENTIAL FILES
  RANDOM 'LIBRARY' FILES USABLE BY THE
  CYBER LOADER

- USAGE

- MAINTENANCE

## OBJECT MODULE LIBRARIES - CONTINUED

## USAGE

- SEQUENTIAL FILES OF RELOCATABLES OR ABSOLUTES PERFORM THE SAME WAY ON BOTH SYSTEMS.

- CAPSULES ARE STORED IN RANDOM LIBRARIES ON BOTH SYSTEMS. THEY ARE ACCESSED IN THE SAME WAY ON BOTH SYSTEMS.

- ABSOLUTES CURRENTLY CANNOT BE STORED IN NOS USER LIBRARIES.

  - IF PURPOSE OF INSERTING ABSOLUTES INTO A USER LIBRARY ON NOS/BE WAS TO SAVE DISK ACCESSES THEN FOL IS <u>FASTER</u> THAN NON-FOL NOS/BE LIBRARIES.

  - GTR (GET RECORDS) CAN BE USED TO ACCOMPLISH SOME FUNCTIONS OF A GLOBAL LIBRARY STATEMENT.

  - CODE IS AVAILABLE (VIA THE UNDERGROUND) FOR LIBGEN. IT MIGHT BE STANDARD WHEN YOU CONVERT.

## OBJECT MODULE LIBRARIES

### USAGE - CONTINUED

- RELOCATABLE MAIN PROGRAMS CAN BE STORED IN NOS USER LIBRARIES BUT ONLY THE CYBER LOADER KNOWS HOW TO READ GLOBAL LIBRARIES. NOS DOES NOT SEARCH GLOBAL LIBRARIES IN RESPONSE TO A NAME CALL LOAD. NOS DOES NOT INVOKE THE CYBER LOADER UNLESS EXPLICITLY CALLED. TRY THE FOLLOWING -

  ```
  LDSET,LIB=LIBNAME.
  MYPROG.
  ```

  CPUMTR (1AJ) CODE IS AVAILABLE VIA THE UNDERGROUND.

- CCL PROCEDURES (THE ONLY KIND) CURRENTLY CANNOT BE STORED IN NOS USER LIBRARIES.

  - PLACE YOUR PROCEDURES INTO AN INDIRECT PF CALLED PROCFIL. ENTER '-PROC-NAME,,P1,P2...' NOS (CCL) WILL DO AN AUTOMATIC GET OF THE PROPER PROCEDURE IN INTERACTIVE.

    'BEGIN,PROC-NAME,,P1,P2... IS REQUIRED IN BATCH.

## OBJECT MODULE LIBRARIES

## MAINTENANCE

- SEQUENTIAL FILES

  - COPYL - THE SAME
  - COPYLM - THE SAME
  - ITEMIZE- THE SAME
  - GTR - VERY HANDY EXTRACTION UTILITY!
  - LIBEDIT- FANCY, DIRECTIVE DRIVEN COPYL.
    CREATES A NEW 'CLEAN' FILE.

- RANDOM FILES (USER LIBRARIES)

  ONLY UTILITY IS LIBGEN. IT COPIES A <u>CLEAN</u> SEQUENTIAL FILE TO USER LIBRARY FORMAT.

  - USE A CCL PROCEDURE TO COMBINE THE MUNDANE TASKS OF GTR, LIBEDIT AND LIBGEN. RESULTS IN NO NOS/BE-LIKE DEAD DISK SPACE!

## OBJECT MODULE LIBRARIES

### MAINTENANCE

CONSIDER THE FOLLOWING TWO CCL PROCEDURES - ONE TO GENERATE AN 'EMPTY' USER LIBRARY AND ANOTHER TO MAINTAIN THE LIBRARY.

```
.PROC,EMPTYLB,LIBNAME,INP=#DATA.

RETURN,TEMP.

FTN,I=INP,L=0,B=TEMP.

LIBGEN,F=TEMP,P=LIBNAME.

RETURN,TEMP.

.DATA

       SUBROUTINE   AAAAAAA

       END
```

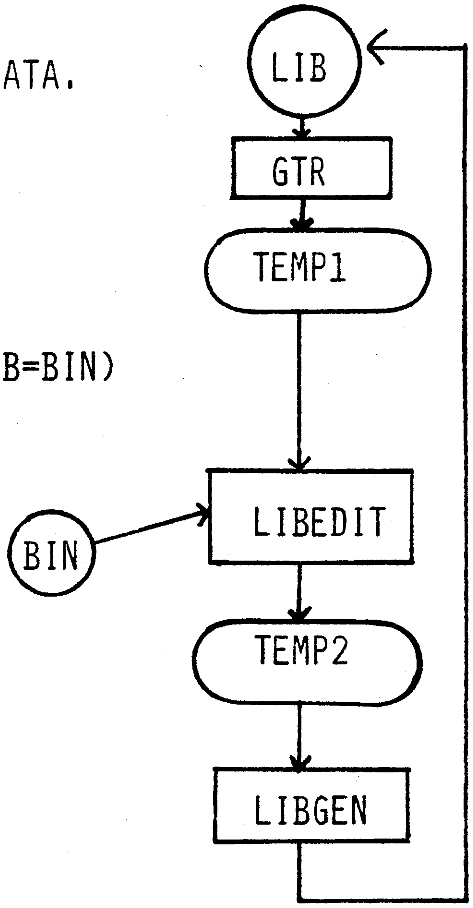
THIS PROCEDURE IS ACCESSED BY -

```
     EMPTYLB,MYLIB.
```

## OBJECT MODULE LIBRARIES

MAINTENANCE - CONTINUED

```
 .PROC,UPLIB,LIBNAME,BIN,INP=#DATA.

 RETURN(TEMP1,TEMP2)

 GTR(LIBNAME,TEMP1)ULIB/LIBNAME

 LIBEDIT(I=INP,P=TEMP1,N=TEMP2,B=BIN)

 LIBGEN(F=TEMP2,P=LIBNAME)

 .DATA

*INSERT,REL/AAAAAAA,REL/*
```



THE PROCEDURE IS ACCESSED BY -

```
UPLIB,MYLIB,MYLGO.
```

## OBJECT MODULE LIBRARIES

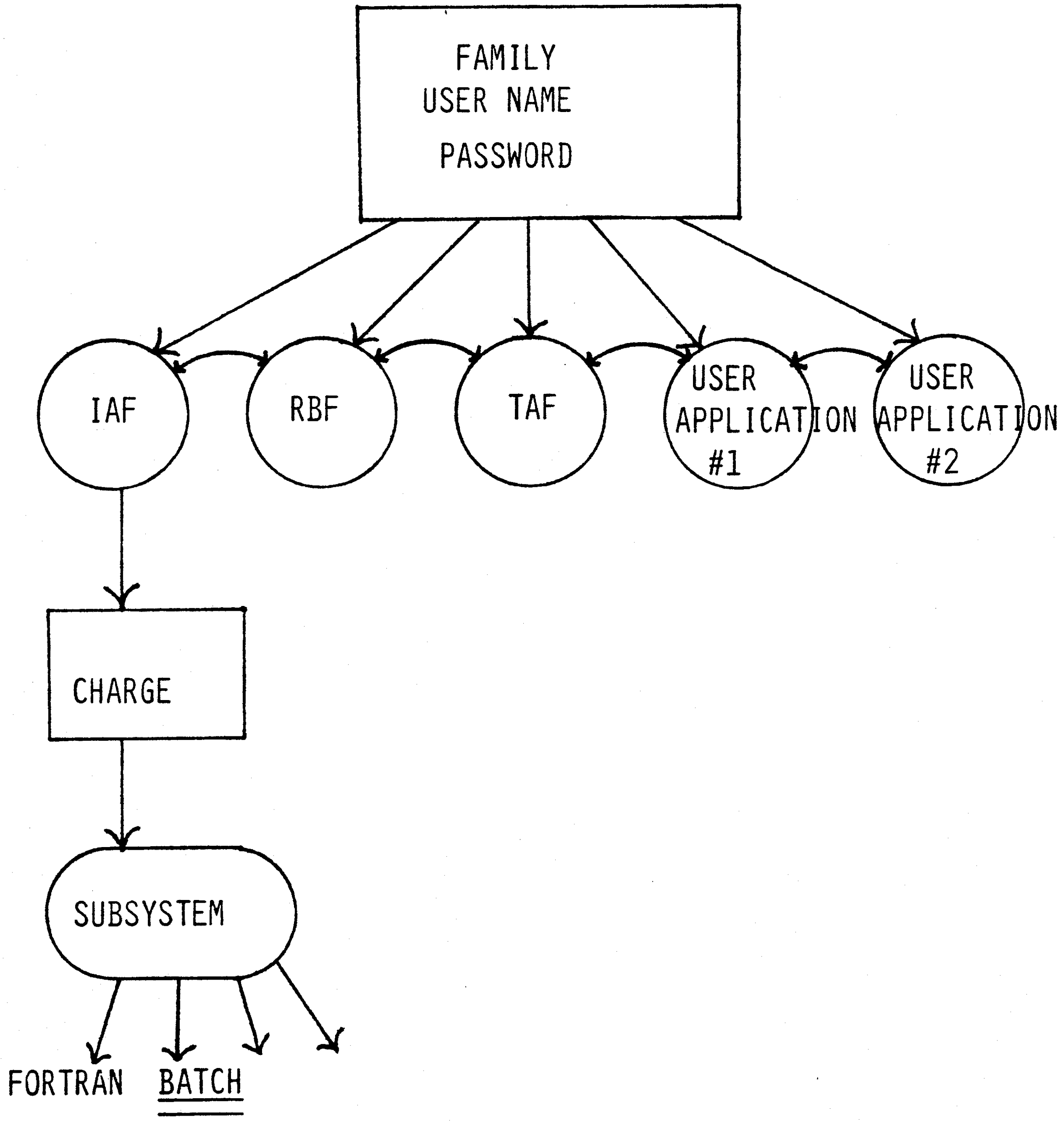
## MAINTENANCE - CONTINUED

GTR - A VERY POWERFUL UTILITY.

```
GTR,OLD-FILE,NEW-FILE. TYPE/NAMES
```

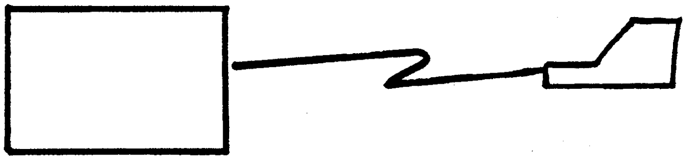
GTR SCANS 'OLD-FILE' FOR THE RECORD(S) CALLED 'NAMES' OF THE SPECIFIED 'TYPE' (REL,ABS,ETC.) AND COPIES THE RECORDS TO 'NEW-FILE'.

INTERACTIVE

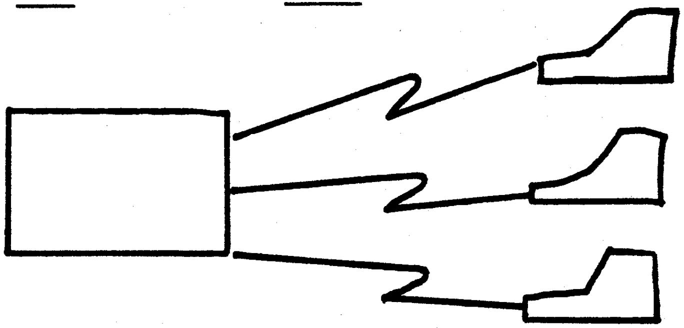


# TYPES OF INTERACTIVE PROCESSING

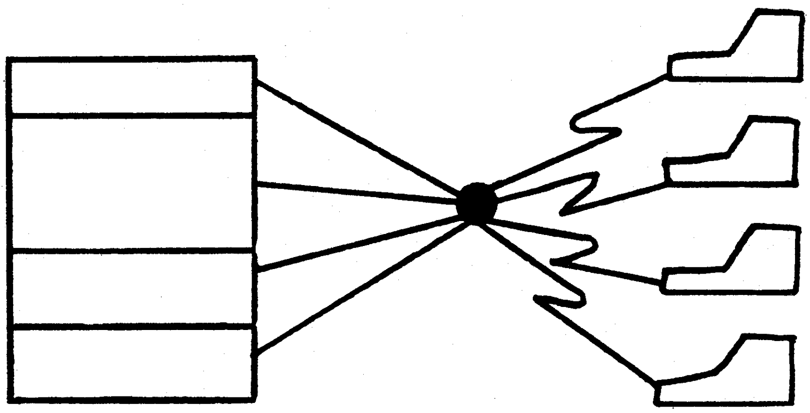
- TIME SHARING (IAF)
  ONE PROGRAM - ONE TERMINAL

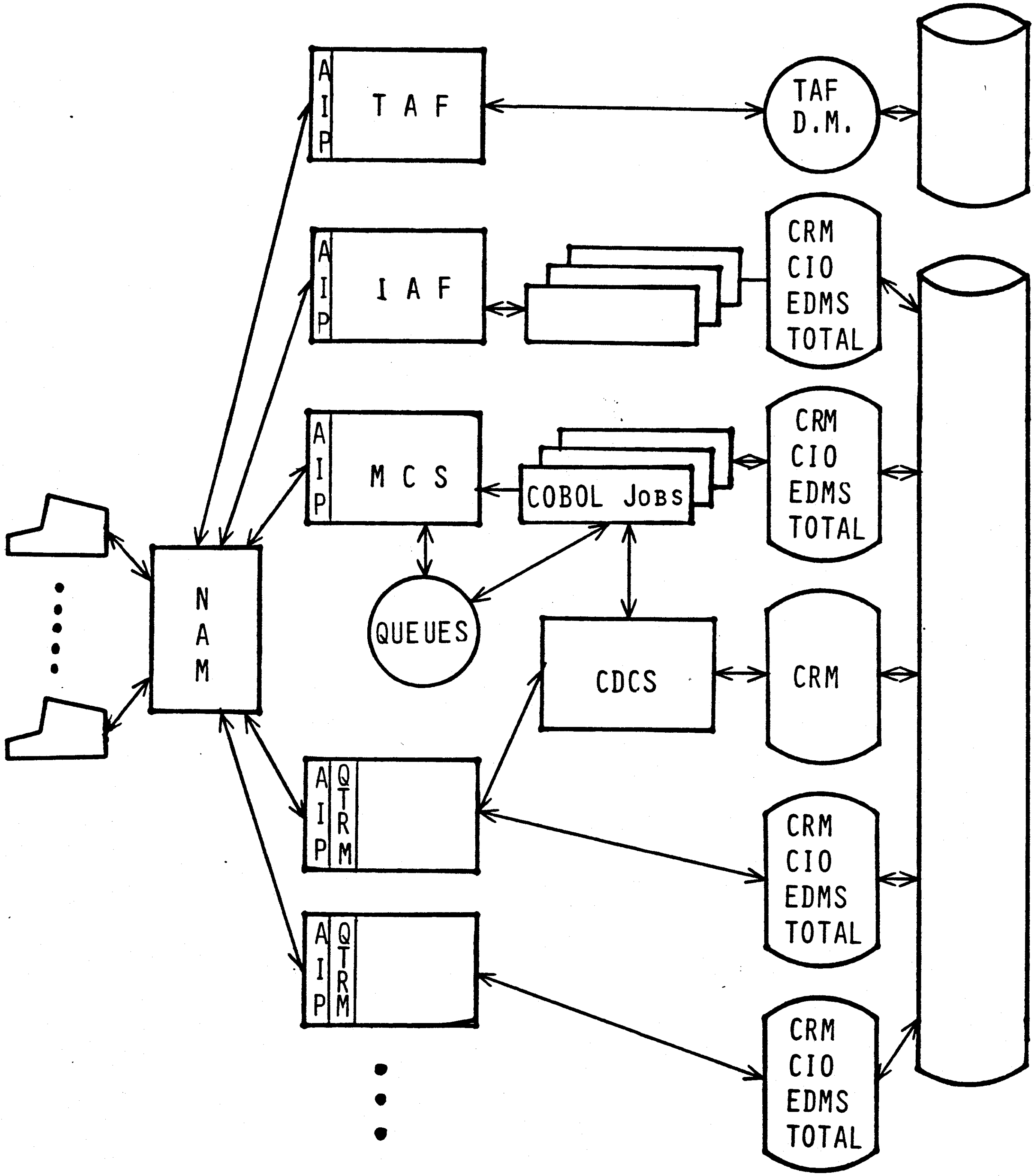


- TELEPROCESSING (QTRM)
  ONE PROGRAM - MANY TERMINALS



- TRANSACTION (TAF)
  MANY TASKS - MANY TERMINALS




AIP
TAF
TAF D.M.
AIP
IAF
CRM CIO EDMS TOTAL
AIP
MCS
COBOL Jobs
CRM CIO EDMS TOTAL
NAM
QUEUES
CDCS
CRM
AIP
QTRM
CRM CIO EDMS TOTAL
AIP
QTRM
CRM CIO EDMS TOTAL

## INTERACTIVE - TIME SHARING

## LOGGING IN

```
79/08/23   14.59.16   TM1017
(22)  SVL   SN614  NOS
FAMILY:   SVL
USER:  JFOX
PASSWORD:  NOWAY
APPLICATION: IAF
TERMINAL  13,NAMIAF
RECOVER/CHARGE: CHARGE,MYDEPT,MYPROJECT

/
```

CAN BE ABREVIATED AND ENTERED ON FAMILY LINE-

SVL,JFOX,NOWAY,IAF

IAF IS READY FOR YOU TO ENTER A COMMAND!   (BATCH SUBSYSTEM)

NOTICE THAT THE PASSWORD WAS CHANGED (PASSWOR) BETWEEN THE PREVIOUS BATCH RUN AND THIS INTERACTIVE SESSION.

## INTERACTIVE

## GENERAL

- INPUT AND OUTPUT ARE 'CONNECTED' BY DEFAULT. MOST PROGRAMS AND UTILITIES SEND DATA TO OUTPUT AND RECEIVE DATA FROM INPUT.

  ASSIGN,MS,OUTPUT WILL ASSIGN OUTPUT OR ANY FILE TO A DISK DEVICE. THIS IS LIKE THE NOS/BE DISCONT STATEMENT.

  ASSIGN,TT,OUTPUT WILL DISCARD ANYTHING CURRENTLY ON OUTPUT AND ASSIGN IT TO THE TERMINAL. THIS IS LIKE THE NOS/BE CONNECT STATEMENT.

- GENERALLY, MOST OF THE BATCH STATEMENTS WORK IN INTERACTIVE - EVEN THE MAGNETIC TAPE STATEMENTS.

- DAYFILE IS A VERY HANDY STATEMENT AND SHOULD BE LEARNED QUICKLY AND EASILY.

- THERE IS ALWAYS A PROMPT (?) WHEN INPUT IS EXPECTED FROM THE TERMINAL USER.

# INTERACTIVE

## GENERAL - CONTINUED

- ASCII (6/12) IS STANDARD IN BASIC AND XEDIT. FCOPY CONVERTS FROM 6/12 TO 8/12 ASCII.
- A CARRIAGE RETURN (CR) BY ITSELF SIGNALS AN EOF. THIS IS LIKE THE NOS/BE %EOF. IF DATA WAS NOT EXPECTED (READ STATEMENT IN PROGRESS) IAF WILL RESPOND WITH THE CURRENT STATUS (EXECUTE,IDLE,ETC.).
- ENQUIRE, OPT. ANOTHER HANDY UTILITY. IF OPT IS 'F' A STATUS OF ALL FILES IS GIVEN - LIKE FILES. 'J' IS THE JOB CONTROL REGISTER. 'R' LISTS RESOURCES USED. JN (JOB NAME) IS VERY USEFUL. THE LIST GOES ON.
- RENAME, NEWLFN=OLDLFN. CHANGES THE <u>LOCAL</u> FILE NAME. USE CHANGE TO CHANGE A PERMANENT FILE NAME.
- REWIND,*. REWINDS ALL FILES.
- RETURN,*. RETURNS ALL FILES. RETURN,OUTPUT SENDS 'OUTPUT' TO THE BIT BUCKET!

## INTERACTIVE

## SPECIAL CONSIDERATIONS

- CYBER LOADER - MUST GROUP DIRECTIVES INTO 'PACKETS' LIKE ON NOS/BE.

  - NOS/BE USES XEQ (WHAT A MESS) FOR ONE TIME EXECUTION.

  - NOS USES THE ENTER DIRECTIVE FOR ONE TIME EXECUTIONS.

    ```
    ENTER./CMD1,/CMD2./CMD3./.../
    ```

    CMD CAN BE <u>ANY</u> STATEMENT.

  - CCL PROCEDURES ARE EVEN MORE POWERFUL AND VERSATILE!

- LIBRARY - CAUTION!

  TO SPECIFY A GLOBAL LIBRARY SET USE X.LIBRARY,LFN... THE IAF 'LIBRARY' STATEMENT BEING CHANGED TO 'LIB' IN A FUTURE SYSTEM.

- DMP - ABNORMAL TERMINATION DMP IS WRITTEN TO ZZZDUMP. USE XEDIT TO PRINT IT.

  - USE DMD. ASSIGN OUTPUT TO MS.
  - BUILD INTO ENTER STATEMENT.
  - USE CCL PROCEDURE

## INTERACTIVE

### SPECIAL CONSIDERATIONS - CONTINUED

- ONLY LAST DAYFILE MESSAGE IS DISPLAYED.

  - USE FTN,A,L=0 ETC.
  - USE DAYFILE

- TYPE AHEAD

  - USE CCL
  - WAIT FOR R6

- DEFAULT CARRIAGE CONTROL IS A BLANK. SPECIAL PROCESSING (CONTROL BYTES) IS REQUIRED BUT VERY POWERFUL FOR NON-DEFAULT CARRIAGE CONTROLS.

## INTERACTIVE

## EDITORS

MANY TEXT EDITORS EXIST ON NOS. IT IS FUNNY TO WATCH PEOPLE TRY TO USE ALL OF THEM AT THE SAME TIME.

CHOOSE ONE!

LEARN IT WELL!

USE IT!

THREE OF THE EDITORS INCLUDE -

- NOS LINE EDITOR - GREAT FOR THE OPERATING SYSTEM
- EDIT
- XEDIT

## INTERACTIVE

### EDITORS - CONTINUED

I CHOOSE XEDIT. IT CAN DO JUST ABOUT EVERYTHING THE NOS/BE EDITOR CAN DO PLUS SOME VERY POWERFUL NEW COMMANDS (COPY, READ, MODIFY AND INTERNAL PF INTERFACE). I MADE TWO COMPROMISES:

- XEDIT IS A POINTER BASED EDITOR. IT TOOK ME ABOUT 2 WEEKS TO 'FORGET' LINE NUMBERS.

- XEDIT COMMANDS CANNOT BE INTERMIXED WITH OPERATING SYSTEM COMMANDS.

## INTERACTIVE

### EDITORS - continued

Learn XEDIT commands in phases. Don't go onto second phase until you are comfortable with first phase.

- INSERT
  DELETE
  PRINT
  LOCATE
  CHANGE
  NEXT
  TOP
  BOTTOM
  END
  STOP
  MODIFY

- DEFTAB
  TABS
  LISTAB
  COPY
  READ
  INSERTB
  REPLACE

- QMOD
  RMARGIN
  DEOR
  TRIM
  WEOR
  WEOF
  DEOF

- Ignore the 'obtuse' commands!

Directives can be placed on the XEDIT control statement. Place the XEDIT control statement in a CCL procedure tailored to each users needs (i.e. tabs, bells, etc.).

## XEDIT

POINTER

ANT

BEAR

CAT

DOG

ELEPHANT

FOX

GOOSE

### POINTER MOVEMENT

TOP
BOTTOM
NEXT
LOCATE

### MODIFICATION

CHANGE
DELETE
INSERT

### DISPLAY

PRINT

### TERMINATION

END

STOP

## INTERACTIVE

## EDITORS - CONTINUED

SPECIAL CONSIDERATIONS

- USE XEDIT IN BATCH
- USE XEDIT FOR PAGE FUNCTIONS
- DON'T USE LINE NUMBERS! THEY ARE A PAIN!
- XEDIT MAKES A LOCAL FILE AND A PERMANENT FILE AT THE SAME TIME.

## REMOTE BATCH

REMOTE BATCH IS ACCOMPLISHED BY RBF. IT HAS COMMANDS SIMILAR TO INTERCOM'S REMOTE BATCH. THE COMMANDS OF IAF AND RBF ARE PROCESSED BY SEPARATE FACILITIES. INTERCOM MERGES THE COMMANDS OF BOTH FACILITIES TOGETHER. RBF COMMANDS INCLUDE -

| | |
|---|---|
| DISPLAY | JOB,IN,PR,PU,DEV,ETC. |
| CHANGE | PRI,REP |
| DIVERT | JOB,IN,PR,PU,EX, ETC.<br>FAM,USR,HST |
| PURGE | JOB,IN,PR,PU,EX, ETC. |
| SKIP | ±N,DFL,END |
| REWIND | |
| RETURN | |

## REMOTE BATCH

## SPECIAL CONSIDERATIONS

- THE RBF BATCH RELATED COMMANDS ARE NOT AVAILABLE IN IAF.

  - VALIDATE USERS FOR IAF AND RBF. THE USER CAN SWITCH BETWEEN IAF AND RBF TO DO THE RBF QUEUE MANIPULATION COMMANDS.

- CAN'T MOVE A PRINT FILE FROM THE PRINT QUEUE TO A LOCAL FILE.

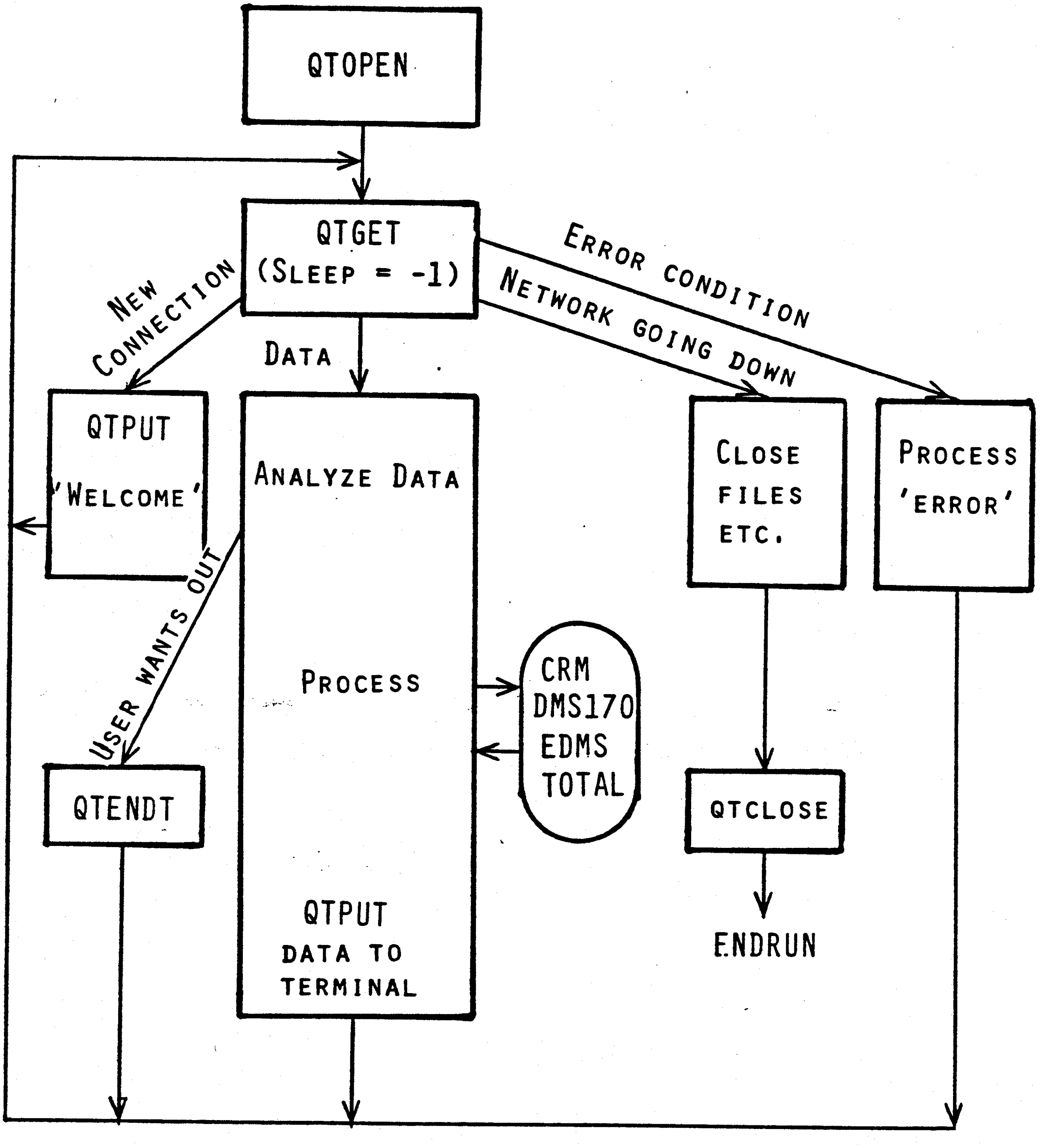
  - STANDARD CODE WILL BE AVAILABLE (1Q80?).

    QGET<br>
    QDROP<br>
    QPURGE

## TELEPROCESSING (MUJ)

TELEPROCESSING (ONE PROGRAM COMMUNICATING WITH MANY TERMINALS) IS ACCOMPLISHED BY NAM/AIP/QTRM. SUPPLY THE PROGRAMMER(S) WITH A NAM APPLICATION NAME AND VALIDATION FOR THAT NAME AND LET THEM HAVE THEIR FUN. QTRM WAS DESIGNED FOR COBOL AND FTN PROGRAMMERS. CALLS INCLUDE:

- QTOPEN - IDENTIFIES PROGRAM (APPLICATION) TO NAM.
- QTPUT - SENDS DATA TO A SPECIFIED TERMINAL CONNECTION.
- QTGET - SCANS TERMINALS FOR INPUT FROM THE TERMINAL USERS.
- QTENDT - TERMINATE (NORMAL OR FORCED) A SPECIFIED TERMINAL CONNECTION.
- QTCLOSE - TERMINATES COMMUNICATION BETWEEN THE PROGRAM AND NAM (ALL TERMINALS TO THIS PROGRAM).


QTOPEN
QTGET
(SLEEP = -1)
NEW CONNECTION
DATA
ERROR CONDITION
NETWORK GOING DOWN
QTPUT
'WELCOME'
ANALYZE DATA
PROCESS
QTPUT
DATA TO
TERMINAL
USER WANTS OUT
QTENDT
CRM
DMS170
EDMS
TOTAL
CLOSE
FILES
ETC.
QTCLOSE
ENDRUN
PROCESS
'ERROR'

## EASING THE TRANSITION

- USE THE MIGRATION AS AN OPPORTUNITY TO REVIEW CURRENT SYSTEMS, OPERATIONAL PROCEDURES, ETC.

- START NOW - USE COMMANDS THAT ARE COMMON ON BOTH SYSTEMS. I.E. USE ROUTE NOT BATCH OR DISPOSE.

- START NOW - USE CCL PROCEDURES FOR MUNDANE TASKS. CHANGE CONTENT OF PROCEDURE, IF NECESSARY, WHEN YOU CHANGE SYSTEMS.

- DON'T TRY TO USE ALL OF THE UTILITIES. A SMALL NUMBER WILL FILL MOST USERS NEEDS. SOME 'OLD' UTILITIES EXIST FOR HYSTERICAL REASONS (IN ADDITION TO HISTORICAL REASONS).

- LEARN THE UTILITIES A LITTLE EACH WEEK.

- CHOOSE GENERAL PURPOSE COMMANDS OVER SINGLE PURPOSE COMMANDS.

- COMPARE TASKS. DON'T COMPARE FEATURES.

- IDENTIFY CURRENT COMPASS MACRO USE. PLAN YOUR ATTACH. CHECK INTO COMMON COMMON DECKS.

## EASING THE TRANSITION - CONTINUED

- PLAN FOR ZERO LOCAL MODS.
- GET ACCOUNTING READY. SOME JOB COSTS WILL GO UP WHILE OTHERS WILL GO DOWN.
- COMPUTER POLICY WILL CHANGE. INFORM USERS.
- TRADE SOMETHING (NEW FEATURES, NEW HARDWARE,ETC.)
- PEOPLE ARE STILL AFRAID OF THE UNKNOWN.
- THE OPERATING SYSTEM MIGRATION WILL BE EASIER THAN THE CONVERSION FROM SCOPE 3.3 → SCOPE 3.4.
- OPPORTUNITY TO REVIEW OVERALL SYSTEMS, OPERATIONAL PROCEDURES, ETC.

## INSTRUCTOR GUIDE

## TABLE OF CONTENTS

# FOREWORD

This course was developed for Control Data Education Company of Control Data Corporation. It presents usage information for the computer user who desires to convert from NOS/BE to NOS. It includes a history of NOS/BE and NOS, characteristics of NOS and system access. The main part of the course contains detailed information on how to do NOS/BE tasks on NOS such as file retention, magnetic tape usage, source usage, object usage and interactive usage (time sharing, teleprocessing and remote batch). The course concludes with suggestions on how to ease the migration.

This instructor guide aids instructors in performing their duties. This document contains course objectives, resource data and other helpful documentation that should be incorporated by the instructor to assure that at the end of this course, the student has been given all information possible.

Due to its nature, this course requires an instructor who is well qualified in the usage of both NOS/BE and NOS. NOS is changing very rapidly as a result of migration concerns. The 'special considerations' sections are especially subject to change. The NOS/BE to NOS Migration office in Arden Hills should be contacted for current information and strategies.

# NOS/BE to NOS USAGE DIFFERENCES

LENGTH: 3 DAYS

This course provides usage information for the computer user who desires to convert from NOS/BE to NOS. It includes a history of NOS/BE and NOS, characteristics of NOS and system access. The main part of the course contains detailed information on how to do NOS/BE tasks on NOS such as file retention, magnetic tape usage, source usage, object usage and interactive usage (time sharing, teleprocessing and remote batch). The course concludes with suggestions on how to ease the migration.

The student should currently be a programmer/analyst on NOS/BE. The course presumes a moderate level of understanding and usage experience with NOS/BE batch and interactive.

# MATERIALS LIST

## INSTRUCTOR MATERIAL

| | |
|---|---|
| Instructor Guide | |
| Student Handout | |
| Visual Package | |
| NOS Reference Manual Vol. 1 | 60435400 |
| NOS Reference Manual Vol. 2 | 60445300 |
| Interactive Facility (IAF) Reference Manual | 60455250 |
| Remote Batch Facility (RBF) Reference Manual | 60499600 |
| XEDIT Reference Manual | 60455730 |

## STUDENT MATERIAL

| | |
|---|---|
| Student Handout | |
| NOS Reference Manual Vol. 1 | 60435400 |
| NOS Reference Manual Vol. 2 | 60445300 |
| Interactive Facility (IAF) Reference Manual | 60455250 |
| Remote Batch Facility (RBF) Reference Manual | 60499600 |
| XEDIT Reference Manual | 60455730 |

## EQUIPMENT REQUIREMENTS

Access to a current version of NOS. Interactive and batch are required. Remote batch is an option and a satisfactory substitute for central batch. Refer to the course chart for detailed requirements.

## TRAINING AIDS

Overhead Projector
Screen
Chalkboard

# NOS/BE TO NOS USAGE DIFFERENCES

## COURSE CHART

| HOUR | DAY 1 | DAY 2 | DAY 3 |
|---|---|---|---|
| 1 | ADMINISTRATIVE DETAILS<br>NOS HISTORY<br>NOS CHARACTERISTICS | REVIEW<br>MAGNETIC TAPES | REVIEW<br>TEXT EDITORS |
| 2 | SYSTEM ACCESS | SOURCE USAGE/MAINTENANCE<br>OBJECT USAGE, STORAGE, MAINTENANCE | LABORATORY FOR TIME SHARING AND TEXT EDITING |
| 3 | BATCH ACCESS | | |
| 4 | PERMANENT FILES | LABORATORY - SOURCE AND OBJECT USAGE AND MAINTENANCE | REMOTE BATCH<br>TELEPROCESSING (QTRM) |
| 5 | LABORATORY - BATCH ACCESS AND PERMANENT FILES | INTERACTIVE OVERVIEW | TRANSACTION PROCESSING<br>HINTS TO EASE THE TRANSITION |
| 6 | | TIME SHARING | CRITIQUE |

# COURSE OUTLINE
# NOS/BE to NOS USAGE DIFFERENCES

1. NOS History and Philosophy (½ hr.)
2. NOS Characteristics (½ hr.)
    - A. Validation
    - B. Security
    - C. Ownership of Resources
    - D. Simplicity
3. System Access (1 hr.)
    - A. General
    - B. Families
    - C. User Indicies
4. Batch (½ hr.)
    - A. Access
    - B. File Structure
    - C. Special Considerations
5. Permanent Files (1½ hr.)
    - A. Ownership
    - B. Types
    - C. Categories
    - D. Access Modes
    - E. Commands
    - F. Special Considerations
6. Magnetic Tape (½ hr.)
    - A. Reservations
    - B. Assignment
    - C. Release
7. Source (½ hr.)
    - A. Storage Techniques
    - B. Usage
    - C. Maintenance

8. Object Module (1½ hr.)

   A. Storage Techniques
   B. Usage
   C. Maintenance
   D. Special Considerations

9. Interactive Overview (½ hr.)

10. Time Sharing (1 hr.)

    A. Access
    B. General
    C. Special Considerations

11. Text Editors (½ hr.)

12. Remote Batch (½ hr.)

13. Teleprocessing (½ hr.)

14. Transaction Processing

15. Easing the Transition (1 hr.)

# LESSON GUIDE 1
# NOS HISTORY AND PHILOSOPHY

## LESSON PREVIEW

This lesson introduces the student to the course, instructor, history of CYBER operating system and NOS characteristics.

## REFERENCES

None

## TRAINING AIDS

Visuals V1-1 through V1-7

## ACTIVITIES

## OBJECTIVES

1. Define course context.
2. Set tone for the course.
3. Describe NOS History and basic Philosophy.
4. Emphasize tasks not machine cycles, bits, etc.

# LESSON 1 OUTLINE

## NOS HISTORY AND PHILOSOPHY

VISUALS TOPICS

V1-1 Discuss major topics for this course.

Emphasis is on tasks not control statement parameters.

V1-2 Tailor this slide to your background.

Explain that your excellent background makes you an ideal candidate to teach this class.

V1-3 A. Describe the audience the class was designed for.

1. System Usage

   Most of the people at the site and the subject of this class.

2. Systems Analysts

   a. They need to attend this class.

   b. They also need to attend a NOS System intervals class.

3. Operators

   a. They do not need to attend this class.

   b. They do need to attend a NOS operator class

4. Discuss the existence of the Migration office and the existence of conversion aids (to be discussed later).

V1-4 B. History of NOS

1. Both NOS/BE and NOS evolved from Chippewa - similar logical file structure.

# LESSON 1 OUTLINE

## NOS HISTORY AND PHILOSOPHY (continued)

| VISUALS | TOPICS |
|---|---|
|  | 2. Common Products have been common since 1973. This is the primary need of system users. |
| V1-5<br>V1-6 | 3. Emphasize that computers are used for 'real' applications not internal functions of the operating system. |
| V1-7 | 4. List needs of the user applications noting how much is already common. |

# NOS/BE TO NOS USAGE MIGRATION

## CONTENT

- BRIEF HISTORY AND ARCHITECTURE OF NOS
- SYSTEM ACCESS
- PERMANENT FILES
- MAGNETIC TAPES
- SOURCE STORAGE, USAGE AND MAINTENANCE
- OBJECT STORAGE, USAGE AND MAINTENANCE
- INTERACTIVE, REMOTE BATCH AND TELEPROCESSING USAGE
- SOURCE EDITORS
- PLANNING FOR THE TRANSITION

## WHY ME?

- CYBER EXPERIENCE STARTED ON SCOPE 3.1.2 IN 1969.

- RECENT CONVERSION EXPERIENCE.

  PERSONALLY CONVERTED IN 1978.

  EMPATHY FOR NOS/BE MIGRATION CONCERNS.

- PRODUCT SET FIELD SUPPORT

  EXPOSURE TO NOS, SCOPE 2 AND NOS/BE USER PROBLEMS.

- USE BOTH SYSTEMS DAILY.

## NOS/BE TO NOS USAGE MIGRATION

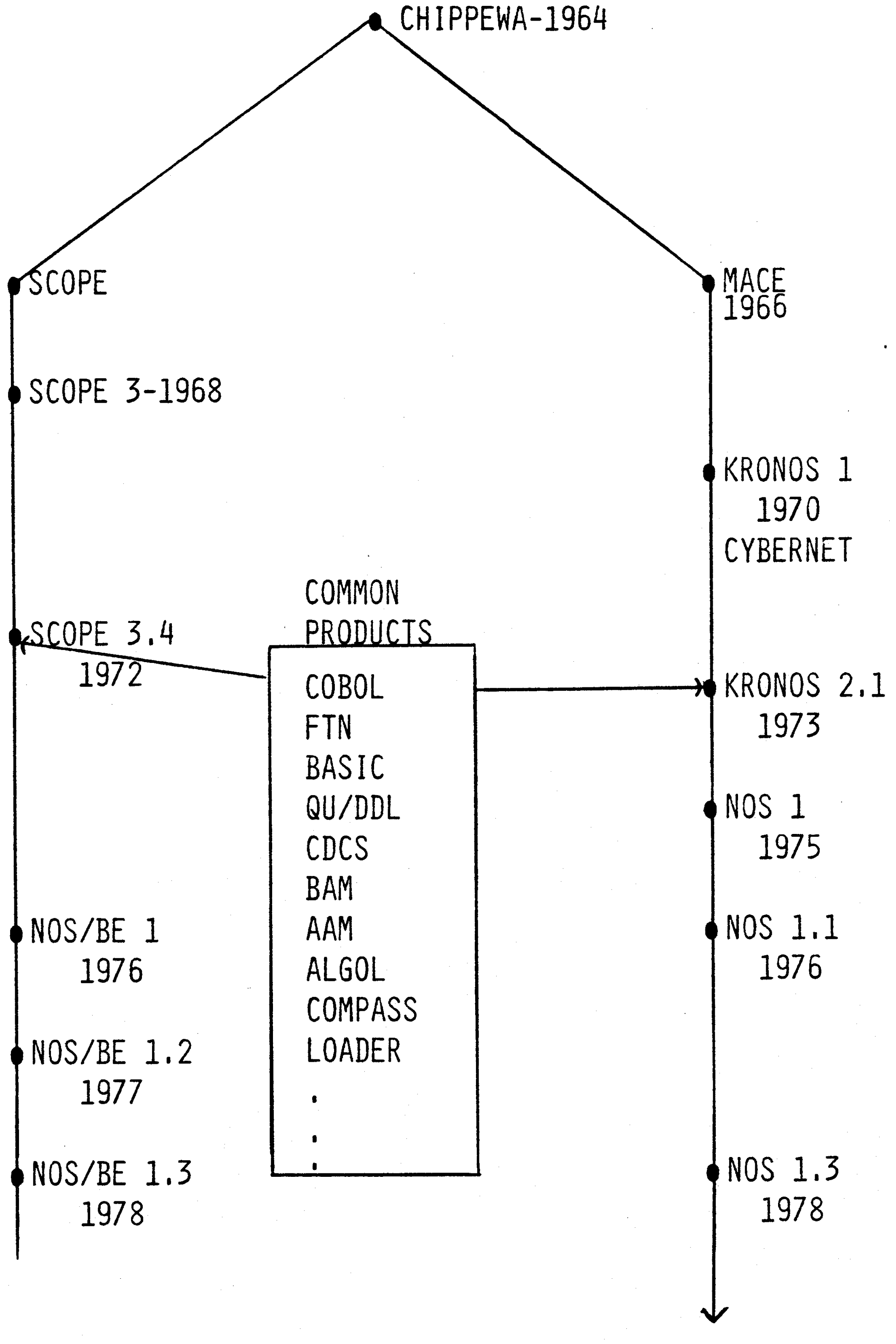
- SYSTEM USAGE - 98% OF PEOPLE AT A SITE.

  THE SUBJECT OF THIS SESSION.

- SYSTEMS ANALYST - 1% OF PEOPLE AT A SITE.

- OPERATOR TRAINING - 1% OF PEOPLE AT A SITE.

- CONVERSION AIDS


CHIPPEWA-1964
SCOPE
SCOPE 3-1968
SCOPE 3.4
1972
NOS/BE 1
1976
NOS/BE 1.2
1977
NOS/BE 1.3
1978
MACE
1966
KRONOS 1
1970
CYBERNET
KRONOS 2.1
1973
NOS 1
1975
NOS 1.1
1976
NOS 1.3
1978
COMMON
PRODUCTS
COBOL
FTN
BASIC
QU/DDL
CDCS
BAM
AAM
ALGOL
COMPASS
LOADER
.
.
.

## COMPUTER IS USED FOR-

- AUTOMOTIVE
- BANKING
- CONSTRUCTION
- EDUCATION
- PETROLEUM
- WEATHER
- ETC.

<u>COMPUTER IS NOT USED FOR</u> -

- CPU MTR
- FANCY OPERATOR DISPLAYS
- BUILDING SYSTEMS
- ETC.

CDC COMPUTERS ARE FOR <u>SOLUTIONS</u> NOT MACHINE <u>CYCLES</u>.

APPLICATIONS REQUIRE -

- COMMON PRODUCTS - COBOL, FTN, CRM, ETC.

- VEHICLE TO RUN COMMON PRODUCTS

- "SERVICE" FACILITIES

  FILE RETENTION
  ACCOUNTING
  ETC.

# LESSON GUIDE 2
# NOS CHARACTERISTICS

## LESSON PREVIEW

This lesson describes the characteristics of NOS. These characteristics help explain why the operating system is put together the way it is.

## REFERENCES

None

## TRAINING AIDS

Visual V2-1

## ACTIVITIES

## OBJECTIVES

1. Define responsibilities of user, operator and operating system.

2. Describe resource ownership, validation, etc.

# LESSON 2 OUTLINE

## NOS CHARACTERISTICS

VISUALS TOPICS

The following NOS characteristics help explain why the operating system and control statements function in the manner that they do.

V2-1

A. All users must be validated.

   1. Avoids waste of resources.

   2. Protects the users resources.

B. The user controls the value of a super password that controls access to the user's resources.

C. All resources belong to a user on NOS. On NOS/BE resources belong to the operating system (PFs, etc.).

D. There is generally a one-to-one relationship between a 'person' and a NOS 'user'.

E. Simplicity

   1. More control statements.

   2. Fewer parameters on each control statement.

F. Operator does not make user decisions.

## NOS CHARACTERISTICS

- SYSTEM CONTROL THRU VALIDATION

  AVOID WASTE OF RESOURCES.
  PROTECTION OF USER'S RESOURCES.

- SECURITY - ACCESS TO PFS, ACCOUNTING, ETC., IS THRU A USER CONTROLLED PASSWORD.

- EVERYTHING BELONGS TO A 'USER'.

- RELIABILITY - CURRENT FAILURE RATE IS LESS THAN 1 PER MONTH. RESULT OF MAINTAINABILITY IMPROVEMENTS THROUGHOUT THE ENTIRE PSR LEVEL 400 SERIES.

- SIMPLICITY - MORE COMMANDS; FEWER PARAMETERS. VERY FEW 'INPUT' DIRECTIVES.

- RESOURCE OWNERSHIP - THERE IS A 1 TO 1 RELATIONSHIP BETWEEN PEOPLE AND USER NUMBERS.

- OPERATOR DOES NOT MAKE USER DECISIONS.

- MANY WAYS TO DO A TASK.

  CHOOSE ONE - I'LL HELP.

# LESSON GUIDE 3
# SYSTEM ACCESS

## LESSON PREVIEW

This lesson describes how the user gains access to NOS batch, remote batch, interactive, etc. The lesson also describes the FAMILY/USER concept.

## REFERENCES

NOS RM Ch. 6 - USER statement.

## TRAINING AIDS

Visuals V3-1 through V3-5

## ACTIVITIES

## OBJECTIVES

1. Understand the concept of a FAMILY, user index and user name.
2. Understand how files are allocated to physical devices.
3. Review basic file structure.

## LESSON 3 OUTLINE

## SYSTEM ACCESS

VISUALS — TOPICS

A. The USER statement is or controls access to -

V3-1 — 1. Required for any access.

- a. Batch
- b. Interactive
- c. Remote Batch
- d. Teleprocessing
- e. Transaction Processing

2. Controlled by a user defined password -

3. Permanent files.

4. Resource Limits (MT, NT, PF size, CP, etc.)

5. Billing

V3-2 — B. Family -

1. Mechanisms to group user communities together.

2. Is a vehicle to make system backup easier.

3. Contains the validation for users in the family.

C. User index.

1. Techniques to spread users evenly over the devices in a family.

2. Access to PF Catalog.

3. Indirect PFs.

4. Some Direct PFs.

## LESSON 3 OUTLINE

## SYSETM ACCESS (continued)

| VISUALS | TOPICS |
|---|---|
| V3-3,4 | D. Auxiliary devices. |
| | 1. Private - who can access. |
| | 2. Public - who can access. |
| | E. Logical devices - multiple spindle device and impact on 'track' size, large files, etc. |
| V3-5 | F. File structure. |
| | 1. Representation of EORs, EOFs, etc. on various devices. |
| | 2. Random file problem on NOS/BE. |

## SYSTEM ACCESS

### USER STATEMENT

- COMMON AND REQUIRED FOR ALL ACCESS

  BATCH
  INTERACTIVE
  REMOTE BATCH
  TELEPROCESSING
  TRANSACTION

- PASSWORD PROTECTION CONTROLLED BY USER

- PERMANENT FILE ACCESS

- RESOURCE LIMITS

- BILLING (ACCOUNTING)

- SECONDARY USER STATEMENTS CAN BE ENTERED IN THE MIDDLE OF A SESSION (SITE MUST CHOOSE THIS OPTION). THIS FEATURE IS PRIMARILY FOR PERMANENT FILE ACCESS.

# FAMILIES AND USER INDICES

- FAMILY IS THE TECHNIQUE TO GROUP MEMBERS OF USER COMMUNITIES TOGETHER.

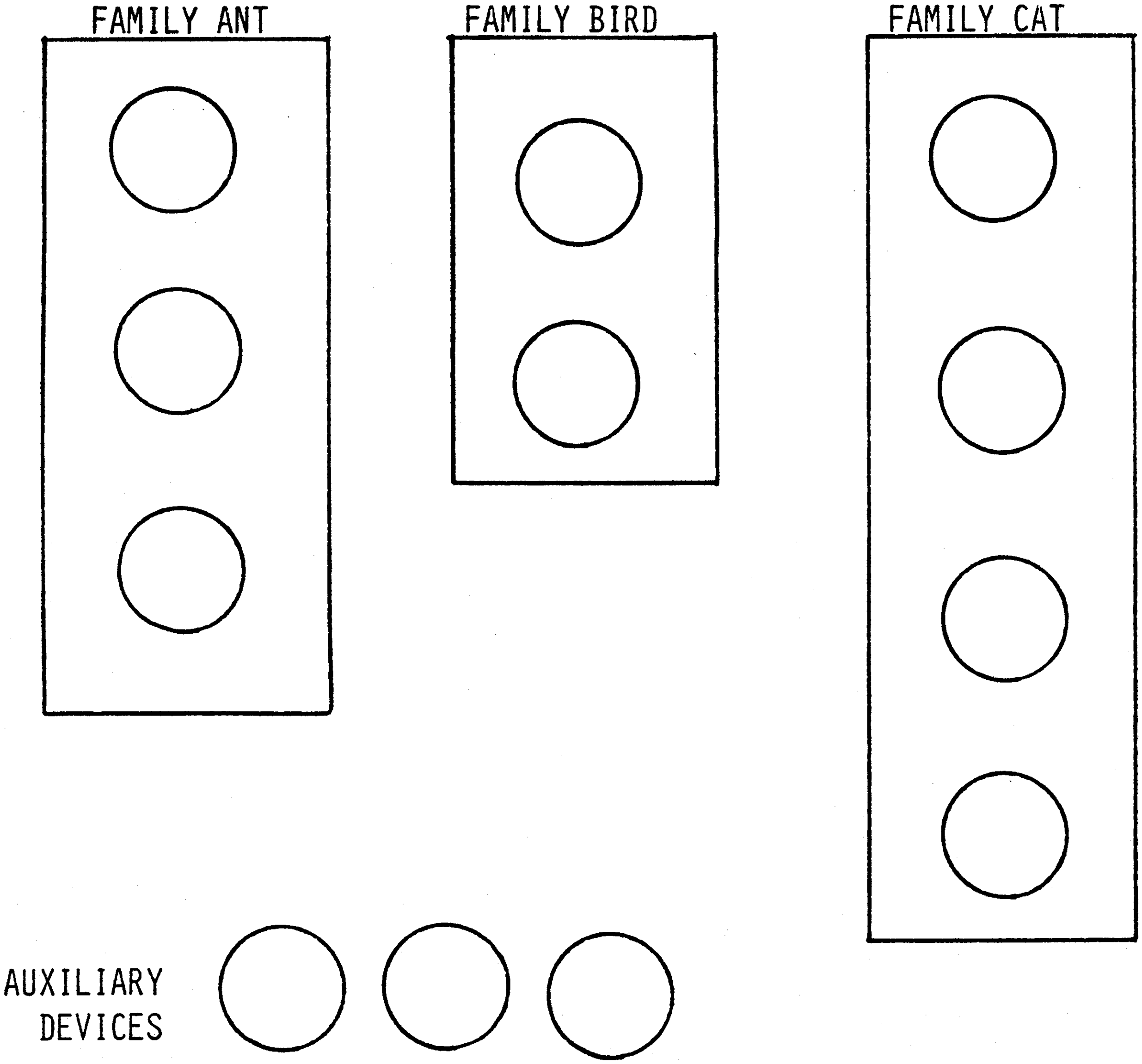
  CONTAINS THE VALIDATION FILE FOR USERS IN THAT FAMILY.

  PROVIDES A BACKUP FACILITY SINCE IT IS VERY EASY TO MOVE USERS FROM A DOWN MACHINE.

- USER INDEX IS THE TECHNIQUE TO SPREAD THE USERS IN A FAMILY OVER THE DISKS IN A FAMILY. THE DISK THAT A USER INDEX IS 'ASSIGNED' CONTAINS THAT USER'S -

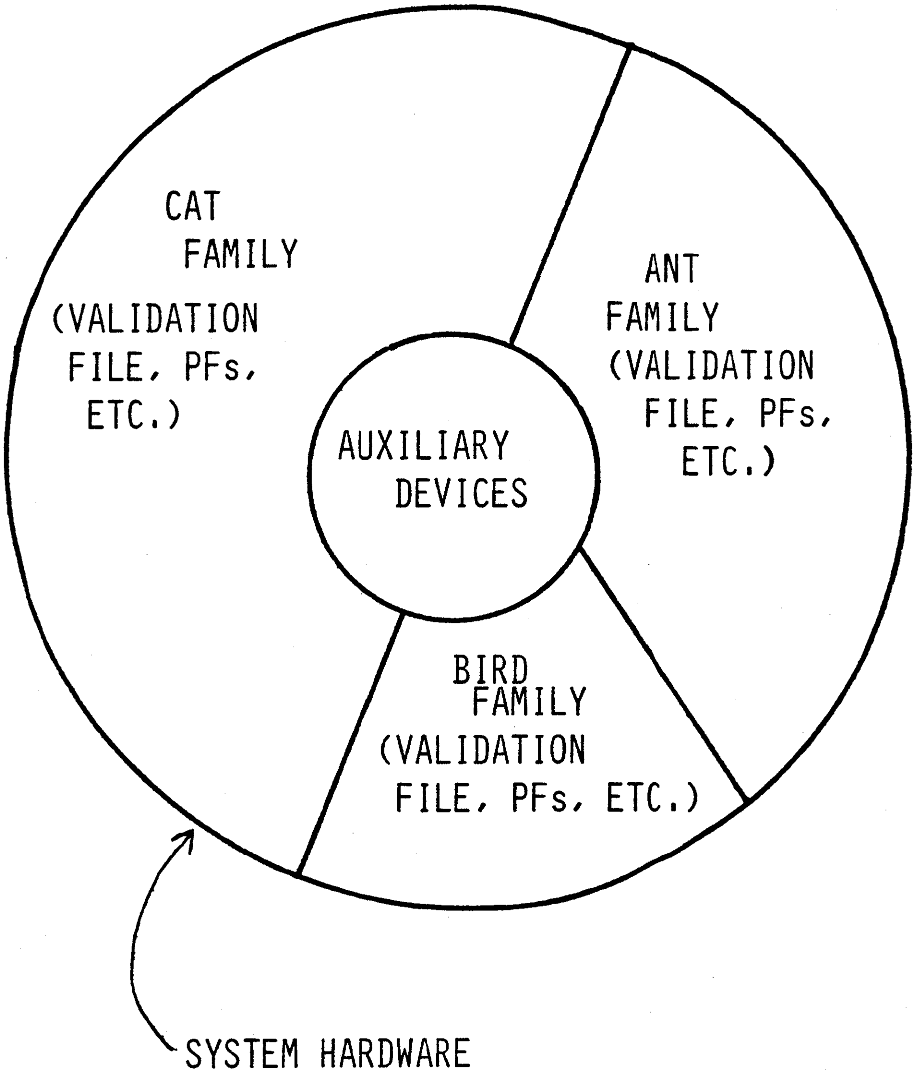
  PF CATALOG
  INDIRECT PFs
  SOME DIRECT PFs

V3-3



EACH 'DEVICE' WITHIN A FAMILY OR AUXILIARY DEVICE MAY BE 1 TO 8 DISK SPINDLES.
EACH AUXILIARY DEVICE MAY BE DECLARED

-PRIVATE - ONLY VALIDATED USER CAN CREATE FILES.
ALTERNATE USERS CAN BE PERMITTED TO ACCESS FILES
-PUBLIC - ANY USER CAN CREATE OR ACCESS FILES



- HARDWARE
- FAMILY
- USER INDEX
- FILES
- AUXILIARY DEVICES

## FILE STRUCTURE

ESSENTIALLY THE SAME.

NOS/BE

| | EOR | EOF | EOI |
|---|---|---|---|
| CARDS | 7/8/9 | IMPOSSIBLE | 6/7/8/9 |
| DISK | SHORT PRU + LEVEL=0 | ZERO LENGTH PRU + LEVEL = 17 | END OF RBR CHAIN |
| TAPE (SI) | THE SAME AS DISK. | THE SAME AS DISK. | DOUBLE TAPEMARKS |

NOS

| | EOR | EOF | EOI |
|---|---|---|---|
| CARDS | 7/8/9 | 6/7/9 | 6/7/8/9 |
| DISK | SHORT PRU + LINK TO NEXT PRU. | ZERO LENGTH PRU + LINK TO NEXT PRU | ZERO LENGTH PRU<br>NO LINK |
| TAPE (SI/I) | SHORT PRU + LEVEL = 0 | ZERO LENGTH PRU + LEVEL = 17 | DOUBLE TAPEMARKS |

- THERE ARE NO RANDOM BIT PROBLEMS ON NOS. ANY FILE (RANDOM OR SEQUENTIAL) MAY BE COPIED WITH THE COPYEI DIRECTIVE

# LESSON GUIDE 4

## BATCH ACCESS/PROCESSING

### LESSON PREVIEW

This lesson introduces the student to batch processing (job structure) on NOS.

### REFERENCES

NOS Reference Manual Chapter 5.

### TRAINING AIDS

Visuals V4-1 through V4-4

### ACTIVITIES

### OBJECTIVES

1. Describe differences between NOS/BE and NOS job processing.
2. Describe Job naming conventions on NOS.
3. How to cope with deficiencies on NOS (job dependency, job names, etc.).

# LESSON 4 OUTLINE

## BATCH ACCESS/PROCESSING

| VISUALS | TOPICS |
|---|---|
| V4-1 | A. Job Statement |
| | 1. Time limit - for the job step, not entire job. |
| | 2. CM |
| | 3. EC |
| | 4. Remainder of NOS/BE parameters are specified in the job stream when needed. |
| V4-2 | B. Job Name |
| | 1. User hash. |
| | 2. System ordinal. |
| | C. USER statement |
| | 1. User name. |
| | 2. Password - Maintenance by the user. |
| | 3. Family of the validation file. |
| | D. CHARGE |
| V4-3,4 | E. Special Considerations |
| | 1. RERUN/NORERUN - Controlled by the user. |
| | 2. EXIT/NOEXIT/ONEXIT - Controlled by the user. |
| | 3. Job Name - How to cope with it. |
| | 4. Job Dependencies - How to simulate on NOS. |

## BATCH ACCESS

```
FOX,T40.
USER,JFOX,WEIRD [,FAMILY].
CHARGE,MYDEPT,MYPROJECT.
:
```

- JOB STATEMENT - ONLY 3 PARAMETERS!

  T - TIME LIMIT IN CP SECONDS FOR EACH JOB STEP.

  CM - SPECIFIES MAXFL. NOT RECOMMENDED. MAXFL MUST SATISFY -

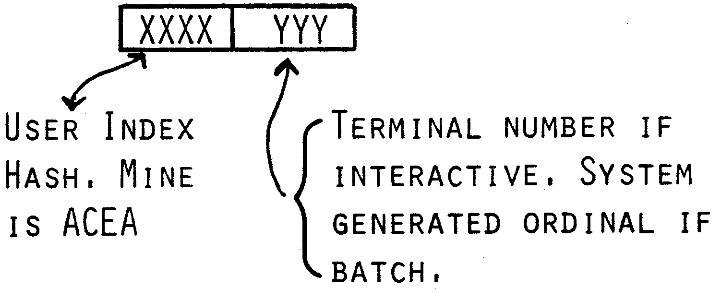
  RFL ≤ MFL ≤ MAXFL ≤ VALIDATED MAXIMUM.

  EC - EXTENDED CENTRAL STORAGE

IN NOS MANY OF THE NOS/BE JOB PARAMETERS ARE SPECIFIED BY CONTROL STATEMENTS WITHIN THE JOB WHEN NEEDED. THERE ARE GENERALLY MORE CONTROL STATEMENTS ON NOS WITH FEWER PARAMETERS.

## BATCH ACCESS - CONTINUED

- JOB NAME IS UNIQUE AND CONSISTS OF TWO PARTS-



XXXX | YYY

USER INDEX HASH. MINE IS ACEA

TERMINAL NUMBER IF INTERACTIVE. SYSTEM GENERATED ORDINAL IF BATCH.

- FAMILY IS VALIDATION FILE TO CHECK FOR USER NAME.
- USER NAME (JFOX) IS THE OPERATIONS SUPPLIED NAME IN THE VALIDATION FILE. THIS IS ASSOCIATED WITH A USER INDEX THAT DETERMINES WHERE MY PF CATALOG AND FILES RESIDE.
- PASSWORD (WEIRD) IS SUPPLIED, CHANGED AND MAINTAINED BY ME (THE USER). I CAN CHANGE IT WHEN APPROPRIATE.
- DEPARTMENT (MYDEPT). EACH USER CAN BE VALIDATED TO CHARGE TO CERTAIN DEPARTMENTS AND PROJECTS WITHIN DEPARTMENTS.

## BATCH ACCESS

### SPECIAL CONSIDERATIONS

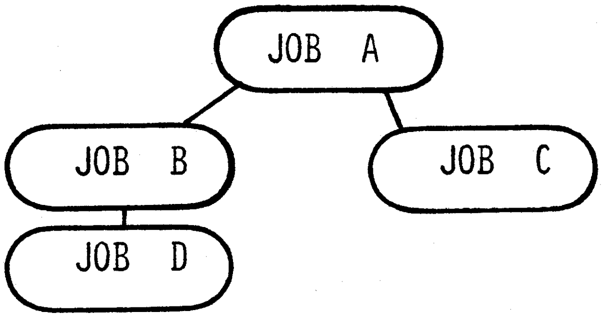
- RERUN NORERUN - User can explicitly turn rerun flag on or off as desired in job. No strange PF tricks like on NOS/BE.

- EXIT NOEXIT ONEXIT - No EXIT(s) but facility to turn EXIT globally on or off

# BATCH ACCESS

## SPECIAL CONSIDERATIONS

- JOB NAME - CODE IS AVAILABLE VIA THE UNDERGROUND. 'JOB ADMINISTRATION' ADDRESSES THE ENTIRE PROBLEM AND IS SCHEDULED FOR A FUTURE RELEASE.



- JOB DEPENDENCIES - NO JOB STATEMENT PARAMETER. CAN USE ANY OF THE FOLLOWING AND AVOID WASTED NOS/BE FNT ENTRIES AND READING JOBS IN BACKWARD.

  - ROUTE OR LDI COMBINED WITH CCL IS A VERY POWERFUL COMBINATION.

  - PF INTERLOCKS COMBINED WITH ROLLOUT.

  - SPECIAL PRODUCTS LIBRARY.

    WAIT, RESUME, LISTDJ, DROP

# LESSON GUIDE 5

# PERMANENT FILES

## LESSON PREVIEW

This lesson presents the permanent file concepts of NOS to the student.

## REFERENCES

NOS Reference Manual Chapter 8.

## TRAINING AIDS

Visual V5-1 through V5-14

## ACTIVITIES

Prepare for laboratory session to reinforce batch access and permanent file processing.

## OBJECTIVES

1. Establish firm base of understanding permanent file concepts - ownership, types, categories, access modes and commands.

2. Provide work-arounds for NOS deficiencies (40 character file name, cycles, user dump/load, etc.).

## LESSON 5 OUTLINE

## PERMANENT FILES

| VISUALS | TOPICS |
|---|---|
| V5-1 | Discuss global architecture of NOS PFs. |
| V5-2 | A. Ownership - PFs always belong to one and only one user. |
| | 1. Alternate users may access. |
| | B. Types |
| | 1. Indirect |
| | a. User gets a copy of the file. |
| | b. Basic unit of storage is PRU. |
| | 2. Direct |
| | a. User uses real file. |
| | b. Basic unit of storage is a track. |
| V5-3 | C. Categories - Expands ownership concepts. |
| | 1. Public - Any user can access. |
| | 2. Private - Specific alternate users must be PERMITted. |
| | 3. Semi-Private - Like Public except that a list of who accessed the file is maintained. |
| V5-4 | D. Access Modes - General concepts. |
| | (N,E,R,RA,RM,A,M,W) |
| V5-5 | E. Commands |
| | 1. Both Direct and Indirect commands. |
| | 2. Direct only commands. |
| | 3. Indirect only commands. |

# LESSON 5 OUTLINE

## PERMANENT FILES (continued)

| VISUALS | TOPICS |
|---|---|
| V5-6,7 | F. Direct Files |
| | 1. Creation |
| | a. Commands/Syntax |
| | b. Category impact. |
| | c. Relationship between category and mode. |
| V5-8 | 2. Access |
| | a. Commands |
| | b. Access mode. |
| | c. Exceeding authorized mode - pfn not found. |
| V5-9 | G. Indirect Files |
| | 1. Creation. |
| V5-10 | 2. Access. |
| | 3. Extension |
| V5-11 | H. Special Considerations |
| | 1. No REQUEST,lfn,*pf. |
| | 2. Rewrite/Overwrite/Extension are automatic. |
| | 3. How to do a NOS/BE ALTER. |
| | 4. CATLIST - AUDIT. |
| | 5. No 'random bit' copy problem on NOS. |
| | 6. 7 character PFN |
| V5-12,13 | 7. No cycle - Simulate with CCL |
| | 8. No user dump/load. |
| | 9. No Archiving. |
| V5-14 | 10. Very large files. |

# PERMANENT FILES

## CHARACTERISTICS

- OWNERSHIP
- TYPES - DIRECT AND INDIRECT
- CATEGORIES - PUBLIC, SEMI-PRIVATE,
  PRIVATE
- ACCESS MODES
- COMMANDS

# PERMANENT FILES

## CHARACTERISTICS

- FILES BELONG TO SPECIFIC USERS.

  NOS IS SET UP TO KEEP USERS AWAY FROM EACH OTHER. IF YOU WANT TO SHARE FILES YOU MUST MAKE THEM PUBLIC OR PERMIT SPECIFIC USERS TO ACCESS THE FILE(S).

- TWO TYPES OF FILES - DIRECT AND INDIRECT

  WHEN AN INDIRECT FILE IS ASSIGNED TO A JOB YOU GET A COPY IN SCRATCH SPACE. WHEN A DIRECT FILE IS ASSIGNED TO A JOB YOU GET A REAL FILE.

  DIRECT FILES TAKE UP 'MORE' DISK SPACE- THE OLD RECORD BLOCK PROBLEM. INDIRECT PERMANENT FILE SPACE IS MAINTAINED IN INCREMENTS OF PRUs BY NOS. DIRECT FILES ARE IDEAL FOR LARGE DATA FILES WHILE INDIRECT FILES ARE IDEAL FOR SMALL (LESS THAN A LOGICAL TRACK) DATA FILES.

## PERMANENT FILES CHARACTERISTICS - CONTINUED

- THREE CATEGORIES

PUBLIC - ANYONE WHO KNOWS USER-NAME, FILE NAME AND PASSWORD IF SPECIFIED CAN ACCESS.

SEMI-PRIVATE - SAME AS PUBLIC EXCEPT THAT A RECORD IS KEPT OF WHO AND WHEN THE FILE WAS ACCESSED.

PRIVATE - ONLY YOU CAN ACCESS OR SPECIFIC USERS PERMITED FOR ONLY SPECIFIC ACCESS MODE.

## PERMANENT FILE CHARACTERISTICS - CONTINUED

- EIGHT ACCESS MODEs. EACH MODE IS 'HIGHER' THAN PREVIOUS.

| | |
|---|---|
| N | - NULL |
| E | - EXECUTE ONLY |
| R | - READ ONLY |
| RA | - READ AND/OR APPEND |
| RM | - READ AND/OR MODIFY |
| A | - APPEND |
| M | - MODIFY |
| W | - WRITE |

## PERMANENT FILE CHARACTERISTICS - CONTINUED

- COMMANDS

| | | |
|---|---|---|
| DIRECT AND INDIRECT | CATLIST | - SORTED LIST OF PFS AND MISC. INFO. LIKE AUDIT EXCEPT MUCH MORE USER FRIENDLY! |
| | PURGE | - REMOVE A FILE FROM PF CATALOG. |
| | PERMIT | - ALLOW OTHER USERS TO ACCESS. |
| | CHANGE | - CHANGE PERMISSIONS. |
| | PURGALL | - PURGE ALL FILES! |
| DIRECT ONLY | DEFINE | - ESTABLISH A DIRECT FILE. |
| | ATTACH | - ACCESS A DIRECT FILE. |
| INDIRECT ONLY | SAVE | - ESTABLISH AN INDIRECT FILE. |
| | GET | - ACCESS AN INDIRECT FILE. |
| | REPLACE | - REPLACE AN EXISTING INDIRECT FILE. |
| | APPEND | - ADD A LOCAL FILE TO THE END OF AN INDIRECT FILE. |

## DIRECT FILES

### CREATION

```
        DEFINE,[LFN=]PFN[/CT=CATEGORY,M=MODE,
                          PW=PASSWORD].
```

PLACE WHERE YOU WOULD NORMALLY INSERT THE NOS/BE 'REQUEST,LFN,*PF' STATEMENT.

'CATEGORY' MAY BE

P  - PRIVATE<br>
S  - SEMI-PRIVATE<br>
PU - PUBLIC

## DIRECT FILES - CONTINUED

THE MEANING OF THE 'MODE' PARAMETER DEPENDS UPON THE CATEGORY OF THE FILE -

PRIVATE - MODE HAS NO MEANING SINCE OWNER CAN ACCESS AT ANY LEVEL AND EACH ALTERNATE USER MUST EXPLICITLY BE PERMITTED TO ACCESS THE FILE.

PUBLIC AND SEMI-PRIVATE - MODE IS THE MAXIMUM ACCESS MODE FOR ALL ALTERNATE USERS EXCEPT THOSE WHO HAVE BEEN EXPLICITLY PERMITTED BY THE OWNER. AN EXPLICIT PERMIT TAKES PRECEDENCE OVER MODE ON THE DEFINE FOR PERMITTED USERS.

## DIRECT FILES - CONTINUED

### ACCESS

```
ATTACH,[LFN=]PFN[/UN=USER-NAME,M=MODE,
                      PW=PASSWORD].
```

'USER-NAME' IS OPTIONAL IF PFN IS YOUR FILE.

'MODE' IS THE ACCESS MODE DESIRED FOR THIS ACCESS. IF AN ALTERNATE USER ATTEMPTS ACCESS OF A PERMITED PRIVATE FILE AND EXCEEDS THE PERMITED MODE, THE FOLLOWING MESSAGE IS DISPLAYED -

```
PFN NOT FOUND
```

THE SAME THING HAPPENS FOR PUBLIC AND SEMI-PRIVATE FILES WHEN THE AUTHORIZED MODE IS EXCEEDED.

DEFAULT 'MODE' IS READ IF M IS NOT SPECIFIED.

## INDIRECT FILES

### CREATION

```
SAVE,[LFN=]PFN[/CT=CATEGORY,M=MODE,
                PW=PASSWORD].
```

COPIES LFN TO AREA NOW CALLED PFN.

```
REPLACE,[LFN=]PFN[/UN=USER-NAME,PW=PASSWORD].
```

REPLACE DOES AN IMPLICIT PURGE AND SAVE. IT IS VERY DIFFICULT TO GO BACK TO NOS/BE AND HASSLE CYCLES AFTER USING THIS COMMAND.

## INDIRECT FILES - CONTINUED

### ACCESS

```
GET,[LFN=]PFN[/UN=USER-NAME,PW=PASSWORD].
```

COPIES PFN TO LFN.

### EXTENSION

```
APPEND,PFN,LFN1,LFN2...[/UN=USER-NAME,
                             PW=PASSWORD].
```

COPIES LFN1, LFN2, ETC., TO THE END OF INFORMATION OF PFN.

## PERMANENT FILES

### SPECIAL CONSIDERATIONS

- NO REQUEST,LFN,*PF
- SEQUENTIAL REWRITE IS AUTOMATIC
- OVERWRITE IS AUTOMATIC
- EXTEND IS ALWAYS AUTOMATIC ON DIRECT FILES
- NO PFD OVERFLOW PROBLEMS
- ALTER

  TO TRUNCATE FILE USE WRITER,LFN.

  TO 'REDEFINE' FILE JUST REWIND AND START WRITING.
- CATLIST IS 'TERMINAL-FRIENDLY'

  FN
  UNIVERSAL CHARACTER
  LO
  LENGTH STATEMENT

## PERMANENT FILES

### SPECIAL CONSIDERATIONS - CONTINUED

- REPLACE IS VERY EASY TO LOVE.

- RANDOM FILES CAN BE COPIED DISK TO DISK (NO RANDOM BIT PROBLEM) AND BE USED. RANDOM FILES CAN ALSO BE COPIED TO TAPE FOR BACKUP PURPOSES AND COPIED BACK TO DISK FOR USE.

- 40 CHARACTER PFN - BITE THE BULLET

- USER DUMP/LOAD

  - IF THE RANDOM BIT IS THE PROBLEM ON NOS/BE; THERE IS NO PROBLEM ON NOS. USE COPYEI.

- ARCHIVING

  SPL - SPECIAL PRODUCTS LIBRARY.
  FUTURE SYSTEM RELEASE (1ST HALF 80).

- CYCLES - MUST BE NICE TO HAVE THAT MUCH DISK SPACE...

  - REWIND THEN WRITE
  - REPLACE
  - USE CCL PROC

```
.PROC,NEWDEF,LOCAL,PERM.

 PURGE,PERM→1/NA.                    PURGE OLDEST

 CHANGE,PERM→1 = PERM→2/NA.          2 TO OLDEST

 CHANGE,PERM→2 = PERM→3/NA.          3 TO 2

 DEFINE,LOCAL = PERM→3.              NEW TO 3




 NEWDEF,PUPPY,DOG.


 NOTES -
         CHANGE,NEW=OLD/NA.
         DOG1 = OLDEST
         DOG2 =
         DOG3 = NEWEST
```

## PERMANENT FILES

### SPECIAL CONSIDERATIONS - CONTINUED

- VERY LARGE FILES - USE POSITIVE CONTROL (KNOW WHERE THE FILE IS LOCATED!!).

  - USE AUXILARY MULTI-SPINDLE LOGICAL DEVICE PRIVATE PACKS.

    PACKNAM

  - USE MULTI-SPINDLE DEVICE FAMILIES.

  EITHER WILL RESULT IN BETTER ACCESS.

- MACROS - WATCH OUT FOR PF FET - CRM FIT CONFLICT.

# LESSON GUIDE 6

# MAGNETIC TAPE USAGE

## LESSON PREVIEW

This lesson explains magnetic tape reservations, assignment and release on NOS.

## REFERENCES

NOS Reference Manual - Chapter 10

## TRAINING AIDS

Visuals V6-1 through V6-4

## ACTIVITIES

## OBJECTIVES

1. Understand the differences between NOS/BE and NOS tape reservations.
2. Emphasize VSN label standard differences.

# LESSON 6 OUTLINE

## MAGNETIC TAPE USAGE

| VISUALS | TOPICS |
|---|---|
| V6-1,2 | A. Reservations |
| | 1. One device - automatic. |
| | 2. More than one device - requires a RESOURC statement before first device is assigned. |
| | 3. Reservation can be lowered with another RESOURC. |
| V6-3 | B. Assignment - LABEL statement. |
| | 1. Discuss general parameters. |
| | 2. PO=R or PO=W controls enforcement of RING. |
| | 3. R or W controls processing of the labels. |
| V6-4 | C. Release. |

## MAGNETIC TAPE

- RESERVATION
- ASSIGNMENT
- RELEASE
- DIFFERENCES

## MAGNETIC TAPE

- RESERVATION

IF ONLY ONE DEVICE (7 TRACK DRIVE OR 9 TRACK DRIVE OR DISK DRIVE) IS REQUIRED NO RESERVATION IS REQUIRED.

IF MORE THAN ONE DEVICE IS REQUIRED THEN A RESOURC STATEMENT MUST BE ENTERED IN THE JOB STREAM BEFORE THE FIRST DEVICE IS ASSIGNED.

```
RESOURC(MT=2,NT=1)
```

THIS STATEMENT INDICATES THAT THE JOB NEEDS TO USE TWO 7 TRACK DRIVES AND ONE 9 TRACK DRIVE AT THE SAME TIME. THE NOS/BE USER WOULD HAVE ENTERED -

```
...MT2,NT1...
```

ON THE JOB STATEMENT.

THE COUNTS MAY BE LOWERED BY ANOTHER RESOURC STATEMENT.

## MAGNETIC TAPE

- ASSIGNMENT

Use the LABEL statement. It does everything. Most of the parameters are identical to NOS/BEs LABEL parameters.

```
LABEL,lfn,
  D = den,       den may be HI, HY, HD, GE, etc.
  F = fmt,       Data format - I,S,L,SI
                 (SI is a NOS/BE System
                 Internal).
 LB = type,      Label type may be -
                 KL - NOS labeled (ANSI)
                 KU - Unlabeled
                 NS - Non-standard
VSN = vsn,
 PO = option,    Processing option A,B,F,R,W,etc.
  R,             Read Existing label
 W)              Write new label
```

There are other parameters listed in the NOS R.M. Notice that there is no RING/NORING parameter. This is accomplished by the PO=W and PO=R options. PO also has many other options (parity error processing, etc.).

## MAGNETIC TAPE

THE VSN PARAMETER APPEARS INNOCENT BUT THERE IS ONE SITUATION TO BE AWARE OF -

NOS/BE IMPLEMENTED AN OLD ANSI LABEL STANDARD WHILE NOS IMPLEMENTED THE NEW ANSI STANDARD. THE VSN IS LEFT JUSTIFIED WITH BLANK FILL ON NOS WHILE IT IS RIGHT JUSTIFIED WITH ZERO FILL ON NOS/BE. TO READ A NOS/BE TAPE ON NOS, SPECIFY THE VSN AS A SIX CHARACTER FIELD WITH LEADING ZEROS.

VSN=4567 ON NOS/BE WOULD BE
VSN=004567 ON NOS

- RELEASE

RETURN AND UNLOAD OPERATE THE SAME ON NOS AND NOS/BE.

# LESSON GUIDE 7

## SOURCE STORAGE, USAGE AND MAINTENANCE

### LESSON PREVIEW

This lesson introduces the student to the source maintenance facilities on NOS.

### REFERENCES

None.

### TRAINING AIDS

Visual V7-1

### ACTIVITIES

### OBJECTIVES

1. Identify source maintenance facilities (UPDATE, MODIFY, XEDIT).
2. Describe ease of using UPDATE on NOS.
3. Introduce systems analysts to MODIFY for operating system modules.

## LESSON 7 OUTLINE

## SOURCE STORAGE, USAGE AND MAINTENANCE

VISUALS TOPICS

V7-1 A. UPDATE

1. Common on both systems.
2. PLs can be copied disk-to-disk.
3. More features are being developed on UPDATE.

B. MODIFY

1. Similar to UPDATE.
2. Required for operating system modules.
3. PL must be on disk.
4. No feature development planned.

C. XEDIT

1. Runs in batch and interactive.
2. More details later in course.

## SOURCE PROGRAM LIBRARIES

- UPDATE - IT IS THE SAME ON BOTH SYSTEMS. YOU KNOW HOW TO USE IT. CONTINUE TO USE IT. PLS CAN BE COPYEIED SINCE THERE IS NO RANDOM BIT PROBLEM. FUTURE DEVELOPMENT IS SCHEDULED FOR UPDATE. E.G. 8 BIT SUPPORT.

- MODIFY - HAS SIMILAR FEATURES AS UPDATE. CANNOT BE USED WHEN THE PL RESIDES ON TAPE. OPERATING SYSTEM SOFTWARE IS MAINTAINED BY MODIFY.

- XEDIT - AN EDITOR THAT WORKS BOTH IN BATCH AND INTERACTIVE. MORE LATER IN THIS PRESENTATION.

# LESSON GUIDE 8

## OBJECT MODULE STORAGE, USAGE AND MAINTENANCE

### LESSON PREVIEW

This lesson provides the student with a firm grasp of how to work with the various types of object modules on NOS.

### REFERENCES

NOS Reference Manual - Chapter 7.

### TRAINING AIDS

Visuals V8-1 through V8-7

### ACTIVITIES

Prepare for laboratory to reinforce source and object module storage, usage and maintenance.

### OBJECTIVES

1. Describe storage techniques - sequential files and random user libraries.

2. Describe what can be stored in <u>and</u> accessed from user libraries.

3. Describe work-arounds for using absolutes and procedure files on NOS.

4. Describe maintenance techniques available on NOS/COPYL, GTR, LIBEDIT, LIBGEN, etc.)

5. Explain NOS/BE EDITLIB work-arounds that are available.

# LESSON 8 OUTLINE

## OBJECT MODULE STORAGE, USAGE AND MAINTENANCE

VISUALS — TOPICS

V8-1 — A. Storage

1. Sequential files - same as NOS/BE.
2. Random CYBER Loader 'library' files.

V8-2 — B. Usage

1. Sequential files - same for relocatable and capsules.
2. Capsules - must be in libraries on both systems.
3. Absolutes - cannot be stored in libraries.
   a. Use FOL on sequential files.
   b. Use GTR for 'libraries' of absolutes.
   c. Pre-release code is available.
4. CCL Procedures cannot be stored in libraries.
   a. Store all procedures on a sequential file.

V8-3 — 5. Relocatable main programs cannot be accessed from user libraries by the operating system.
   a. Invoke the CYBER Loader.

V8-4 — C. Maintenance

1. Sequential
   a. COPYL (COPYLM)
   b. ITEMIZE
   c. GTR
   d. LIBEDIT
2. Random CYBER Loader 'libraries'.

## LESSON 8 OUTLINE

OBJECT MODULE STORAGE, USAGE AND MAINTENANCE (continued)

| VISUALS | TOPICS |
|---|---|
| | a. LIBGEN - Only a partial solution. |
| V8-5,6 | b. CCL procedures to create and maintain user libraries on NOS. |
| V8-7 | c. GTR - Very powerful utility. |

## OBJECT MODULE 'LIBRARIES'

- STORAGE

  SEQUENTIAL FILES
  RANDOM 'LIBRARY' FILES USABLE BY THE
  CYBER LOADER

- USAGE

- MAINTENANCE

## OBJECT MODULE LIBRARIES - CONTINUED

## USAGE

- SEQUENTIAL FILES OF RELOCATABLES OR ABSOLUTES PERFORM THE SAME WAY ON BOTH SYSTEMS.

- CAPSULES ARE STORED IN RANDOM LIBRARIES ON BOTH SYSTEMS. THEY ARE ACCESSED IN THE SAME WAY ON BOTH SYSTEMS.

- ABSOLUTES CURRENTLY CANNOT BE STORED IN NOS USER LIBRARIES.

  - IF PURPOSE OF INSERTING ABSOLUTES INTO A USER LIBRARY ON NOS/BE WAS TO SAVE DISK ACCESSES THEN FOL IS FASTER THAN NON-FOL NOS/BE LIBRARIES.

  - GTR (GET RECORDS) CAN BE USED TO ACCOMPLISH SOME FUNCTIONS OF A GLOBAL LIBRARY STATEMENT.

  - CODE IS AVAILABLE (VIA THE UNDERGROUND) FOR LIBGEN. IT MIGHT BE STANDARD WHEN YOU CONVERT.

## OBJECT MODULE LIBRARIES

### USAGE - CONTINUED

- RELOCATABLE MAIN PROGRAMS CAN BE STORED IN NOS USER LIBRARIES BUT ONLY THE CYBER LOADER KNOWS HOW TO READ GLOBAL LIBRARIES. NOS DOES NOT SEARCH GLOBAL LIBRARIES IN RESPONSE TO A NAME CALL LOAD. NOS DOES NOT INVOKE THE CYBER LOADER UNLESS EXPLICITLY CALLED. TRY THE FOLLOWING -

```
LDSET,LIB=LIBNAME.
MYPROG.
```

  CPUMTR (1AJ) CODE IS AVAILABLE VIA THE UNDERGROUND.

- CCL PROCEDURES (THE ONLY KIND) CURRENTLY CANNOT BE STORED IN NOS USER LIBRARIES.

  - PLACE YOUR PROCEDURES INTO AN INDIRECT PF CALLED PROCFIL. ENTER '-PROC-NAME,,P1,P2...' NOS (CCL) WILL DO AN AUTOMATIC GET OF THE PROPER PROCEDURE IN INTERACTIVE.

    'BEGIN,PROC-NAME,,P1,P2... IS REQUIRED IN BATCH.

## OBJECT MODULE LIBRARIES

## MAINTENANCE

- SEQUENTIAL FILES

  - COPYL - THE SAME
  - COPYLM - THE SAME
  - ITEMIZE- THE SAME
  - GTR - VERY HANDY EXTRACTION UTILITY!
  - LIBEDIT- FANCY, DIRECTIVE DRIVEN COPYL. CREATES A NEW 'CLEAN' FILE.

- RANDOM FILES (USER LIBRARIES)

  ONLY UTILITY IS LIBGEN. IT COPIES A CLEAN SEQUENTIAL FILE TO USER LIBRARY FORMAT.

  - USE A CCL PROCEDURE TO COMBINE THE MUNDANE TASKS OF GTR, LIBEDIT AND LIBGEN. RESULTS IN NO NOS/BE-LIKE DEAD DISK SPACE!

## OBJECT MODULE LIBRARIES

## MAINTENANCE

Consider the following two CCL procedures - one to generate an 'empty' user library and another to maintain the library.

```
.PROC,EMPTYLB,LIBNAME,INP=#DATA.
RETURN,TEMP.
FTN,I=INP,L=0,B=TEMP.
LIBGEN,F=TEMP,P=LIBNAME.
RETURN,TEMP.
.DATA
      SUBROUTINE   AAAAAAA
      END
```

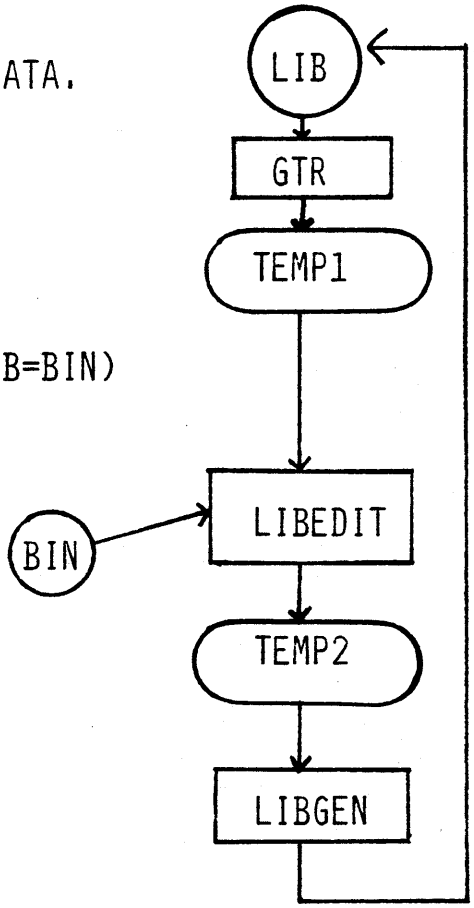
This procedure is accessed by -

```
EMPTYLB,mylib.
```

## OBJECT MODULE LIBRARIES

## MAINTENANCE - CONTINUED

```
 .PROC,UPLIB,LIBNAME,BIN,INP=#DATA.

 RETURN(TEMP1,TEMP2)

 GTR(LIBNAME,TEMP1)ULIB/LIBNAME

 LIBEDIT(I=INP,P=TEMP1,N=TEMP2,B=BIN)

 LIBGEN(F=TEMP2,P=LIBNAME)

 .DATA

*INSERT,REL/AAAAAAA,REL/*
```



THE PROCEDURE IS ACCESSED BY -

```
        UPLIB,MYLIB,MYLGO.
```

## OBJECT MODULE LIBRARIES

## MAINTENANCE - CONTINUED

GTR - A VERY POWERFUL UTILITY.

```
GTR,OLD-FILE,NEW-FILE. TYPE/NAMES
```

GTR SCANS 'OLD-FILE' FOR THE RECORD(S) CALLED 'NAMES' OF THE SPECIFIED 'TYPE' (REL,ABS,ETC.) AND COPIES THE RECORDS TO 'NEW-FILE'.

# LESSON GUIDE 9

# INTERACTIVE OVERVIEW

## LESSON PREVIEW

This lesson provides an overview of the interactive facilities that are available on NOS.

## REFERENCES

## TRAINING AIDS

Visuals V9-1 through V9-3

## ACTIVITIES

## OBJECTIVES

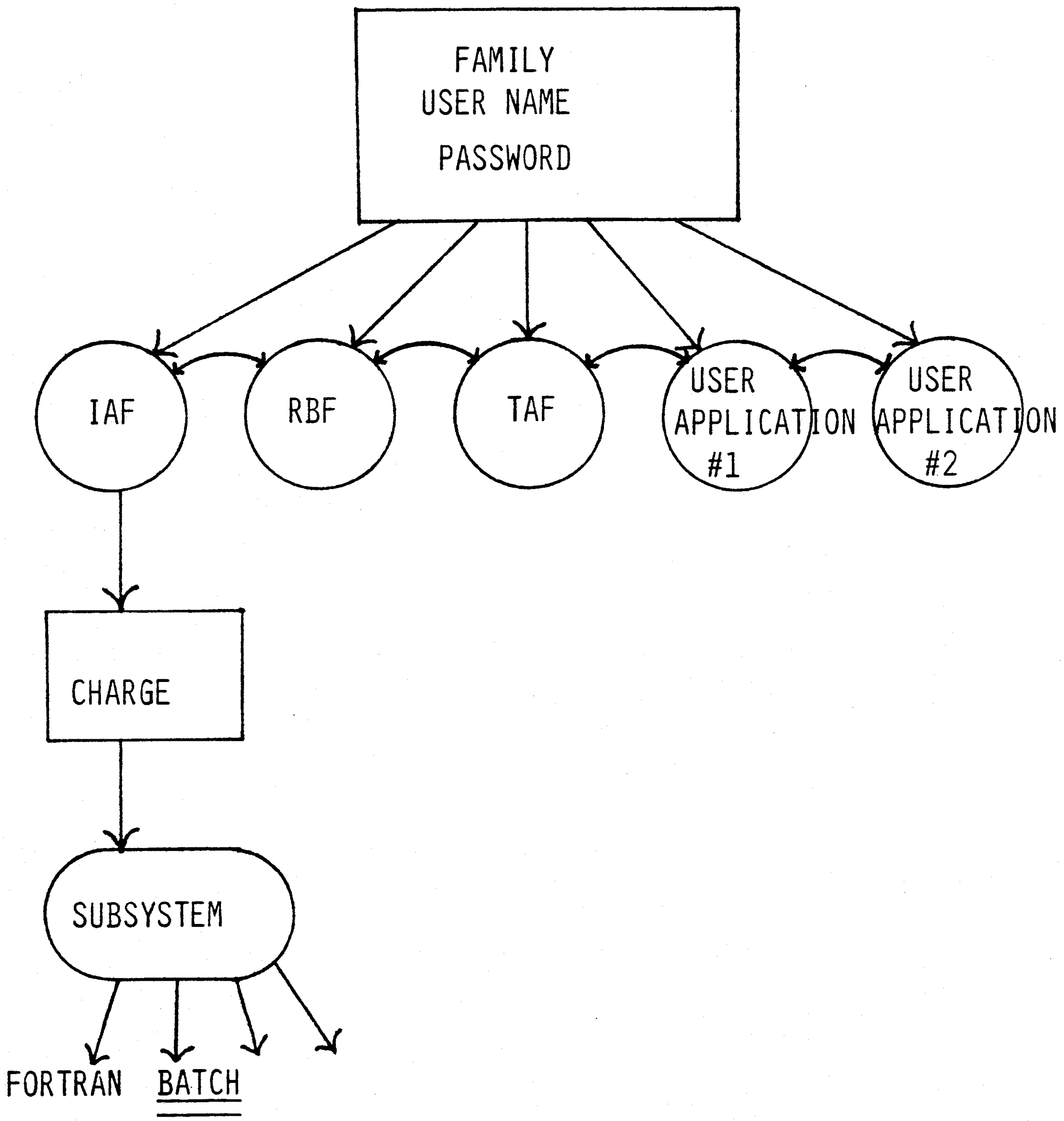
1. Provide an overview of the interactive facilities that are available on NOS - time sharing, remote batch, teleprocessing and transaction processing.

# LESSON 9 OUTLINE

## INTERACTIVE OVERVIEW

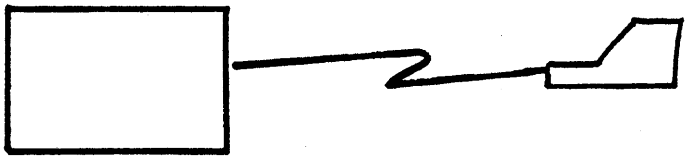
| VISUALS | TOPICS |
|---|---|
| V9-1 | A. System access to interactive. |
| V9-2 | B. Types of interactive access through NAM. |
| | 1. Time sharing (IAF). |
| | a. One program interacting with one terminal. |
| | 2. Teleprocessing (QTRM). |
| | a. User defines (codes) application. |
| | b. One program interacting with many terminals. |
| | 3. Transaction (TAF). |
| | a. Many tasks interacting with many terminals. |
| | 4. Remote Batch (RBF). |
| | a. Card reading. |
| | b. Line printing. |
| | c. Card punching. |
| | d. Plotting. |

## INTERACTIVE

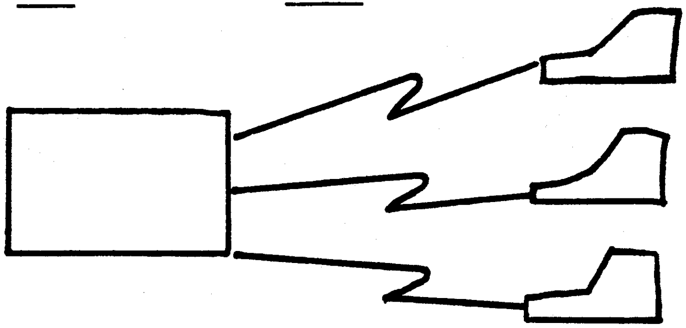


## TYPES OF INTERACTIVE PROCESSING

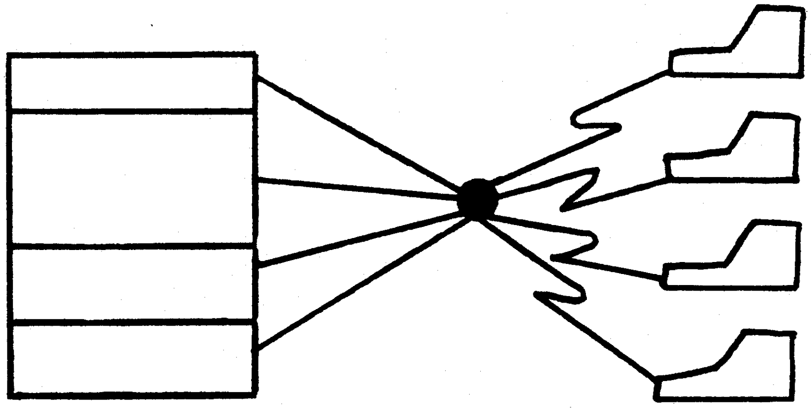
- TIME SHARING (IAF)
  ONE PROGRAM - ONE TERMINAL

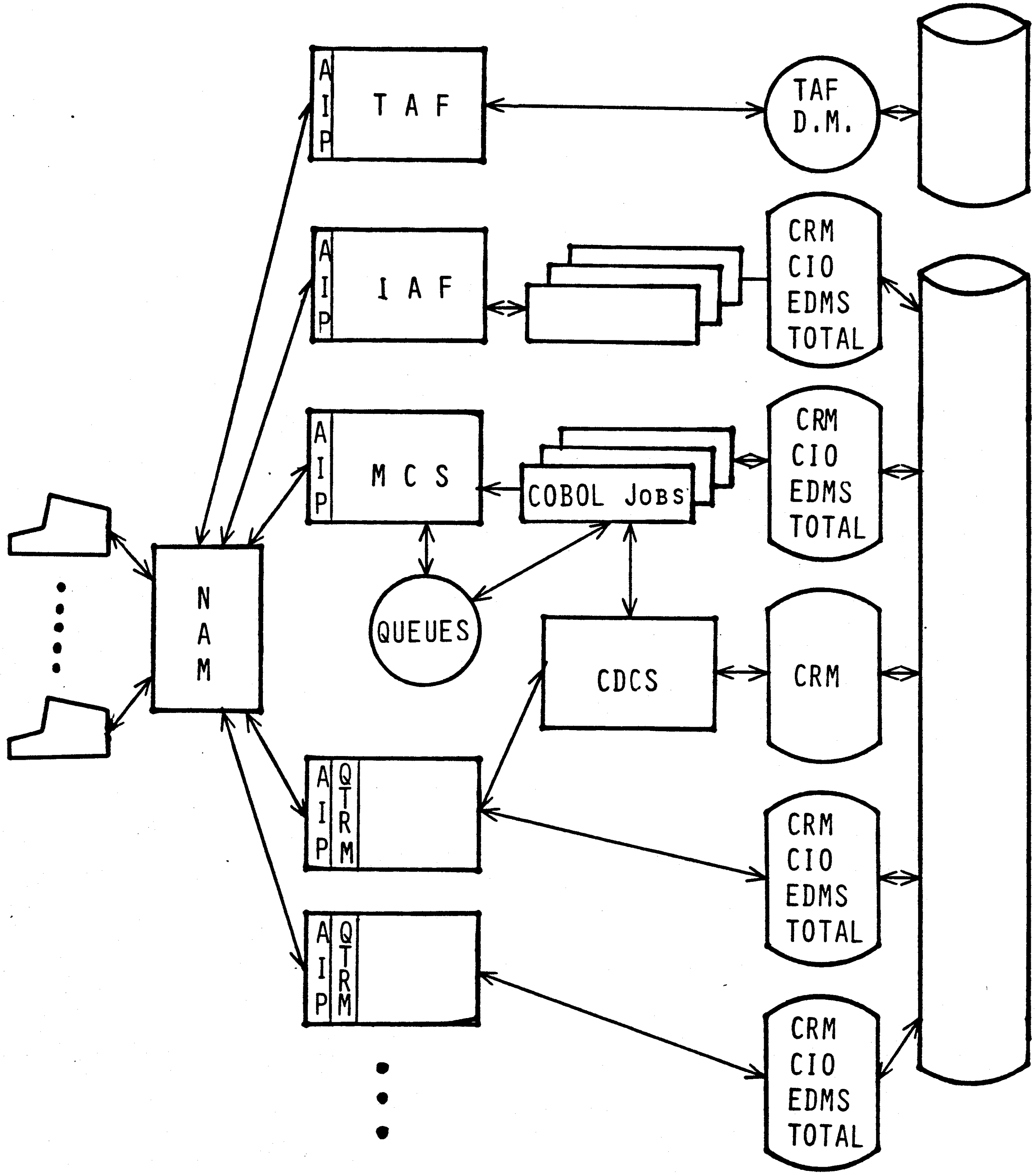


- TELEPROCESSING (QTRM)
  ONE PROGRAM - MANY TERMINALS



- TRANSACTION (TAF)
  MANY TASKS - MANY TERMINALS




AIP
TAF
TAF D.M.
AIP
IAF
CRM CIO EDMS TOTAL
AIP
MCS
COBOL JOBS
CRM CIO EDMS TOTAL
NAM
QUEUES
CDCS
CRM
AIP
QTRM
CRM CIO EDMS TOTAL
AIP
QTRM
CRM CIO EDMS TOTAL

# LESSON GUIDE 10

# TIME SHARING

## LESSON PREVIEW

This lesson provides the student with general and specific knowledge necessary to work with interactive time sharing on NOS.

## REFERENCES

IAF Reference Manual

## TRAINING AIDS

Visuals V10-1 through V10-5

## ACTIVITIES

## OBJECTIVES

1. Describe logon and general procedures.
2. Identify terminal I/O differences (prompts, 'connected' files, etc.).
3. Identify new commands - DAYFILE, ENQUIRE, etc.
4. Describe how to use the CYBER Loader in interactive.
5. Describe how dumps are handled (DMPX to ZZZDUMP).
6. Emphasize that only the last dayfile message is displayed and describe work arounds.

## LESSON 10 OUTLINE

## TIME SHARING

VISUALS — TOPICS

V10-1 — A. Logging in to time sharing (IAF).

1. Family, user, password, applications.
2. Charge.
3. Example uses default batch subsystem.
4. Terminal number for recovery.

V10-2 — B. Time Sharing on NOS.

1. INPUT and OUTPUT are 'connected' by default.
    a. Utilities interact with INPUT and OUTPUT.
    b. ASSIGN,MS,lfn.
    c. ASSIGN,TT,lfn - discards current lfn.
2. Most of the batch control statements work in time sharing - even magnetic tape.
3. There is always a prompt when input is requested.
4. Escape characters.
5. ASCII is standard in BASIC and XEDIT.
6. (V10-3) ENQUIRE replaces many NOS/BE control statements (FILES, ASSETS, etc.).

V10-4 — C. Special Considerations

1. CYBER Loader 'load sequences'.
    a. Use ENTER control statement.
    b. Use CCL.
2. Use X,LIBRARY,lfn.

## LESSON 10 OUTLINE

## TIME SHARING (continued)

VISUALS TOPICS

3. DMPX is written to ZZZDUMP.

   a. Use DMD with OUTPUT assigned to MS.

   b. Build DMx into CCL procedure.

   c. Use CCL procedures.

V10-5

4. Only last dayfile message is displayed.

   a. Modify applications.

   b. Use FTN,A,L=0 etc., to get only diagnostics.

   c. Use DAYFILE,OP=I.

5. Type Ahead does not exist.

6. Carriage controls do not exist - control bytes work for some applications.

## INTERACTIVE - TIME SHARING

## LOGGING IN

```
79/08/23   14.59.16   TM1017
(22)  SVL   SN614  NOS
FAMILY:   SVL
USER:  JFOX
PASSWORD:  NOWAY
APPLICATION: IAF
TERMINAL  13,NAMIAF
RECOVER/CHARGE: CHARGE,MYDEPT,MYPROJECT

/
```

CAN BE ABREVIATED AND ENTERED ON FAMILY LINE-

SVL,JFOX,NOWAY,IAF

IAF IS READY FOR YOU TO ENTER A COMMAND!   (BATCH SUBSYSTEM)

NOTICE THAT THE PASSWORD WAS CHANGED (PASSWOR) BETWEEN THE PREVIOUS BATCH RUN AND THIS INTERACTIVE SESSION.

## INTERACTIVE

### GENERAL

- INPUT AND OUTPUT ARE 'CONNECTED' BY DEFAULT. MOST PROGRAMS AND UTILITIES SEND DATA TO OUTPUT AND RECEIVE DATA FROM INPUT.

  ASSIGN,MS,OUTPUT WILL ASSIGN OUTPUT OR ANY FILE TO A DISK DEVICE. THIS IS LIKE THE NOS/BE DISCONT STATEMENT.

  ASSIGN,TT,OUTPUT WILL DISCARD ANYTHING CURRENTLY ON OUTPUT AND ASSIGN IT TO THE TERMINAL. THIS IS LIKE THE NOS/BE CONNECT STATEMENT.

- GENERALLY, MOST OF THE BATCH STATEMENTS WORK IN INTERACTIVE - EVEN THE MAGNETIC TAPE STATEMENTS.

- DAYFILE IS A VERY HANDY STATEMENT AND SHOULD BE LEARNED QUICKLY AND EASILY.

- THERE IS ALWAYS A PROMPT (?) WHEN INPUT IS EXPECTED FROM THE TERMINAL USER.

## INTERACTIVE

### GENERAL - CONTINUED

- ASCII (6/12) IS STANDARD IN BASIC AND XEDIT. FCOPY CONVERTS FROM 6/12 TO 8/12 ASCII.

- A CARRIAGE RETURN (CR) BY ITSELF SIGNALS AN EOF. THIS IS LIKE THE NOS/BE %EOF. IF DATA WAS NOT EXPECTED (READ STATEMENT IN PROGRESS) IAF WILL RESPOND WITH THE CURRENT STATUS (EXECUTE,IDLE,ETC.).

- ENQUIRE, OPT. ANOTHER HANDY UTILITY. IF OPT IS 'F' A STATUS OF ALL FILES IS GIVEN - LIKE FILES. 'J' IS THE JOB CONTROL REGISTER. 'R' LISTS RESOURCES USED. JN (JOB NAME) IS VERY USEFUL. THE LIST GOES ON.

- RENAME, NEWLFN=OLDLFN. CHANGES THE <u>LOCAL</u> FILE NAME. USE CHANGE TO CHANGE A PERMANENT FILE NAME.

- REWIND,*. REWINDS ALL FILES.

- RETURN,*. RETURNS ALL FILES. RETURN,OUTPUT SENDS 'OUTPUT' TO THE BIT BUCKET!

## INTERACTIVE

## SPECIAL CONSIDERATIONS

- CYBER LOADER - MUST GROUP DIRECTIVES INTO 'PACKETS' LIKE ON NOS/BE.

  - NOS/BE USES XEQ (WHAT A MESS) FOR ONE TIME EXECUTION.

  - NOS USES THE ENTER DIRECTIVE FOR ONE TIME EXECUTIONS.

    ```
    ENTER./CMD1,/CMD2./CMD3./.../
    ```

    CMD CAN BE ANY STATEMENT.

  - CCL PROCEDURES ARE EVEN MORE POWERFUL AND VERSATILE!

- LIBRARY - CAUTION!

  TO SPECIFY A GLOBAL LIBRARY SET USE X.LIBRARY,LFN...
  THE IAF 'LIBRARY' STATEMENT BEING CHANGED TO
  'LIB' IN A FUTURE SYSTEM.

- DMP - ABNORMAL TERMINATION DMP IS WRITTEN TO ZZZDUMP. USE XEDIT TO PRINT IT.

  - USE DMD. ASSIGN OUTPUT TO MS.
  - BUILD INTO ENTER STATEMENT.
  - USE CCL PROCEDURE

## INTERACTIVE

## SPECIAL CONSIDERATIONS - CONTINUED

- ONLY LAST DAYFILE MESSAGE IS DISPLAYED.

  - USE FTN,A,L=0 ETC.
  - USE DAYFILE

- TYPE AHEAD

  - USE CCL
  - WAIT FOR R6

- DEFAULT CARRIAGE CONTROL IS A BLANK. SPECIAL PROCESSING (CONTROL BYTES) IS REQUIRED BUT VERY POWERFUL FOR NON-DEFAULT CARRIAGE CONTROLS.

# LESSON GUIDE 11

# TEXT EDITORS

## LESSON PREVIEW

This lesson presents the student with advantages and disadvantages of the various text editors on NOS. It also provides detailed information about XEDIT.

## REFERENCES

XEDIT Reference Manual.

## TRAINING AIDS

Visuals V11-1 through V11-5

## ACTIVITIES

Prepare for laboratory session on time sharing and text editing.

## OBJECTIVES

1. Instill the philosophy of "choose one editor; learn it; use it".
2. Compare INTERCOM tasks on XEDIT.
3. Introduce 'pointer' editor concepts.

# LESSON 11 OUTLINE

## TEXT EDITORS

| VISUALS | TOPICS |
|---|---|
| V11-1 | A. Text editor situation on NOS. |
| | B. Compare/Contrast various text editors. |
| | 1. Choose one editor. |
| | 2. Learn it well. |
| | 3. Use it. |
| V11-2 | C. XEDIT. |
| | 1. Compromises. |
| | a. Pointer based instead of line numbers. |
| | b. Operating system control statements cannot be intermixed with XEDIT directives. |
| V11-3 | 2. Learn in phases. |
| V11-4 | 3. Pointer movement, etc. |
| V11-5 | D. Special Considerations. |
| | 1. XEDIT can be used in batch. |
| | 2. XEDIT can be used for NOS/BE PAGE functions. |
| | 3. XEDIT has an excellent PF interface - eliminates NOS/BE 'SAVE' and 'CATALOG', etc. |

## INTERACTIVE

## EDITORS

MANY TEXT EDITORS EXIST ON NOS. IT IS FUNNY TO WATCH PEOPLE TRY TO USE ALL OF THEM AT THE SAME TIME.

CHOOSE ONE!

LEARN IT WELL!

USE IT!

THREE OF THE EDITORS INCLUDE -

- NOS LINE EDITOR - GREAT FOR THE OPERATING SYSTEM
- EDIT
- XEDIT

## INTERACTIVE

### EDITORS - CONTINUED

I CHOOSE XEDIT. IT CAN DO JUST ABOUT EVERYTHING THE NOS/BE EDITOR CAN DO PLUS SOME VERY POWERFUL NEW COMMANDS (COPY, READ, MODIFY AND INTERNAL PF INTERFACE). I MADE TWO COMPROMISES:

- XEDIT IS A POINTER BASED EDITOR. IT TOOK ME ABOUT 2 WEEKS TO 'FORGET' LINE NUMBERS.

- XEDIT COMMANDS CANNOT BE INTERMIXED WITH OPERATING SYSTEM COMMANDS.

## INTERACTIVE

## EDITORS - CONTINUED

LEARN XEDIT COMMANDS IN PHASES. DON'T GO ONTO SECOND PHASE UNTIL YOU ARE COMFORTABLE WITH FIRST PHASE.

- INSERT
  DELETE
  PRINT
  LOCATE
  CHANGE
  NEXT
  TOP
  BOTTOM
  END
  STOP
  MODIFY

- DEFTAB
  TABS
  LISTAB
  COPY
  READ
  INSERTB
  REPLACE

- QMOD
  RMARGIN
  DEOR
  TRIM
  WEOR
  WEOF
  DEOF

- IGNORE THE 'OBTUSE' COMMANDS!

DIRECTIVES CAN BE PLACED ON THE XEDIT CONTROL STATEMENT. PLACE THE XEDIT CONTROL STATEMENT IN A CCL PROCEDURE TAILORED TO EACH USERS NEEDS (I.E. TABS, BELLS, ETC.).

V11-4

## XEDIT

POINTER ANT

BEAR

CAT

DOG

ELEPHANT

FOX

GOOSE

### POINTER MOVEMENT

TOP
BOTTOM
NEXT
LOCATE

### MODIFICATION

CHANGE
DELETE
INSERT

### DISPLAY

PRINT

### TERMINATION

END

STOP

## INTERACTIVE

## EDITORS - CONTINUED

SPECIAL CONSIDERATIONS

- USE XEDIT IN BATCH
- USE XEDIT FOR PAGE FUNCTIONS
- DON'T USE LINE NUMBERS! THEY ARE A PAIN!
- XEDIT MAKES A LOCAL FILE AND A PERMANENT FILE AT THE SAME TIME.

# LESSON GUIDE 12

# REMOTE BATCH

## LESSON PREVIEW

This lesson introduces the student to remote batch processing on NOS.

## REFERENCES

RBF Reference Manual

## TRAINING AIDS

Visual V12-1 through V12-2

## ACTIVITIES

## OBJECTIVES

1. Describe how to read cards and print lines on a remote batch terminal.
2. Describe how to manipulate queues and files at the remote batch terminal.
3. Describe use of each display.
4. Provide alternatives to the queue manipulation commands of NOS/BE that are not available on NOS.

# LESSON 12 OUTLINE

## REMOTE BATCH

| VISUALS | TOPICS |
|---|---|
| V12-1 | A. Remote Batch Processing (RBF). |
| | 1. Card Reading. |
| | 2. Card Punching. |
| | 3. Line Printing. |
| | 4. Plotting. |
| | 5. Commands. |
| | a. Displays. |
| | b. File manipulation. |
| | c. Queue manipulation. |
| | d. Entry/Exit. |
| V12-2 | B. Special Considerations |
| | 1. Can't use RBF control statements in IAF. (Validate user for both and switch between RBF-IAF). |
| | 2. Can't move print file to 'local'. (Code is being developed). |

## REMOTE BATCH

Remote batch is accomplished by RBF. It has commands similar to INTERCOM's remote batch. The commands of IAF and RBF are processed by separate facilities. INTERCOM merges the commands of both facilities together. RBF commands include -

| | |
|---|---|
| DISPLAY | JOB,IN,PR,PU,DEV,etc. |
| CHANGE | PRI,REP |
| DIVERT | JOB,IN,PR,PU,EX, etc.<br>FAM,USR,HST |
| PURGE | JOB,IN,PR,PU,EX, etc. |
| SKIP | ±n,DFL,END |
| REWIND | |
| RETURN | |

## REMOTE BATCH

## SPECIAL CONSIDERATIONS

- The RBF batch related commands are not available in IAF.

  - Validate users for IAF AND RBF. The user can switch between IAF and RBF to do the RBF queue manipulation commands.

- Can't move a print file from the print queue to a local file.

  - Standard code will be available (1Q80?).

    QGET
    QDROP
    QPURGE

# LESSON GUIDE 13

# TELEPROCESSING

## LESSON PREVIEW

This lesson provides the student with knowledge of the NOS/BE Multi User Job (MUJ) facility on NOS. The process is called teleprocessing while the facility is called QTRM.

## REFERENCES

## TRAINING AIDS

Visuals V13-1 through V13-2

## ACTIVITIES

## OBJECTIVES

1. Describe the teleprocessing facilities available on NOS.

# LESSON 13 OUTLINE

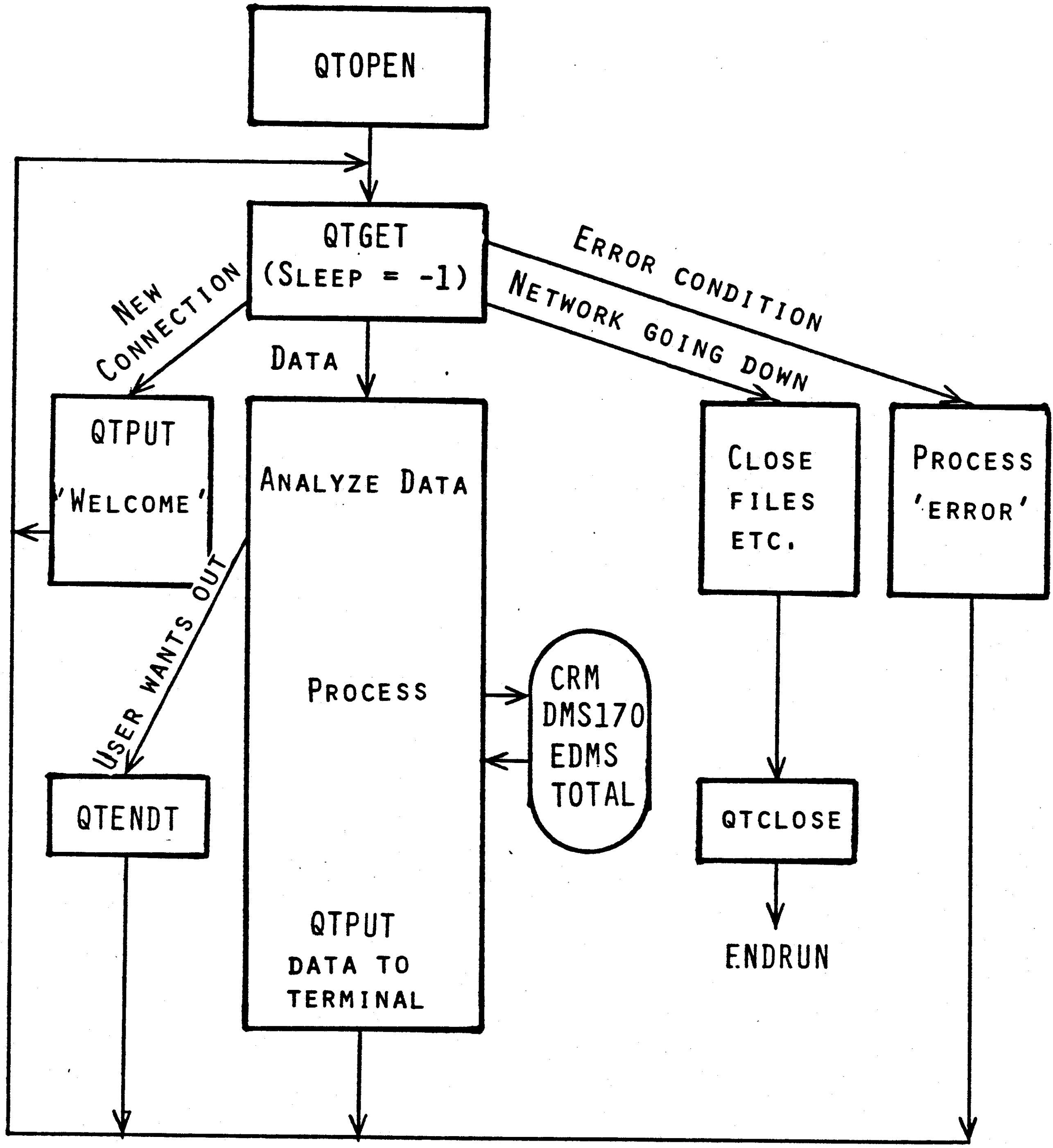
## TELEPROCESSING

| VISUALS | TOPICS |
|---|---|
| V13-1 | A. Architecture/Concepts |
| | One program interacting with many terminals. |
| | B. Procedure calls available from COBOL, COMPASS, FTN. |
| | 1. QTOPEN |
| | 2. QTGET |
| | 3. QTPUT |
| | 4. QTENDT |
| | 5. QTCLOSE |
| V13-2 | C. General program flow. |

## TELEPROCESSING (MUJ)

TELEPROCESSING (ONE PROGRAM COMMUNICATING WITH MANY TERMINALS) IS ACCOMPLISHED BY NAM/AIP/QTRM. SUPPLY THE PROGRAMMER(S) WITH A NAM APPLICATION NAME AND VALIDATION FOR THAT NAME AND LET THEM HAVE THEIR FUN. QTRM WAS DESIGNED FOR COBOL AND FTN PROGRAMMERS. CALLS INCLUDE:

- QTOPEN - IDENTIFIES PROGRAM (APPLICATION) TO NAM.
- QTPUT - SENDS DATA TO A SPECIFIED TERMINAL CONNECTION.
- QTGET - SCANS TERMINALS FOR INPUT FROM THE TERMINAL USERS.
- QTENDT - TERMINATE (NORMAL OR FORCED) A SPECIFIED TERMINAL CONNECTION.
- QTCLOSE - TERMINATES COMMUNICATION BETWEEN THE PROGRAM AND NAM (ALL TERMINALS TO THIS PROGRAM).


QTOPEN
QTGET
(SLEEP = -1)
NEW CONNECTION
DATA
NETWORK GOING DOWN
ERROR CONDITION
QTPUT
'WELCOME'
ANALYZE DATA
PROCESS
QTPUT
DATA TO
TERMINAL
USER WANTS OUT
QTENDT
CRM
DMS170
EDMS
TOTAL
CLOSE
FILES
ETC.
QTCLOSE
ENDRUN
PROCESS
'ERROR'

# LESSON GUIDE 14

# EASING THE TRANSITION

## LESSON PREVIEW

This lesson provides the student with tasks that the user may do now on NOS/BE to ease the transition to NOS.

## REFERENCES

## TRAINING AIDS

Visuals V14-1 through V14-2

## ACTIVITIES

## OBJECTIVES

1. Identify commands that are common between the two operating systems.

2. Identify tasks suitable for inclusion into CCL procedures.

## EASING THE TRANSITION

- USE THE MIGRATION AS AN OPPORTUNITY TO REVIEW CURRENT SYSTEMS, OPERATIONAL PROCEDURES, ETC.
- START NOW - USE COMMANDS THAT ARE COMMON ON BOTH SYSTEMS. I.E. USE ROUTE NOT BATCH OR DISPOSE.
- START NOW - USE CCL PROCEDURES FOR MUNDANE TASKS. CHANGE CONTENT OF PROCEDURE, IF NECESSARY, WHEN YOU CHANGE SYSTEMS.
- DON'T TRY TO USE ALL OF THE UTILITIES. A SMALL NUMBER WILL FILL MOST USERS NEEDS. SOME 'OLD' UTILITIES EXIST FOR HYSTERICAL REASONS (IN ADDITION TO HISTORICAL REASONS).
- LEARN THE UTILITIES A LITTLE EACH WEEK.
- CHOOSE GENERAL PURPOSE COMMANDS OVER SINGLE PURPOSE COMMANDS.
- COMPARE TASKS. DON'T COMPARE FEATURES.
- IDENTIFY CURRENT COMPASS MACRO USE. PLAN YOUR ATTACH. CHECK INTO COMMON COMMON DECKS.

## EASING THE TRANSITION - CONTINUED

- PLAN FOR ZERO LOCAL MODS.
- GET ACCOUNTING READY. SOME JOB COSTS WILL GO UP WHILE OTHERS WILL GO DOWN.
- COMPUTER POLICY WILL CHANGE. INFORM USERS.
- TRADE SOMETHING (NEW FEATURES, NEW HARDWARE,ETC.)
- PEOPLE ARE STILL AFRAID OF THE UNKNOWN.
- THE OPERATING SYSTEM MIGRATION WILL BE EASIER THAN THE CONVERSION FROM SCOPE 3.3 ⟶ SCOPE 3.4.
- OPPORTUNITY TO REVIEW OVERALL SYSTEMS, OPERATIONAL PROCEDURES, ETC.

STUDENT HANDOUT

TABLE OF CONTENTS

# GENERAL COURSE DESCRIPTION

## COURSE TITLE

NOS/BE to NOS Usage Difference

## COURSE NUMBER

## COURSE LENGTH

Three days (lecture and laboratory).

## DESCRIPTION

This course is designed to provide usage information for the computer user who desires to convert from NOS/BE to NOS. It includes a history of NOS/BE and NOS, characteristics of NOS and system access. The main part of the course contains detailed information on how to do NOS/BE tasks on NOS such as file retention, magnetic tape usage, source usage, object usage and interactive usage (time sharing, teleprocessing, and remote batch). The course concludes with suggestions on how to ease the migration.

## PREREQUISITES

The student should currently be a programmer/analyst on NOS/BE. The course presumes a moderate level of understanding and usage experience with NOS/BE batch and interactive.

NOS/BE TO NOS USAGE DIFFERENCES

COURSE CHART

| HOUR | DAY 1 | DAY 2 | DAY 3 |
|---|---|---|---|
| 1 | ADMINISTRATIVE DETAILS<br>NOS HISTORY<br>NOS CHARACTERISTICS | REVIEW<br>MAGNETIC TAPES | REVIEW<br>TEXT EDITORS |
| 2 | SYSTEM ACCESS | SOURCE USAGE/MAINTENANCE<br>OBJECT USAGE, STORAGE, MAINTENANCE | LABORATORY FOR TIME SHARING AND TEXT EDITING |
| 3 | BATCH ACCESS | | |
| 4 | PERMANENT FILES | LABORATORY - SOURCE AND OBJECT USAGE AND MAINTENANCE | REMOTE BATCH<br>TELEPROCESSING (QTRM) |
| 5 | LABORATORY-BATCH ACCESS AND PERMANENT FILES | INTERACTIVE OVERVIEW | TRANSACTION PROCESSING<br>HINTS TO EASE THE TRANSITION |
| 6 | | TIME SHARING | CRITIQUE |

# COURSE OUTLINE
## NOS/BE to NOS USAGE DIFFERENCES

1. NOS History and Philosophy (½ hr.)

2. NOS Characteristics (½ hr.)

   A. Validation

   B. Security

   C. Ownership of Resources

   D. Simplicity

3. System Access (1 hr.)

   A. General

   B. Families

   C. User Indicies

4. Batch (½ hr.)

   A. Access

   B. File Structure

   C. Special Considerations

5. Permanent Files (1½ hr.)

   A. Ownership

   B. Types

   C. Categories

   D. Access Modes

   E. Commands

   F. Special Considerations

6. Magnetic Tape (½ hr.)

   A. Reservations

   B. Assignment

   C. Release

7. Source (½ hr.)

   A. Storage Techniques

   B. Usage

   C. Maintenance

8. Object Module (1½ hr.)

    A. Storage Techniques<br>
    B. Usage
    C. Maintenance
    D. Special Considerations

9. Interactive Overview (½ hr.)

10. Time Sharing (1 hr.)

    A. Access
    B. General
    C. Special Considerations

11. Text Editors (½ hr.)

12. Remote Batch (½ hr.)

13. Teleprocessing (½ hr.)

14. Transaction Processing

15. Easing the Transition (1 hr.)

# LESSON 1

# NOS HISTORY AND PHILOSOPHY

## LESSON PREVIEW

This lesson introduces the student to the course, instructor, history of CYBER operating systems and the NOS philosophy.

## OBJECTIVES

After completing this lesson the student should -

- Understand the history of NOS.
- Understand the basic philosophy of NOS.

## REFERENCES

None.

## NOS/BE TO NOS USAGE MIGRATION

## CONTENT

- BRIEF HISTORY AND ARCHITECTURE OF NOS
- SYSTEM ACCESS
- PERMANENT FILES
- MAGNETIC TAPES
- SOURCE STORAGE, USAGE AND MAINTENANCE
- OBJECT STORAGE, USAGE AND MAINTENANCE
- INTERACTIVE, REMOTE BATCH AND TELEPROCESSING USAGE
- SOURCE EDITORS
- PLANNING FOR THE TRANSITION

# NOS/BE TO NOS USAGE MIGRATION

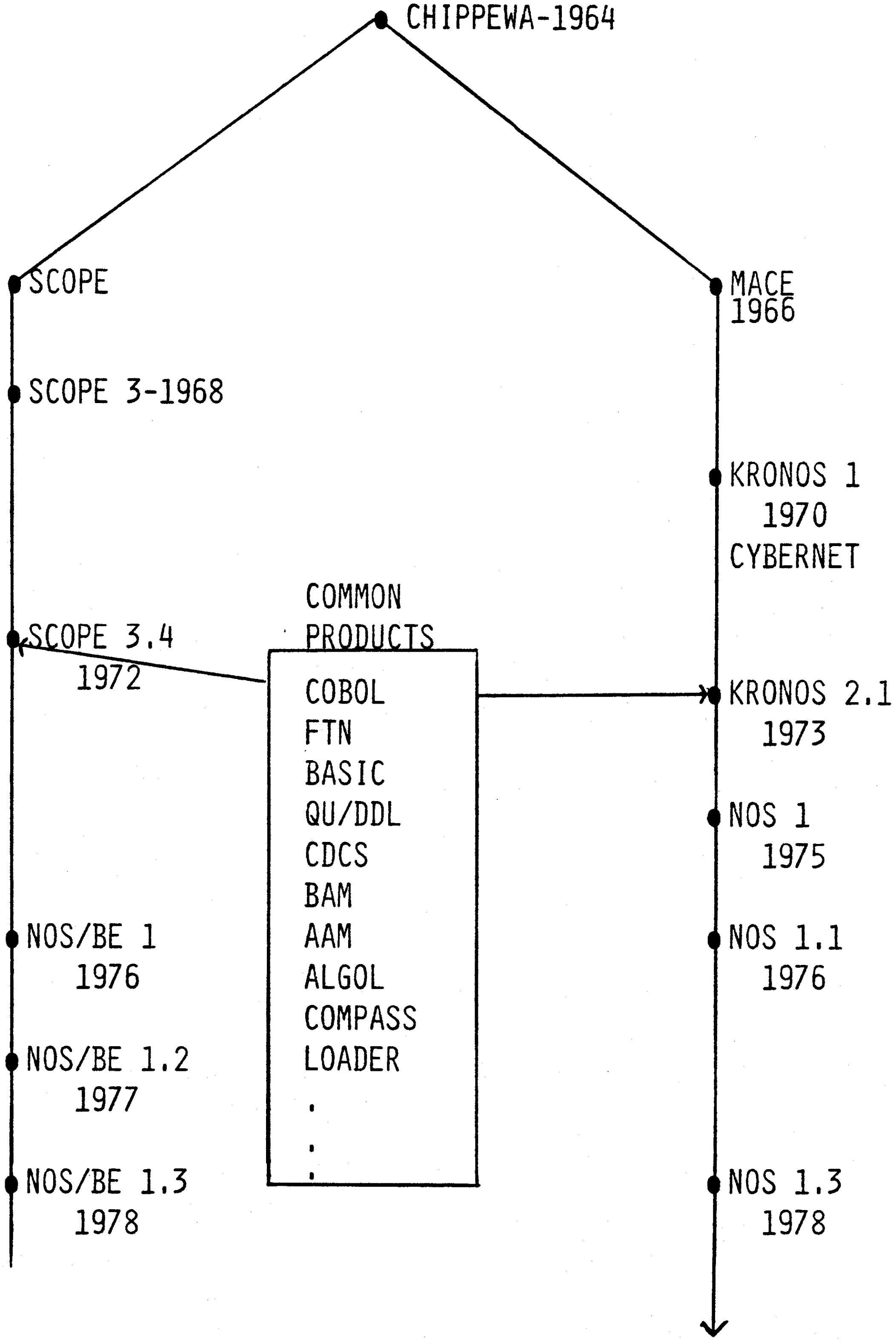
- SYSTEM USAGE - 98% OF PEOPLE AT A SITE.

  THE SUBJECT OF THIS SESSION.

- SYSTEMS ANALYST - 1% OF PEOPLE AT A SITE.

- OPERATOR TRAINING - 1% OF PEOPLE AT A SITE.

- CONVERSION AIDS


CHIPPEWA-1964
SCOPE
MACE
1966
SCOPE 3-1968
KRONOS 1
1970
CYBERNET
COMMON
PRODUCTS
SCOPE 3.4
1972
COBOL
FTN
BASIC
QU/DDL
CDCS
BAM
AAM
ALGOL
COMPASS
LOADER
.
.
.
KRONOS 2.1
1973
NOS 1
1975
NOS/BE 1
1976
NOS 1.1
1976
NOS/BE 1.2
1977
NOS/BE 1.3
1978
NOS 1.3
1978

COMPUTER IS USED FOR-

- AUTOMOTIVE
- BANKING
- CONSTRUCTION
- EDUCATION
- PETROLEUM
- WEATHER
- ETC.

COMPUTER IS NOT USED FOR -

- CPU MTR
- FANCY OPERATOR DISPLAYS
- BUILDING SYSTEMS
- ETC.

CDC COMPUTERS ARE FOR SOLUTIONS NOT MACHINE CYCLES.

## APPLICATIONS REQUIRE -

- COMMON PRODUCTS - COBOL, FTN, CRM, ETC.
- VEHICLE TO RUN COMMON PRODUCTS
- "SERVICE" FACILITIES

  FILE RETENTION
  ACCOUNTING
  ETC.

# LESSON 2

# NOS CHARACTERISTICS

## LESSON PREVIEW

This lesson describes the characteristics of NOS. These characteristics help explain why the operating system is put together the way it is.

## OBJECTIVES

1. Define responsibilities of user, operator and operating system.

2. Describe resource ownership, validation, etc.

## REFERENCES

None.

## NOS CHARACTERISTICS

- SYSTEM CONTROL THRU VALIDATION

  AVOID WASTE OF RESOURCES.
  PROTECTION OF USER'S RESOURCES.

- SECURITY - ACCESS TO PFS, ACCOUNTING, ETC., IS THRU A USER CONTROLLED PASSWORD.

- EVERYTHING BELONGS TO A 'USER'.

- RELIABILITY - CURRENT FAILURE RATE IS LESS THAN 1 PER MONTH. RESULT OF MAINTAINABILITY IMPROVEMENTS THROUGHOUT THE ENTIRE PSR LEVEL 400 SERIES.

- SIMPLICITY - MORE COMMANDS; FEWER PARAMETERS. VERY FEW 'INPUT' DIRECTIVES.

- RESOURCE OWNERSHIP - THERE IS A 1 TO 1 RELATIONSHIP BETWEEN PEOPLE AND USER NUMBERS.

- OPERATOR DOES NOT MAKE USER DECISIONS.

- MANY WAYS TO DO A TASK.

  CHOOSE ONE - I'LL HELP.

# LESSON 3

## SYSTEM ACCESS

### LESSON PREVIEW

This lesson describes how the user gains access to NOS batch, remote batch, interactive, etc. The lesson also describes the FAMILY/USER concept.

### OBJECTIVES

1. Understand the concept of a FAMILY, user index and user name.

2. Understand how files are allocated to physical devices.

3. Review basic file structure.

### REFERENCES

NOS Reference Manual Vol. 1 Ch. 6 - USER Statement.

## SYSTEM ACCESS

### USER STATEMENT

- COMMON AND REQUIRED FOR ALL ACCESS

  BATCH
  INTERACTIVE
  REMOTE BATCH
  TELEPROCESSING
  TRANSACTION

- PASSWORD PROTECTION CONTROLLED BY USER

- PERMANENT FILE ACCESS

- RESOURCE LIMITS

- BILLING (ACCOUNTING)

- SECONDARY USER STATEMENTS CAN BE ENTERED IN THE MIDDLE OF A SESSION (SITE MUST CHOOSE THIS OPTION). THIS FEATURE IS PRIMARILY FOR PERMANENT FILE ACCESS.

## FAMILIES AND USER INDICES

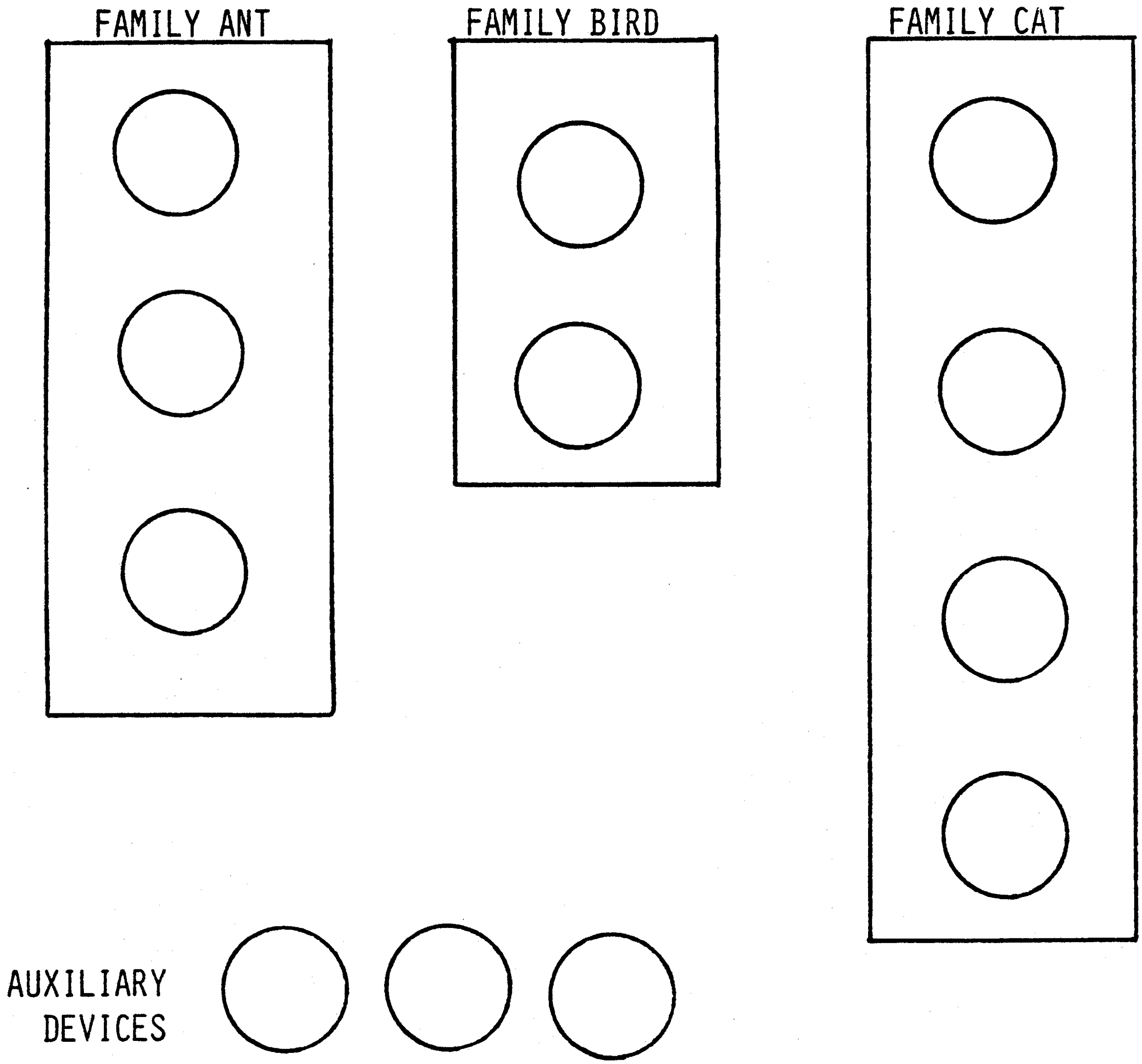
- FAMILY IS THE TECHNIQUE TO GROUP MEMBERS OF USER COMMUNITIES TOGETHER.

  CONTAINS THE VALIDATION FILE FOR USERS IN THAT FAMILY.

  PROVIDES A BACKUP FACILITY SINCE IT IS VERY EASY TO MOVE USERS FROM A DOWN MACHINE.

- USER INDEX IS THE TECHNIQUE TO SPREAD THE USERS IN A FAMILY OVER THE DISKS IN A FAMILY. THE DISK THAT A USER INDEX IS 'ASSIGNED' CONTAINS THAT USER'S -

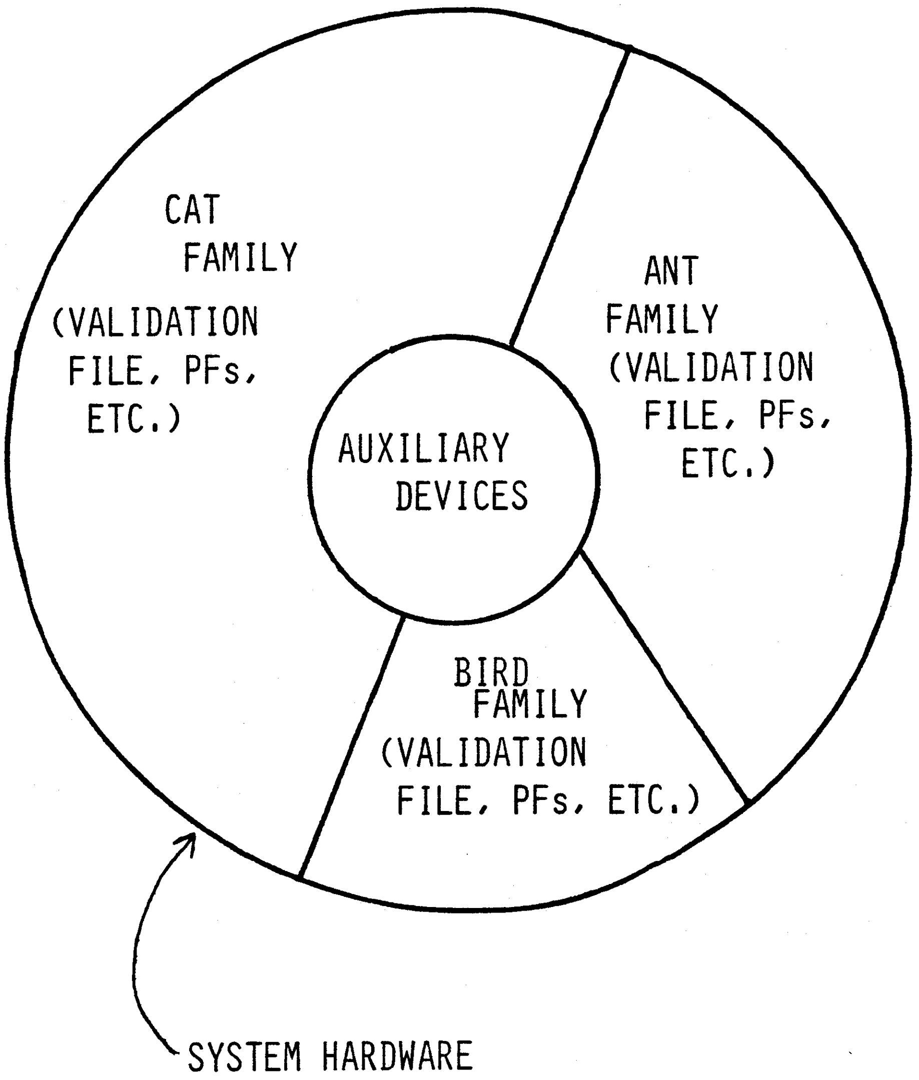
  PF CATALOG
  INDIRECT PFs
  SOME DIRECT PFs



EACH 'DEVICE' WITHIN A FAMILY OR AUXILIARY DEVICE MAY BE 1 TO 8 DISK SPINDLES.
EACH AUXILIARY DEVICE MAY BE DECLARED

-PRIVATE - ONLY VALIDATED USER CAN CREATE FILES.
ALTERNATE USERS CAN BE PERMITTED TO ACCESS FILES

-PUBLIC - ANY USER CAN CREATE OR ACCESS FILES


CAT FAMILY (VALIDATION FILE, PFs, ETC.)
ANT FAMILY (VALIDATION FILE, PFs, ETC.)
AUXILIARY DEVICES
BIRD FAMILY (VALIDATION FILE, PFs, ETC.)
SYSTEM HARDWARE

- HARDWARE
- FAMILY
- USER INDEX
- FILES

- AUXILIARY DEVICES

## FILE STRUCTURE

ESSENTIALLY THE SAME.

NOS/BE

| | EOR | EOF | EOI |
|---|---|---|---|
| CARDS | 7/8/9 | IMPOSSIBLE | 6/7/8/9 |
| DISK | SHORT PRU + LEVEL=0 | ZERO LENGTH PRU + LEVEL = 17 | END OF RBR CHAIN |
| TAPE (SI) | THE SAME AS DISK. | THE SAME AS DISK. | DOUBLE TAPEMARKS |

NOS

| | EOR | EOF | EOI |
|---|---|---|---|
| CARDS | 7/8/9 | 6/7/9 | 6/7/8/9 |
| DISK | SHORT PRU + LINK TO NEXT PRU. | ZERO LENGTH PRU + LINK TO NEXT PRU | ZERO LENGTH PRU NO LINK |
| TAPE (SI/I) | SHORT PRU + LEVEL = 0 | ZERO LENGTH PRU + LEVEL = 17 | DOUBLE TAPEMARKS |

- THERE ARE NO RANDOM BIT PROBLEMS ON NOS. ANY FILE (RANDOM OR SEQUENTIAL) MAY BE COPIED WITH THE COPYEI DIRECTIVE

# LESSON 4

## BATCH ACCESS/PROCESSING

### LESSON PREVIEW

This lesson introduces the student to batch access and processing (job structure) on NOS. It also equips the student to cope with the difference between NOS/BE and NOS batch.

### OBJECTIVES

1. Describe difference between NOS/BE and NOS job processing.
2. Describe Job naming conventions on NOS.
3. How to cope with deficiencies on NOS (job dependency, job names, etc.).

### REFERENCES

NOS Reference Manual Vol. 1 - Chapter 5.

V4-1

## BATCH ACCESS

```
FOX,T40.
USER,JFOX,WEIRD [,FAMILY].
CHARGE,MYDEPT,MYPROJECT.
:
```

- JOB STATEMENT - ONLY 3 PARAMETERS!

  T - TIME LIMIT IN CP SECONDS FOR EACH JOB STEP.

  CM - SPECIFIES MAXFL. NOT RECOMMENDED. MAXFL MUST SATISFY -

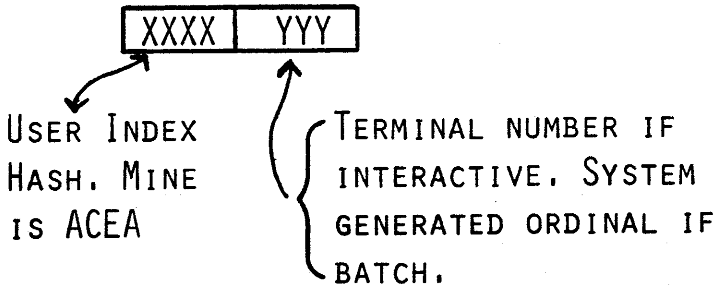
  RFL ≤ MFL ≤ MAXFL ≤ VALIDATED MAXIMUM.

  EC - EXTENDED CENTRAL STORAGE

IN NOS MANY OF THE NOS/BE JOB PARAMETERS ARE SPECIFIED BY CONTROL STATEMENTS WITHIN THE JOB WHEN NEEDED. THERE ARE GENERALLY MORE CONTROL STATEMENTS ON NOS WITH FEWER PARAMETERS.

## BATCH ACCESS - CONTINUED

- JOB NAME IS UNIQUE AND CONSISTS OF TWO PARTS-



USER INDEX HASH. MINE IS ACEA

TERMINAL NUMBER IF INTERACTIVE. SYSTEM GENERATED ORDINAL IF BATCH.

- FAMILY IS VALIDATION FILE TO CHECK FOR USER NAME.

- USER NAME (JFOX) IS THE OPERATIONS SUPPLIED NAME IN THE VALIDATION FILE. THIS IS ASSOCIATED WITH A USER INDEX THAT DETERMINES WHERE MY PF CATALOG AND FILES RESIDE.

- PASSWORD (WEIRD) IS SUPPLIED, CHANGED AND MAINTAINED BY ME (THE USER). I CAN CHANGE IT WHEN APPROPRIATE.

- DEPARTMENT (MYDEPT). EACH USER CAN BE VALIDATED TO CHARGE TO CERTAIN DEPARTMENTS AND PROJECTS WITHIN DEPARTMENTS.

## BATCH ACCESS

### SPECIAL CONSIDERATIONS

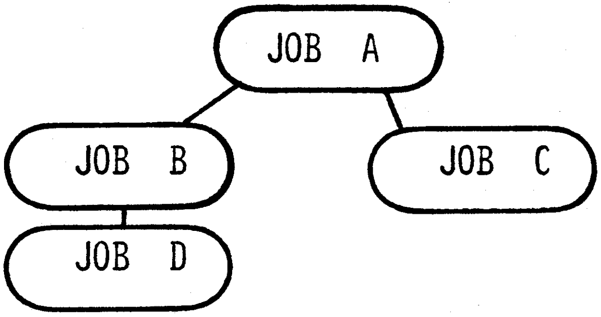
- RERUN<br>NORERUN - User can explicitly turn rerun flag on or off as desired in job. No strange PF tricks like on NOS/BE.

- EXIT<br>NOEXIT<br>ONEXIT - No EXIT(s) but facility to turn EXIT globally on or off

## BATCH ACCESS

### SPECIAL CONSIDERATIONS

- JOB NAME - CODE IS AVAILABLE VIA THE UNDERGROUND. 'JOB ADMINISTRATION' ADDRESSES THE ENTIRE PROBLEM AND IS SCHEDULED FOR A FUTURE RELEASE.



- JOB DEPENDENCIES - NO JOB STATEMENT PARAMETER. CAN USE ANY OF THE FOLLOWING AND AVOID WASTED NOS/BE FNT ENTRIES AND READING JOBS IN BACKWARD.

  - ROUTE OR LDI COMBINED WITH CCL IS A VERY POWERFUL COMBINATION.

  - PF INTERLOCKS COMBINED WITH ROLLOUT.

  - SPECIAL PRODUCTS LIBRARY.

    WAIT, RESUME, LISTDJ, DROP

# LESSON 5

# PERMANENT FILES

## LESSON PREVIEW

This lesson presents the permanent file concepts of NOS to the student.

## OBJECTIVES

1. Establish firm base of understanding permanent file concepts - ownership, types, categories, access modes and commands.

2. Provide work-arounds for NOS deficiencies (40 character file name, cycles, user dump/load, etc.).

## REFERENCES

NOS Reference Manual Vol. 1 - Chapter 8.

# PERMANENT FILES

## CHARACTERISTICS

- OWNERSHIP
- TYPES - DIRECT AND INDIRECT
- CATEGORIES - PUBLIC, SEMI-PRIVATE,
  PRIVATE
- ACCESS MODES
- COMMANDS

# PERMANENT FILES

## CHARACTERISTICS

- FILES BELONG TO SPECIFIC USERS.

  NOS IS SET UP TO KEEP USERS AWAY FROM EACH OTHER. IF YOU WANT TO SHARE FILES YOU MUST MAKE THEM PUBLIC OR PERMIT SPECIFIC USERS TO ACCESS THE FILE(S).

- TWO TYPES OF FILES - DIRECT AND INDIRECT

  WHEN AN INDIRECT FILE IS ASSIGNED TO A JOB YOU GET A COPY IN SCRATCH SPACE. WHEN A DIRECT FILE IS ASSIGNED TO A JOB YOU GET A REAL FILE.

  DIRECT FILES TAKE UP 'MORE' DISK SPACE- THE OLD RECORD BLOCK PROBLEM. INDIRECT PERMANENT FILE SPACE IS MAINTAINED IN INCREMENTS OF PRUs BY NOS. DIRECT FILES ARE IDEAL FOR LARGE DATA FILES WHILE INDIRECT FILES ARE IDEAL FOR SMALL (LESS THAN A LOGICAL TRACK) DATA FILES.

## PERMANENT FILES CHARACTERISTICS - CONTINUED

- THREE CATEGORIES

PUBLIC - ANYONE WHO KNOWS USER-NAME, FILE NAME AND PASSWORD IF SPECIFIED CAN ACCESS.

SEMI-PRIVATE - SAME AS PUBLIC EXCEPT THAT A RECORD IS KEPT OF WHO AND WHEN THE FILE WAS ACCESSED.

PRIVATE - ONLY YOU CAN ACCESS OR SPECIFIC USERS PERMITED FOR ONLY SPECIFIC ACCESS MODE.

## PERMANENT FILE CHARACTERISTICS - continued

- Eight Access MODEs. Each mode is 'higher' than previous.

N - NULL<br>
E - EXECUTE ONLY<br>
R - READ ONLY<br>
RA - READ and/or APPEND<br>
RM - READ and/or MODIFY<br>
A - APPEND<br>
M - MODIFY<br>
W - WRITE

## PERMANENT FILE CHARACTERISTICS - CONTINUED

- COMMANDS

| | | | |
|---|---|---|---|
| DIRECT AND INDIRECT | CATLIST | - | SORTED LIST OF PFS AND MISC. INFO. LIKE AUDIT EXCEPT MUCH MORE USER FRIENDLY! |
| | PURGE | - | REMOVE A FILE FROM PF CATALOG. |
| | PERMIT | - | ALLOW OTHER USERS TO ACCESS. |
| | CHANGE | - | CHANGE PERMISSIONS. |
| | PURGALL | - | PURGE ALL FILES! |
| DIRECT ONLY | DEFINE | - | ESTABLISH A DIRECT FILE. |
| | ATTACH | - | ACCESS A DIRECT FILE. |
| INDIRECT ONLY | SAVE | - | ESTABLISH AN INDIRECT FILE. |
| | GET | - | ACCESS AN INDIRECT FILE. |
| | REPLACE | - | REPLACE AN EXISTING INDIRECT FILE. |
| | APPEND | - | ADD A LOCAL FILE TO THE END OF AN INDIRECT FILE. |

## DIRECT FILES

### CREATION

```
DEFINE,[LFN=]PFN[/CT=CATEGORY,M=MODE,
                  PW=PASSWORD].
```

PLACE WHERE YOU WOULD NORMALLY INSERT THE NOS/BE 'REQUEST,LFN,*PF' STATEMENT.

'CATEGORY' MAY BE

P - PRIVATE
S - SEMI-PRIVATE
PU - PUBLIC

## DIRECT FILES - CONTINUED

THE MEANING OF THE 'MODE' PARAMETER DEPENDS UPON THE CATEGORY OF THE FILE -

PRIVATE - MODE HAS NO MEANING SINCE OWNER CAN ACCESS AT ANY LEVEL AND EACH ALTERNATE USER MUST EXPLICITLY BE PERMITTED TO ACCESS THE FILE.

PUBLIC AND SEMI-PRIVATE - MODE IS THE MAXIMUM ACCESS MODE FOR ALL ALTERNATE USERS EXCEPT THOSE WHO HAVE BEEN EXPLICITLY PERMITTED BY THE OWNER. AN EXPLICIT PERMIT TAKES PRECEDENCE OVER MODE ON THE DEFINE FOR PERMITTED USERS.

## DIRECT FILES - CONTINUED

### ACCESS

```
ATTACH,[LFN=]PFN[/UN=USER-NAME,M=MODE,
                        PW=PASSWORD].
```

'USER-NAME' IS OPTIONAL IF PFN IS YOUR FILE.

'MODE' IS THE ACCESS MODE DESIRED FOR THIS ACCESS. IF AN ALTERNATE USER ATTEMPTS ACCESS OF A PERMITED PRIVATE FILE AND EXCEEDS THE PERMITED MODE, THE FOLLOWING MESSAGE IS DISPLAYED -

```
PFN NOT FOUND
```

THE SAME THING HAPPENS FOR PUBLIC AND SEMI-PRIVATE FILES WHEN THE AUTHORIZED MODE IS EXCEEDED.

DEFAULT 'MODE' IS READ IF M IS NOT SPECIFIED.

## INDIRECT FILES

### CREATION

```
SAVE,[LFN=]PFN[/CT=CATEGORY,M=MODE,
                PW=PASSWORD].
```

COPIES LFN TO AREA NOW CALLED PFN.

```
REPLACE,[LFN=]PFN[/UN=USER-NAME,PW=PASSWORD].
```

REPLACE DOES AN IMPLICIT PURGE AND SAVE.
IT IS VERY DIFFICULT TO GO BACK TO NOS/BE
AND HASSLE CYCLES AFTER USING THIS COMMAND.

## INDIRECT FILES - CONTINUED

### ACCESS

```
GET,[LFN=]PFN[/UN=USER-NAME,PW=PASSWORD].
```

COPIES PFN TO LFN.

### EXTENSION

```
APPEND,PFN,LFN1,LFN2...[/UN=USER-NAME,
                        PW=PASSWORD].
```

COPIES LFN1, LFN2, ETC., TO THE END OF INFORMATION OF PFN.

## PERMANENT FILES

### SPECIAL CONSIDERATIONS

- NO REQUEST,LFN,*PF
- SEQUENTIAL REWRITE IS AUTOMATIC
- OVERWRITE IS AUTOMATIC
- EXTEND IS ALWAYS AUTOMATIC ON DIRECT FILES
- NO PFD OVERFLOW PROBLEMS
- ALTER

  To TRUNCATE FILE USE WRITER,LFN.

  To 'REDEFINE' FILE JUST REWIND AND START WRITING.
- CATLIST IS 'TERMINAL-FRIENDLY'

  FN
  UNIVERSAL CHARACTER
  LO
  LENGTH STATEMENT

## PERMANENT FILES

### SPECIAL CONSIDERATIONS - CONTINUED

- REPLACE IS VERY EASY TO LOVE.
- RANDOM FILES CAN BE COPIED DISK TO DISK (NO RANDOM BIT PROBLEM) AND BE USED. RANDOM FILES CAN ALSO BE COPIED TO TAPE FOR BACKUP PURPOSES AND COPIED BACK TO DISK FOR USE.
- 40 CHARACTER PFN - BITE THE BULLET
- USER DUMP/LOAD
  - IF THE RANDOM BIT IS THE PROBLEM ON NOS/BE; THERE IS NO PROBLEM ON NOS. USE COPYEI.
- ARCHIVING

  SPL - SPECIAL PRODUCTS LIBRARY.
  FUTURE SYSTEM RELEASE (1ST HALF 80).
- CYCLES - MUST BE NICE TO HAVE THAT MUCH DISK SPACE...
  - REWIND THEN WRITE
  - REPLACE
  - USE CCL PROC

```
.PROC,NEWDEF,LOCAL,PERM.

 PURGE,PERM→1/NA.                   Purge oldest

 CHANGE,PERM→1 = PERM→2/NA.         2 to oldest

 CHANGE,PERM→2 = PERM→3/NA.         3 to 2

 DEFINE,LOCAL = PERM→3.             New to 3
```

```
NEWDEF,PUPPY,DOG.
```

NOTES -

```
        CHANGE,NEW=OLD/NA.

        DOG1 = OLDEST
        DOG2 =

        DOG3 = NEWEST
```

# PERMANENT FILES

## SPECIAL CONSIDERATIONS - CONTINUED

- VERY LARGE FILES - USE POSITIVE CONTROL (KNOW WHERE THE FILE IS LOCATED!!).

  - USE AUXILARY MULTI-SPINDLE LOGICAL DEVICE PRIVATE PACKS.

    PACKNAM

  - USE MULTI-SPINDLE DEVICE FAMILIES.

  EITHER WILL RESULT IN BETTER ACCESS.

- MACROS - WATCH OUT FOR PF FET - CRM FIT CONFLICT.

# LESSON 6

# MAGNETIC TAPE USAGE

## LESSON PREVIEW

This lesson explains magnetic tape reservation, assignment and release on NOS.

## OBJECTIVES

1. Understand the difference between NOS/BE and NOS tape reservation.

2. Understand VSN label standard differences.

## REFERENCES

NOS Reference Manual Vol. 1 - Chapter 10

V6-1

## MAGNETIC TAPE

- RESERVATION
- ASSIGNMENT
- RELEASE
- DIFFERENCES

## MAGNETIC TAPE

- RESERVATION

IF ONLY ONE DEVICE (7 TRACK DRIVE OR 9 TRACK DRIVE OR DISK DRIVE) IS REQUIRED NO RESERVATION IS REQUIRED.

IF MORE THAN ONE DEVICE IS REQUIRED THEN A RESOURC STATEMENT MUST BE ENTERED IN THE JOB STREAM BEFORE THE FIRST DEVICE IS ASSIGNED.

```
RESOURC(MT=2,NT=1)
```

THIS STATEMENT INDICATES THAT THE JOB NEEDS TO USE TWO 7 TRACK DRIVES AND ONE 9 TRACK DRIVE AT THE SAME TIME. THE NOS/BE USER WOULD HAVE ENTERED -

```
...MT2,NT1...
```

ON THE JOB STATEMENT.

THE COUNTS MAY BE LOWERED BY ANOTHER RESOURC STATEMENT.

## MAGNETIC TAPE

- ASSIGNMENT

Use the LABEL statement. It does everything. Most of the parameters are identical to NOS/BEs LABEL parameters.

```
LABEL,lfn,
   D = den,      den may be HI, HY, HD, GE, etc.
   F = fmt,      Data format - I,S,L,SI
                 (SI is a NOS/BE System
                 Internal).
  LB = type,     Label type may be -
                 KL - NOS labeled (ANSI)
                 KU - Unlabeled
                 NS - Non-standard
 VSN = vsn,
  PO = option,   Processing option A,B,F,R,W,etc.
   R,            Read Existing label
  W)             Write new label
```

There are other parameters listed in the NOS R.M. Notice that there is no RING/NORING parameter. This is accomplished by the PO=W and PO=R options. PO also has many other options (parity error processing, etc.).

## MAGNETIC TAPE

THE VSN PARAMETER APPEARS INNOCENT BUT THERE IS ONE SITUATION TO BE AWARE OF -

NOS/BE IMPLEMENTED AN OLD ANSI LABEL STANDARD WHILE NOS IMPLEMENTED THE NEW ANSI STANDARD. THE VSN IS LEFT JUSTIFIED WITH BLANK FILL ON NOS WHILE IT IS RIGHT JUSTIFIED WITH ZERO FILL ON NOS/BE. TO READ A NOS/BE TAPE ON NOS, SPECIFY THE VSN AS A SIX CHARACTER FIELD WITH LEADING ZEROS.

VSN=4567 ON NOS/BE WOULD BE
VSN=004567 ON NOS

- RELEASE

RETURN AND UNLOAD OPERATE THE SAME ON NOS AND NOS/BE.

# LESSON 7

## SOURCE STORAGE, USAGE AND MAINTENANCE

### LESSON PREVIEW

This lesson introduces the student to the source maintenance facilities available on NOS.

### OBJECTIVES

1. Identify source maintenance facilities (UPDATE, MODIFY, XEDIT).

2. Introduce systems analysts to MODIFY for operating system modules.

### REFERENCES

None.

## SOURCE PROGRAM LIBRARIES

- UPDATE - It is the same on both systems. You know how to use it. Continue to use it. PLs can be COPYEIed since there is no random bit problem. Future development is scheduled for UPDATE, e.g. 8 bit support.

- MODIFY - Has similar features as UPDATE. Cannot be used when the PL resides on tape. Operating system software is maintained by MODIFY.

- XEDIT - An editor that works both in batch and interactive. More later in this presentation.

# LESSON 8

## OBJECT MODULE STORAGE, USAGE AND MAINTENANCE

### LESSON PREVIEW

This lesson provides the student with a firm grasp of how to work with the various types of object modules on NOS.

### OBJECTIVES

1. Describe storage techniques - sequential files and random user libraries.
2. Describe what can be stored in <u>and</u> accessed from user libraries.
3. Describe work-arounds for using absolutes and procedure files on NOS.
4. Describe maintenance techniques available on NOS (COPYL, GTR, LIBEDIT, LIBGEN, etc.).
5. Explain NOS/BE EDITLIB work-arounds that are available.

### REFERENCES

NOS Reference Manual Vol. 1 - Chapter 7

## OBJECT MODULE 'LIBRARIES'

- STORAGE

  SEQUENTIAL FILES<br>
  RANDOM 'LIBRARY' FILES USABLE BY THE
  CYBER LOADER

- USAGE

- MAINTENANCE

## OBJECT MODULE LIBRARIES - CONTINUED

## USAGE

- SEQUENTIAL FILES OF RELOCATABLES OR ABSOLUTES PERFORM THE SAME WAY ON BOTH SYSTEMS.
- CAPSULES ARE STORED IN RANDOM LIBRARIES ON BOTH SYSTEMS. THEY ARE ACCESSED IN THE SAME WAY ON BOTH SYSTEMS.
- ABSOLUTES CURRENTLY CANNOT BE STORED IN NOS USER LIBRARIES.
    - IF PURPOSE OF INSERTING ABSOLUTES INTO A USER LIBRARY ON NOS/BE WAS TO SAVE DISK ACCESSES THEN FOL IS FASTER THAN NON-FOL NOS/BE LIBRARIES.
    - GTR (GET RECORDS) CAN BE USED TO ACCOMPLISH SOME FUNCTIONS OF A GLOBAL LIBRARY STATEMENT.
    - CODE IS AVAILABLE (VIA THE UNDERGROUND) FOR LIBGEN. IT MIGHT BE STANDARD WHEN YOU CONVERT.

## OBJECT MODULE LIBRARIES

### USAGE - CONTINUED

- RELOCATABLE MAIN PROGRAMS CAN BE STORED IN NOS USER LIBRARIES BUT ONLY THE CYBER LOADER KNOWS HOW TO READ GLOBAL LIBRARIES. NOS DOES NOT SEARCH GLOBAL LIBRARIES IN RESPONSE TO A NAME CALL LOAD. NOS DOES NOT INVOKE THE CYBER LOADER UNLESS EXPLICITLY CALLED. TRY THE FOLLOWING -

  ```
  LDSET,LIB=LIBNAME.
  MYPROG.
  ```

  CPUMTR (1AJ) CODE IS AVAILABLE VIA THE UNDERGROUND.

- CCL PROCEDURES (THE ONLY KIND) CURRENTLY CANNOT BE STORED IN NOS USER LIBRARIES.

  - PLACE YOUR PROCEDURES INTO AN INDIRECT PF CALLED PROCFIL. ENTER '-PROC-NAME,,P1,P2...' NOS (CCL) WILL DO AN AUTOMATIC GET OF THE PROPER PROCEDURE IN INTERACTIVE.

    'BEGIN,PROC-NAME,,P1,P2... IS REQUIRED IN BATCH.

## OBJECT MODULE LIBRARIES

## MAINTENANCE

- SEQUENTIAL FILES

  - COPYL - THE SAME
  - COPYLM - THE SAME
  - ITEMIZE- THE SAME
  - GTR - VERY HANDY EXTRACTION UTILITY!
  - LIBEDIT- FANCY, DIRECTIVE DRIVEN COPYL. CREATES A NEW 'CLEAN' FILE.

- RANDOM FILES (USER LIBRARIES)

  ONLY UTILITY IS LIBGEN. IT COPIES A CLEAN SEQUENTIAL FILE TO USER LIBRARY FORMAT.

  - USE A CCL PROCEDURE TO COMBINE THE MUNDANE TASKS OF GTR, LIBEDIT AND LIBGEN. RESULTS IN NO NOS/BE-LIKE DEAD DISK SPACE!

## OBJECT MODULE LIBRARIES

## MAINTENANCE

CONSIDER THE FOLLOWING TWO CCL PROCEDURES - ONE TO GENERATE AN 'EMPTY' USER LIBRARY AND ANOTHER TO MAINTAIN THE LIBRARY.

```
.PROC,EMPTYLB,LIBNAME,INP=#DATA.

RETURN,TEMP.

FTN,I=INP,L=0,B=TEMP.

LIBGEN,F=TEMP,P=LIBNAME.

RETURN,TEMP.

.DATA

       SUBROUTINE   AAAAAAA

       END
```

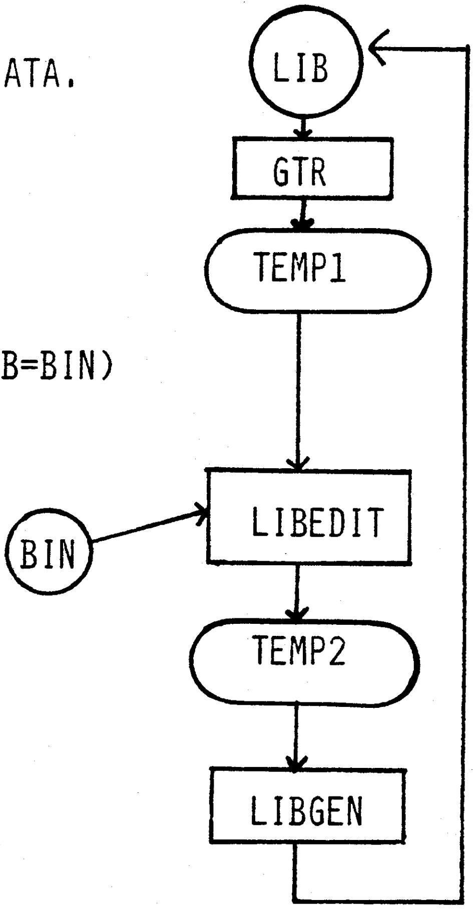
THIS PROCEDURE IS ACCESSED BY -

```
     EMPTYLB,MYLIB.
```

## OBJECT MODULE LIBRARIES

### MAINTENANCE - CONTINUED

```
 .PROC,UPLIB,LIBNAME,BIN,INP=#DATA.

 RETURN(TEMP1,TEMP2)

 GTR(LIBNAME,TEMP1)ULIB/LIBNAME

 LIBEDIT(I=INP,P=TEMP1,N=TEMP2,B=BIN)

 LIBGEN(F=TEMP2,P=LIBNAME)

 .DATA

*INSERT,REL/AAAAAAA,REL/*
```



THE PROCEDURE IS ACCESSED BY -

```
        UPLIB,MYLIB,MYLGO.
```

## OBJECT MODULE LIBRARIES

## MAINTENANCE - CONTINUED

GTR - A VERY POWERFUL UTILITY.

```
GTR,OLD-FILE,NEW-FILE. TYPE/NAMES
```

GTR SCANS 'OLD-FILE' FOR THE RECORD(S) CALLED 'NAMES' OF THE SPECIFIED 'TYPE' (REL,ABS,ETC.) AND COPIES THE RECORDS TO 'NEW-FILE'.

# LESSON 9

# INTERACTIVE OVERVIEW

## LESSON PREVIEW

This lesson provides an overview of the interactive facilities that are available on NOS.

## OBJECTIVES

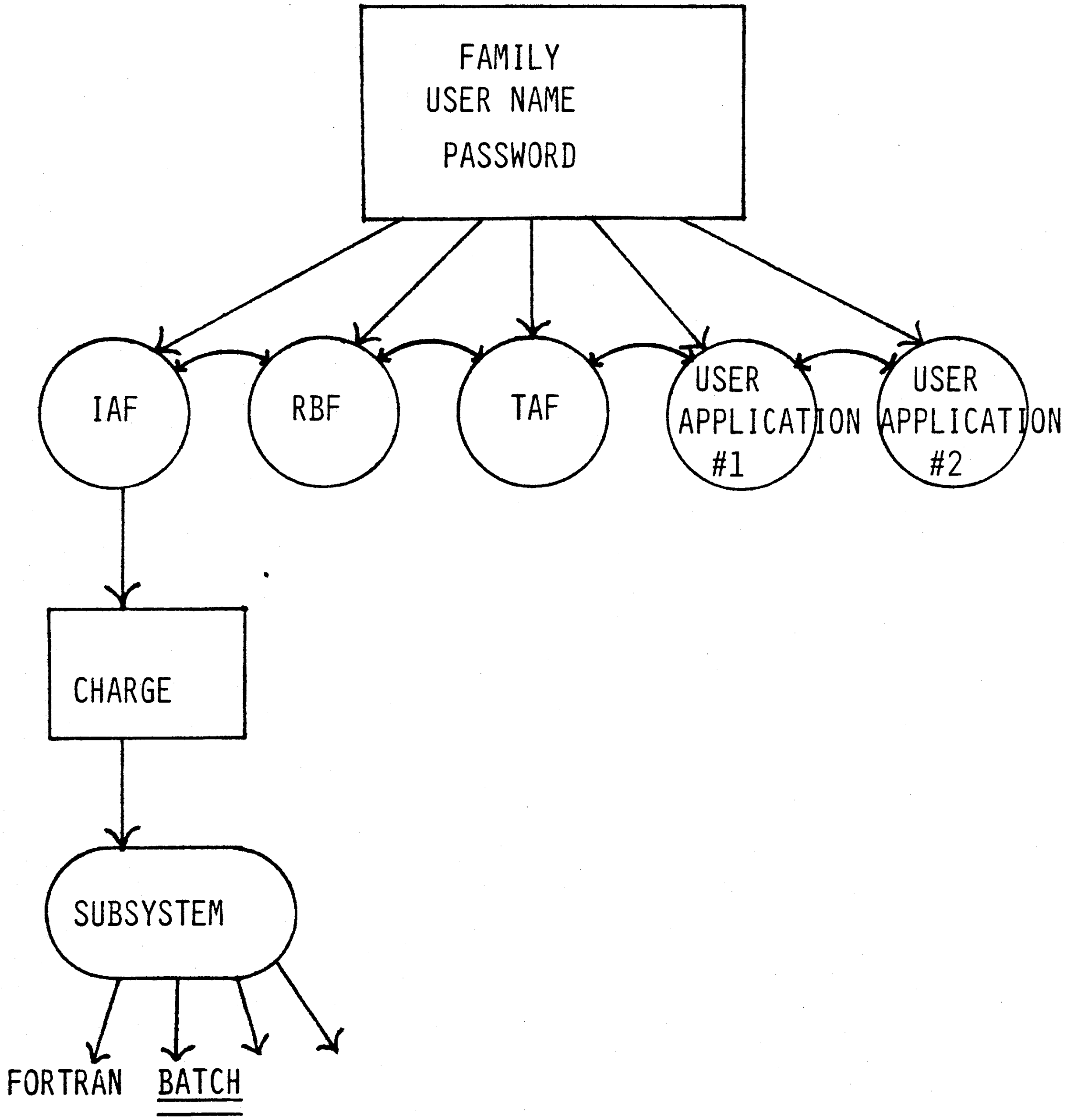
Provide an overview of the interactive facilities and capabilities that are available on NOS - time sharing, remote batch, teleprocessing and transaction processing.

## REFERENCES

None.

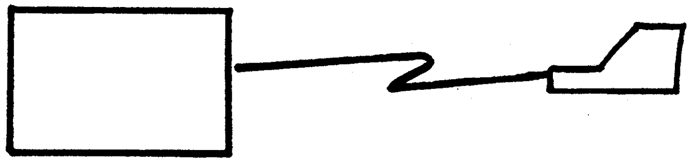
V9-1

## INTERACTIVE

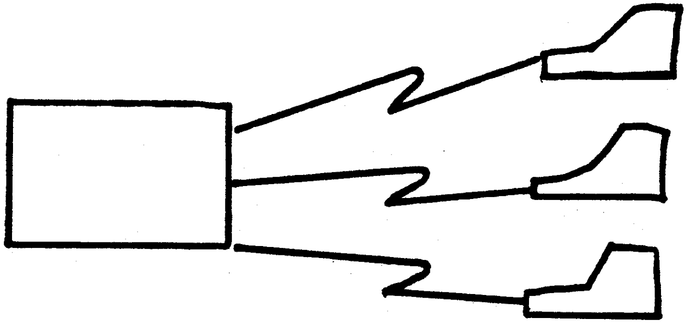


## TYPES OF INTERACTIVE PROCESSING

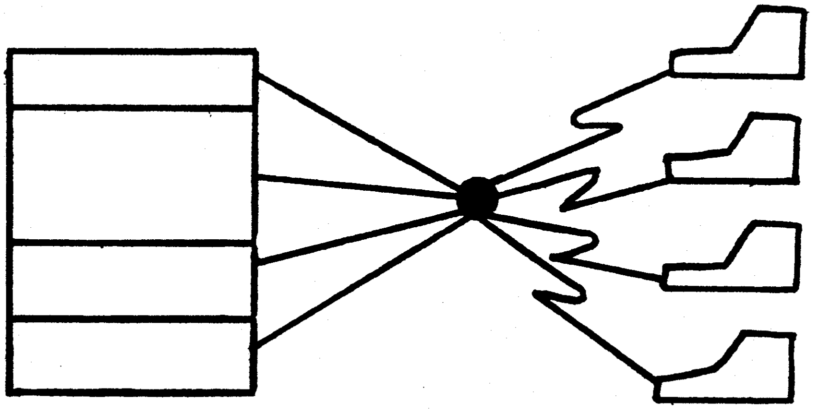
- TIME SHARING (IAF)
  ONE PROGRAM - ONE TERMINAL

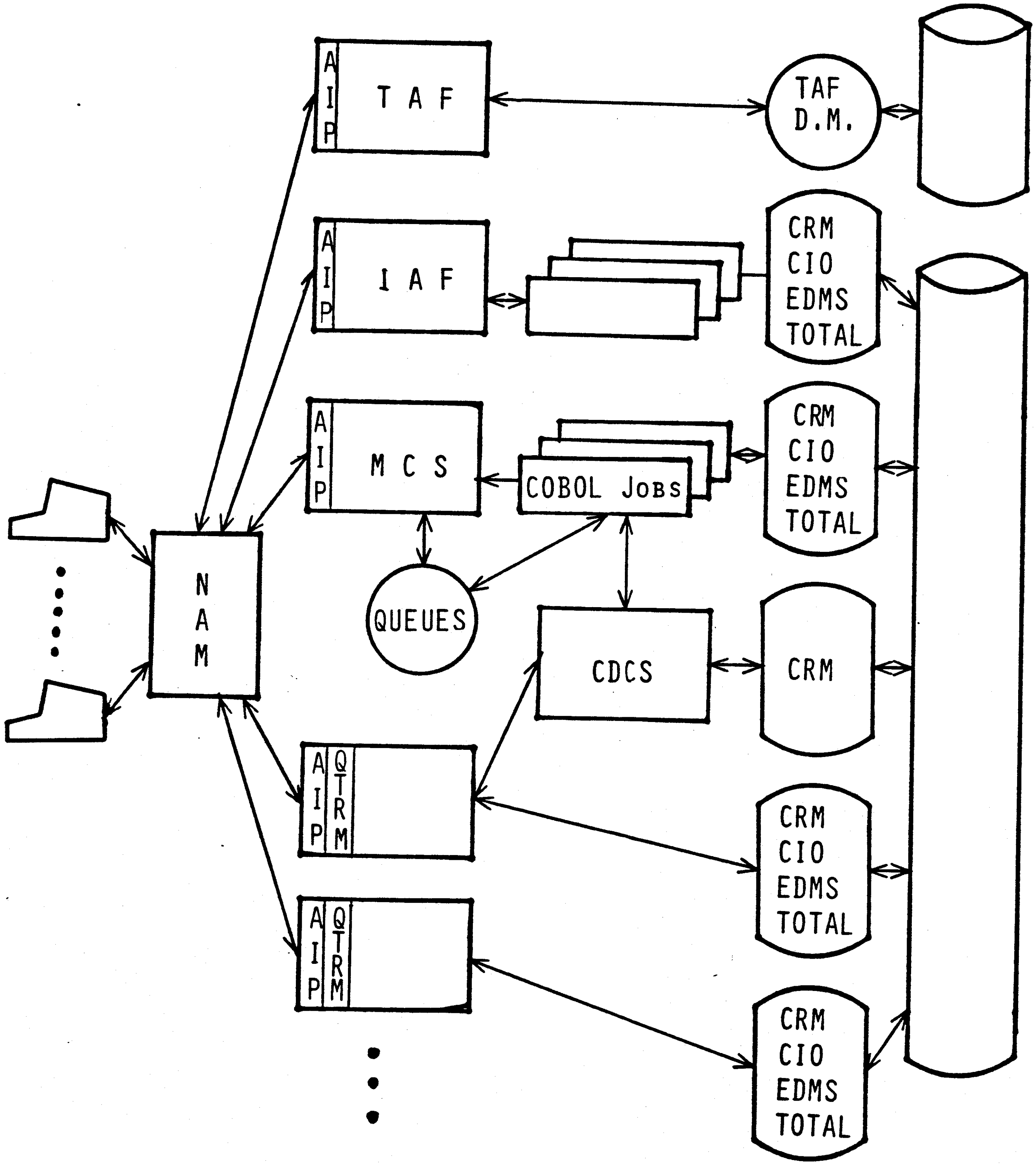


- TELEPROCESSING (QTRM)
  ONE PROGRAM - MANY TERMINALS



- TRANSACTION (TAF)
  MANY TASKS - MANY TERMINALS




AIP
TAF
TAF D.M.
AIP
IAF
CRM CIO EDMS TOTAL
AIP
MCS
COBOL JOBS
CRM CIO EDMS TOTAL
NAM
QUEUES
CDCS
CRM
AIP QTRM
CRM CIO EDMS TOTAL
AIP QTRM
CRM CIO EDMS TOTAL

# LESSON 10

# TIME SHARING

## LESSON PREVIEW

This lesson provides the student with general and specific knowledge necessary to work with interactive time sharing on NOS.

## OBJECTIVES

1. Describe logon and general procedures.
2. Identify terminal I/O differences (prompts, 'connected' files, etc.).
3. Identify new commands - DAYFILE, ENQUIRE, etc.
4. Describe how to use the CYBER Loader in interactive.
5. Describe how dumps are handled (DMPX to ZZZDUMP).
6. Understand that only the last dayfile message is displayed and describe work arounds.

## REFERENCES

IAF Reference Manual.

## INTERACTIVE - TIME SHARING

## LOGGING IN

```
79/08/23   14.59.16   TM1017
(22)  SVL   SN614  NOS
FAMILY:   SVL
USER:  JFOX
PASSWORD:  NOWAY
APPLICATION: IAF
TERMINAL  13,NAMIAF
RECOVER/CHARGE: CHARGE,MYDEPT,MYPROJECT

/
```

CAN BE ABREVIATED AND ENTERED ON FAMILY LINE-

SVL,JFOX,NOWAY,IAF

IAF IS READY FOR YOU TO ENTER A COMMAND! (BATCH SUBSYSTEM)

NOTICE THAT THE PASSWORD WAS CHANGED (PASSWOR) BETWEEN THE PREVIOUS BATCH RUN AND THIS INTERACTIVE SESSION.

## INTERACTIVE

### GENERAL

- INPUT AND OUTPUT ARE 'CONNECTED' BY DEFAULT. MOST PROGRAMS AND UTILITIES SEND DATA TO OUTPUT AND RECEIVE DATA FROM INPUT.

  ASSIGN,MS,OUTPUT WILL ASSIGN OUTPUT OR ANY FILE TO A DISK DEVICE. THIS IS LIKE THE NOS/BE DISCONT STATEMENT.

  ASSIGN,TT,OUTPUT WILL <u>DISCARD</u> ANYTHING CURRENTLY ON OUTPUT AND ASSIGN IT TO THE TERMINAL. THIS IS LIKE THE NOS/BE CONNECT STATEMENT.

- GENERALLY, MOST OF THE BATCH STATEMENTS WORK IN INTERACTIVE - EVEN THE MAGNETIC TAPE STATEMENTS.

- DAYFILE IS A <u>VERY</u> HANDY STATEMENT AND SHOULD BE LEARNED QUICKLY AND EASILY.

- THERE IS ALWAYS A PROMPT (?) WHEN INPUT IS EXPECTED FROM THE TERMINAL USER.

## INTERACTIVE

### GENERAL - CONTINUED

- ASCII (6/12) IS STANDARD IN BASIC AND XEDIT. FCOPY CONVERTS FROM 6/12 TO 8/12 ASCII.

- A CARRIAGE RETURN (CR) BY ITSELF SIGNALS AN EOF. THIS IS LIKE THE NOS/BE %EOF. IF DATA WAS NOT EXPECTED (READ STATEMENT IN PROGRESS) IAF WILL RESPOND WITH THE CURRENT STATUS (EXECUTE,IDLE,ETC.).

- ENQUIRE, OPT. ANOTHER HANDY UTILITY. IF OPT IS 'F' A STATUS OF ALL FILES IS GIVEN - LIKE FILES. 'J' IS THE JOB CONTROL REGISTER. 'R' LISTS RESOURCES USED. JN (JOB NAME) IS VERY USEFUL. THE LIST GOES ON.

- RENAME, NEWLFN=OLDLFN. CHANGES THE LOCAL FILE NAME. USE CHANGE TO CHANGE A PERMANENT FILE NAME.

- REWIND,*. REWINDS ALL FILES.

- RETURN,*. RETURNS ALL FILES. RETURN,OUTPUT SENDS 'OUTPUT' TO THE BIT BUCKET!

## INTERACTIVE

## SPECIAL CONSIDERATIONS

- CYBER LOADER - MUST GROUP DIRECTIVES INTO 'PACKETS' LIKE ON NOS/BE.

  - NOS/BE USES XEQ (WHAT A MESS) FOR ONE TIME EXECUTION.

  - NOS USES THE ENTER DIRECTIVE FOR ONE TIME EXECUTIONS.

    ```
    ENTER./CMD1,/CMD2./CMD3./.../
    ```

    CMD CAN BE ANY STATEMENT.

  - CCL PROCEDURES ARE EVEN MORE POWERFUL AND VERSATILE!

- LIBRARY - CAUTION!

  TO SPECIFY A GLOBAL LIBRARY SET USE X.LIBRARY,LFN...
  THE IAF 'LIBRARY' STATEMENT BEING CHANGED TO 'LIB' IN A FUTURE SYSTEM.

- DMP - ABNORMAL TERMINATION DMP IS WRITTEN TO ZZZDUMP. USE XEDIT TO PRINT IT.

  - USE DMD. ASSIGN OUTPUT TO MS.
  - BUILD INTO ENTER STATEMENT.
  - USE CCL PROCEDURE

## INTERACTIVE

## SPECIAL CONSIDERATIONS - CONTINUED

- ONLY LAST DAYFILE MESSAGE IS DISPLAYED.

  - USE FTN,A,L=0 ETC.
  - USE DAYFILE

- TYPE AHEAD

  - USE CCL
  - WAIT FOR R6

- DEFAULT CARRIAGE CONTROL IS A BLANK. SPECIAL PROCESSING (CONTROL BYTES) IS REQUIRED BUT VERY POWERFUL FOR NON-DEFAULT CARRIAGE CONTROLS.

# LESSON 11

# TEXT EDITORS

## LESSON PREVIEW

This lesson presents the student with advantages and disadvantages of the various text editors on NOS. It also provides detailed information about XEDIT.

## OBJECTIVES

1. Ready to - "choose one editor; learn it; use it".

2. Compare INTERCOM tasks on XEDIT.

3. Understand 'pointer' editor concepts.

## REFERENCES

XEDIT Reference Manual.

# INTERACTIVE EDITORS

MANY TEXT EDITORS EXIST ON NOS. IT IS FUNNY TO WATCH PEOPLE TRY TO USE ALL OF THEM AT THE SAME TIME.

CHOOSE ONE!

LEARN IT WELL!

USE IT!

THREE OF THE EDITORS INCLUDE -

- NOS LINE EDITOR - GREAT FOR THE OPERATING SYSTEM
- EDIT
- XEDIT

# INTERACTIVE

# EDITORS

MANY TEXT EDITORS EXIST ON NOS. IT IS FUNNY TO WATCH PEOPLE TRY TO USE ALL OF THEM AT THE SAME TIME.

CHOOSE ONE!

LEARN IT WELL!

USE IT!

THREE OF THE EDITORS INCLUDE -

- NOS LINE EDITOR - GREAT FOR THE OPERATING SYSTEM
- EDIT
- XEDIT

## INTERACTIVE

## EDITORS - CONTINUED

I CHOOSE XEDIT. IT CAN DO JUST ABOUT EVERYTHING THE NOS/BE EDITOR CAN DO PLUS SOME VERY POWERFUL NEW COMMANDS (COPY, READ, MODIFY AND INTERNAL PF INTERFACE). I MADE TWO COMPROMISES:

- XEDIT IS A POINTER BASED EDITOR. IT TOOK ME ABOUT 2 WEEKS TO 'FORGET' LINE NUMBERS.

- XEDIT COMMANDS CANNOT BE INTERMIXED WITH OPERATING SYSTEM COMMANDS.

## INTERACTIVE

### EDITORS - CONTINUED

LEARN XEDIT COMMANDS IN PHASES. DON'T GO ONTO SECOND PHASE UNTIL YOU ARE COMFORTABLE WITH FIRST PHASE.

- INSERT
  DELETE
  PRINT
  LOCATE
  CHANGE
  NEXT
  TOP
  BOTTOM
  END
  STOP
  MODIFY

- DEFTAB
  TABS
  LISTAB
  COPY
  READ
  INSERTB
  REPLACE

- QMOD
  RMARGIN
  DEOR
  TRIM
  WEOR
  WEOF
  DEOF

- IGNORE THE 'OBTUSE' COMMANDS!

DIRECTIVES CAN BE PLACED ON THE XEDIT CONTROL STATEMENT. PLACE THE XEDIT CONTROL STATEMENT IN A CCL PROCEDURE TAILORED TO EACH USERS NEEDS (I.E. TABS, BELLS, ETC.).

## XEDIT

POINTER ANT

BEAR

CAT

DOG

ELEPHANT

FOX

GOOSE

### POINTER MOVEMENT

TOP
BOTTOM
NEXT
LOCATE

### MODIFICATION

CHANGE
DELETE
INSERT

### DISPLAY

PRINT

### TERMINATION

END

STOP

## INTERACTIVE

## EDITORS - CONTINUED

SPECIAL CONSIDERATIONS

- USE XEDIT IN BATCH
- USE XEDIT FOR PAGE FUNCTIONS
- DON'T USE LINE NUMBERS! THEY ARE A PAIN!
- XEDIT MAKES A LOCAL FILE AND A PERMANENT FILE AT THE SAME TIME.

# LESSON 12

# REMOTE BATCH

## LESSON PREVIEW

This lesson introduces the student to remote batch processing on NOS.

## OBJECTIVES

1. Describe how to read cards and print lines on a remote batch terminal.

2. Describe how to manipulate queues and files at the remote batch terminal.

3. Describe use of each display.

4. Understand alternatives to the queue manipulation commands of NOS/BE that are not available on NOS.

## REFERENCES

RBF Reference Manual.

## REMOTE BATCH

Remote batch is accomplished by RBF. It has commands similar to INTERCOM's remote batch. The commands of IAF and RBF are processed by separate facilities. INTERCOM merges the commands of both facilities together. RBF commands include -

| | |
|---|---|
| DISPLAY | JOB,IN,PR,PU,DEV,etc. |
| CHANGE | PRI,REP |
| DIVERT | JOB,IN,PR,PU,EX, etc.<br>FAM,USR,HST |
| PURGE | JOB,IN,PR,PU,EX, etc. |
| SKIP | ±n,DFL,END |
| REWIND | |
| RETURN | |

## REMOTE BATCH

## SPECIAL CONSIDERATIONS

- THE RBF BATCH RELATED COMMANDS ARE NOT AVAILABLE IN IAF.

  - VALIDATE USERS FOR IAF AND RBF. THE USER CAN SWITCH BETWEEN IAF AND RBF TO DO THE RBF QUEUE MANIPULATION COMMANDS.

- CAN'T MOVE A PRINT FILE FROM THE PRINT QUEUE TO A LOCAL FILE.

  - STANDARD CODE WILL BE AVAILABLE (1Q80?).

```
QGET
QDROP
QPURGE
```

# LESSON 13

# TELEPROCESSING

## LESSON PREVIEW

This lesson provides the student with knowledge of the NOS/BE Multi User Job (MUJ) facility on NOS. The process is called teleprocessing while the facility is called QTRM.

## OBJECTIVES

1. Describe the teleprocessing facilities available on NOS.

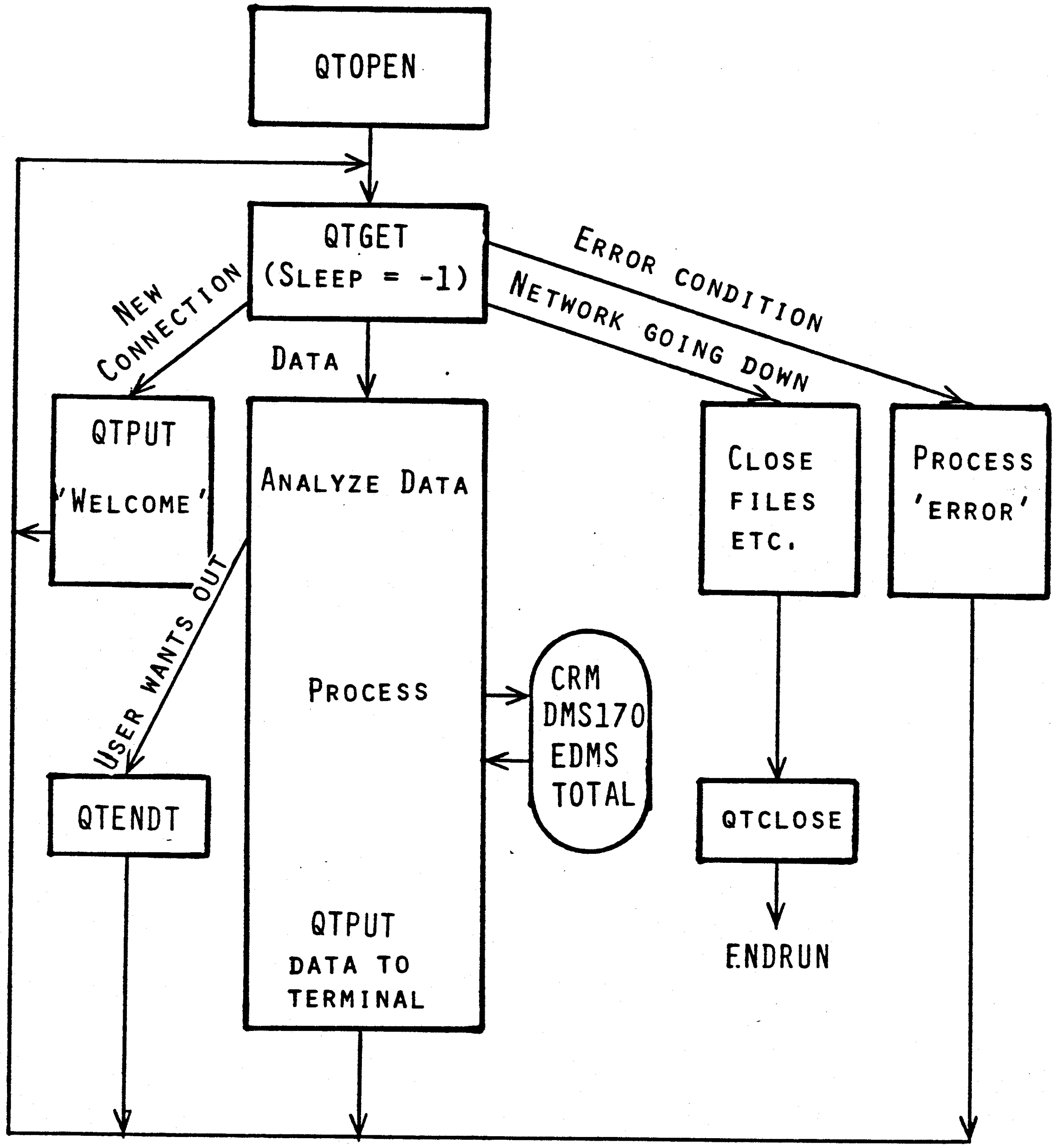
## REFERENCES

None.

## TELEPROCESSING (MUJ)

TELEPROCESSING (ONE PROGRAM COMMUNICATING WITH MANY TERMINALS) IS ACCOMPLISHED BY NAM/AIP/QTRM. SUPPLY THE PROGRAMMER(S) WITH A NAM APPLICATION NAME AND VALIDATION FOR THAT NAME AND LET THEM HAVE THEIR FUN. QTRM WAS DESIGNED FOR COBOL AND FTN PROGRAMMERS. CALLS INCLUDE:

- QTOPEN - IDENTIFIES PROGRAM (APPLICATION) TO NAM.
- QTPUT - SENDS DATA TO A SPECIFIED TERMINAL CONNECTION.
- QTGET - SCANS TERMINALS FOR INPUT FROM THE TERMINAL USERS.
- QTENDT - TERMINATE (NORMAL OR FORCED) A SPECIFIED TERMINAL CONNECTION.
- QTCLOSE - TERMINATES COMMUNICATION BETWEEN THE PROGRAM AND NAM (ALL TERMINALS TO THIS PROGRAM).


QTOPEN
QTGET
(SLEEP = -1)
NEW CONNECTION
DATA
ERROR CONDITION
NETWORK GOING DOWN
QTPUT
'WELCOME'
ANALYZE DATA
PROCESS
QTPUT
DATA TO TERMINAL
USER WANTS OUT
QTENDT
CRM
DMS170
EDMS
TOTAL
CLOSE
FILES
ETC.
QTCLOSE
ENDRUN
PROCESS
'ERROR'

# LESSON 14

# EASING THE TRANSITION

## LESSON PREVIEW

This lesson provides the student with tasks that the user may do now on NOS/BE to ease the transition to NOS.

## OBJECTIVES

1. Identify commands that are common between the two operating systems.
2. Identify tasks suitable for inclusion into CCL procedures.

## REFERENCES

None.

## EASING THE TRANSITION

- USE THE MIGRATION AS AN OPPORTUNITY TO REVIEW CURRENT SYSTEMS, OPERATIONAL PROCEDURES, ETC.

- START NOW - USE COMMANDS THAT ARE COMMON ON BOTH SYSTEMS. I.E. USE ROUTE NOT BATCH OR DISPOSE.

- START NOW - USE CCL PROCEDURES FOR MUNDANE TASKS. CHANGE CONTENT OF PROCEDURE, IF NECESSARY, WHEN YOU CHANGE SYSTEMS.

- DON'T TRY TO USE ALL OF THE UTILITIES. A SMALL NUMBER WILL FILL MOST USERS NEEDS. SOME 'OLD' UTILITIES EXIST FOR HYSTERICAL REASONS (IN ADDITION TO HISTORICAL REASONS).

- LEARN THE UTILITIES A LITTLE EACH WEEK.

- CHOOSE GENERAL PURPOSE COMMANDS OVER SINGLE PURPOSE COMMANDS.

- COMPARE TASKS. DON'T COMPARE FEATURES.

- IDENTIFY CURRENT COMPASS MACRO USE. PLAN YOUR ATTACH. CHECK INTO COMMON COMMON DECKS.

## EASING THE TRANSITION - CONTINUED

- PLAN FOR ZERO LOCAL MODS.
- GET ACCOUNTING READY. SOME JOB COSTS WILL GO UP WHILE OTHERS WILL GO DOWN.
- COMPUTER POLICY WILL CHANGE. INFORM USERS.
- TRADE SOMETHING (NEW FEATURES, NEW HARDWARE,ETC.)
- PEOPLE ARE STILL AFRAID OF THE UNKNOWN.
- THE OPERATING SYSTEM MIGRATION WILL BE EASIER THAN THE CONVERSION FROM SCOPE 3.3 → SCOPE 3.4.
- OPPORTUNITY TO REVIEW OVERALL SYSTEMS, OPERATIONAL PROCEDURES, ETC.

# NOS/BE to NOS USAGE DIFFERENCES

JOHN S. FOX
CONTROL DATA CORP.
FIELD SUPPORT
SUNNYVALE, CALIF.

This handout is a supplement to the NOS/BE to NOS USAGE DIFFERENCES class being presented. This handout consists of the slides from the verbal presentation, general overview of the pertinent aspects of NOS differences, detailed 'how to do it' instructions and miscellaneous attachments.

The purpose of this class is to provide usage information for the user who desires to convert from NOS/BE to NOS. The target audience is the computer programmer/user. Systems analyst and operator training are provided in the appropriate analysis and operator classes. The intent of this class is not to provide 'another' list of feature differences but rather to explain some of the architecture of NOS and then how to do NOS/BE tasks on NOS. There is also an important section on 'Easing the Transition'. Many of the conversion headaches can be minimized by informing the current NOS/BE community of commands that are common on both systems, work-arounds that can be implemented on NOS/BE now, and which NOS commands should be learned or ignored.

## HISTORY

Both NOS and NOS/BE have a common ancestor - the Chippewa Operating System (COS). COS was developed by Control Data Corporation at its Chippewa Laboratory during the early 1960s. COS was designed as a test-bed operating system to aid in the checkout and demonstration of the initial CDC 6600 Computer Systems.

Subsequently, development began on that version of COS known as Chippewa 1.1. This development resulted in the first version of the system now known as SCOPE 3. Shortly thereafter, a remote batch communications package (EXPORT/IMPORT) was added, followed by an interactive communications package (INTERCOM). By February, 1976, SCOPE 3 had evolved into NOS/BE, which provided support for CYBER 170 features.

During the mid-60s, a small, independent group within Control Data was given responsibility for continuing the development of the Chippewa Operating System. Its objective was to enhance the Chippewa system into an efficient, high-performance product. As the system evolved over the years, its name changed from Chippewa to MACE, from MACE to KRONOS, and with the advent of the CYBER 170, to NOS (Network Operating System).

In early 1970, KRONOS Version 1.0 was announced as a standard product. At the same time, Control Data initiated a significant development effort to upgrade the system leading to the release of KRONOS Version 2.0 in August 1971. Further major enhancements resulted in KRONOS Version 2.1 which was released in July, 1973. NOS 1.0, containing CYBER 170 feature code, was released in May, 1975.

While the emphasis and major use of KRONOS and NOS was in the time-sharing environment, these systems also supported concurrent multi-mode operations for remote batch, local batch, deferred batch and transaction processing.

In 1972 (SCOPE 3.4) and 1973 (KRONOS 2.1), the compiler and data management products were made common on both operating systems.

Slide number V1-A is a graphic representation of the history of the operating systems.

## SYSTEM ACCESS

The FAMILY is a very important concept on the NOS system. A family consists of the following -

a) Mass storage devices - up to 63 multi-spindle devices.
b) Validation file - the validation file consists of a list of authorized users, their current access password, their resource limits and authorized charge numbers.
c) Permanent file catalogs - Each user has a permanent file catalog. The catalog is assigned a device within the family by the operations staff.
d) Permanent files - The actual direct and indirect permanent files reside on the devices within the family.

Families provide a technique to group communities of users together and keep them separated from other groups running on the same system. Files are not shared between families. Families provide a backup system for multiple mainframes. If one mainframe is unavailable (maintainance, etc.) then the devices of that mainframe may be accessed by another mainframe.

A second classification of mass storage device exists on NOS called an auxilary device (one to many spindles). The auxilary device supplements the mass storage provided by family devices. Each auxilary device is a self-contained permanent file device. An auxilary device may be either public or private. Anyone permitted to use auxilary devices who supplies the appropriate pack name can create, replace and access files on a public device. Only one user (of one family), the owner, can create and replace files on a private auxilary device, but others may access those files as permitted by the owner. Auxilary devices can be shared between families of users.

## USER STATEMENT

The USER control statement accomplishes many functions on NOS. It must be placed immediately after the job statement if in batch. The operating system validates the user by examining the validation file of the specified family. Some attributes of the USER control statement include -

A) It is common and required for access from all sources including batch, interactive, remote batch, teleprocessing and transaction.

B) The USER statement consists of a password. The password is controlled by the user. If the user feels that his user-name is being or might be compromised, the user can change the password at any time as often as desired. This moves unauthorized system usage responsibility from operations to the user where it rightfully belongs.

C) The USER statement controls access to the user's permanent files. In NOS/BE terminology it provides an implicit ID parameter on all permanent file commands. It is also a super password. Combined with the M parameter on permanent file commands it provides the facilities of the NOS/BE RD, MD, EX, CN, MR, RW and XR parameters.

D) It establishes maximum resource limits. A user can be limited to any resource desired by the operations staff - CM, CP, files, tapes, disks, etc.

## PERMANENT FILES

The permanent files on both NOS and NOS/BE appear the same to the source program running on both systems. The philosophy, architecture and permanent file manipulation commands, however, are different and require an understanding of the NOS philosophy. In particular, NOS permanent files have the following characteristics.

- Files belong to specific users. NOS is set up to keep users away from each other. If you want to share files you must explicitly make them 'public' or explicitly PERMIT specific users to access the file(s).

- There are two types of permanent files - Direct and Indirect. When an indirect file is assigned to a job you get a copy in scratch space (a local file). When a direct file is assigned to a job you get a real file.

Direct files use slightly more physical disk space than indirect files. The 'allocatable' unit of direct disk space is the track while indirect disk space is the PRU. A PRU is 64 CM words while a track varies depending on the device type. Typical track size values range from 50 to 650 PRUs. Indirect file space is maintained by NOS in units of PRUs. Direct file space is maintained in units of tracks.

Direct files are ideal for large data files while indirect files are ideal for small data files. The definition of small is a subject of continual debate. At our site, it is one track.

- All permanent files may be one of the following three <u>categories</u> -

  PUBLIC - Any user who knows the user-name, permanent file name and password, if specified, can access the file. The access <u>mode</u> (read, write, etc.) can be restricted for alternate users.

  SEMI-PRIVATE - This is a strange name! It is the same as a PUBLIC file except that a record (log) is kept of which users accessed and when the file was accessed.

  PRIVATE - This is the most secure file category. Each alternate users (if any) must be explicitly PERMITted to access the file for a specific maximum access mode. You may freely access the file.

- All permanent files may be assigned an <u>access mode</u>. The modes are listed below in <u>order</u> from lowest to highest.

NULL Alternate users can not access the file under any conditions.

EXECUTE Alternate users can only execute the file.

READ Alternate users can execute the file, read its contents, and save a copy of it.

APPEND Alternate users can only take information from another local file and append it to the permanent file.

MODIFY Alternate users can replace the original contents of the file with modified entries and also append information to the end of the file (only applies to direct access files).

READMD Alternate users can read the contents of the file while someone else modifies the file (only applies to direct access files).

READAP Alternate users can read the contents of the file while someone else is appending information to the end of the file (only applies to direct access files).

WRITE Alternate users can write information into any part of the file, read its contents, save a copy of it, execute it, and purge it from the system.

Since there are two types of permanent files (direct and indirect), there are commands that apply only to direct files, only to indirect files and others that apply to both types. The commands include -

DEFINE Establish a direct permanent file.

ATTACH Access a direct permanent file.

SAVE Establish an indirect file.

GET Access an indirect file.

REPLACE Replace an existing indirect file (implicit PURGE/SAVE).

PURGE Remove the specified permanent file name and the associated disk space from the permanent file catalog. Direct or indirect files.

| | |
|---|---|
| APPEND | Add new information to the end of an existing <u>indirect</u> file. |
| CHANGE | Change attributes of a direct or indirect file. (Mode, type, password or permanent file name). |
| PERMIT | Allow specific alternate users to access the specified permanent file for a specific maximum access code. |
| CATLIST | List the contents of the users permanent file catalog. (Permanent file name, creation date/time, last access, file size, etc.). |

The NOS Reference Manual details all of the parameters for all of the above commands. The M (mode) and CT (category) need some clarification. The DEFINE and ATTACH commands will be used as examples. The indirect commands (GET,SAVE,etc.) function in a similar manner.

```
DEFINE,[1fn=]pfn[/CT=category,M=mode,PW=password].

ATTACH,[1fn=]pfn[/M=mode,PW=password,UN=user-name].
```

Notice that the mode parameter is used on both commands. It has a totally different function on the two commands. On a file creation command (DEFINE, SAVE) the M parameter does the following for each file category -

A) PUBLIC - Defines the <u>maximum</u> access <u>mode</u> for alternate users. If an explicit user(s) is PERMITted to access the file, the PERMIT mode value overrides the DEFINE mode. This allows the file owner to restrict access to most users (M=R) yet allows certain users fuller access to the file (M=W etc.).

B) SEMI-PRIVATE - Same as PUBLIC files.

C) PRIVATE - Does nothing. Can be left off. If an alternate user desires to access the file then explicit PERMIT must be executed to identify the user-name and maximum access mode for the user.

The *creator* of the file can always access the file for any access mode. The mode parameter has no impact. On a file *access* command (ATTACH) the M parameter establishes the access mode for *this* access. The access mode requested by alternate users must be less than or equal to the access mode authorized by the creator of the file. If an alternate user exceeds the authorized access mode, the following message is displayed -

```
pfn NOT FOUND
```

## INDIRECT ACCESS VS DIRECT ACCESS FILES

### Indirect Access Files

1. To create an indirect access file, the user first enters information into a local file. Then, the user issues a SAVE command to establish a permanent version of the local file on mass storage.

2. When a user subsequently accesses an indirect access file to use it as a local file, the user obtains a copy of the permanent file. The user enters a GET command to access indirect access files as local files.

3. When a user modifies an indirect access file, he or she modifies a working copy of the original permanent file. The permanently cataloged version of an indirect file is only changed if the user issues a REPLACE command (the modified working copy replaces the original file in the user's catalog).

4. Indirect access files apply to programs or smaller data files.

### Direct Access Files

1. To create a direct access file, the user first issues a DEFINE command. Then, data is written directly into the file (on mass storage). Typically, direct access files are generated from executing programs.

2. When a user subsequently accesses a direct access file to use it as a local file, the user obtains the permanent file itself (not a copy of the file). The user enters the ATTACH command to access a direct access file as a local file.

3. When a user modifies a direct access file, he or she actually modifies the permanent file itself (information is written directly into the only version of the file retained by NOS).

4. Direct access files generally store large data bases.

5. Indirect access files do not operate under write interlock because users only manipulate a working copy of the permaent file.

5. Direct access files possess a write interlock. This means that a file cannot be accessed by one user if another user is already accessing it in write mode. Similarly if other users are already accessing the file (e.g., in read mode), then a user who wants to write into the file cannot do so until the other users finish their activities.

6. The user can access an indirect access file as a local file.

6. The user can only attach a direct access file to a local file.

### MAGNETIC TAPES

The difference between NOS and NOS/BE in the area of magnetic tapes is very small for the user. The primary area of difference is the reservation technique. It has been my experience that magnetic tapes are very mysterious to many people. I have included my 'job control' magnetic tape usage notes at the end of this handout.

### SOURCE STORAGE, USAGE and MAINTENANCE

NOS has three major facilities to maintain source programs in batch and interactive. The facilities are UPDATE, MODIFY and XEDIT.

UPDATE is the same on both systems. You probably know how to use UPDATE. There is no reason to stop using it. NOS does not have the NOS/BE 'random bit' file copy problem. Among other things, this means that an UPDATE PL can be COPYEIed from disk to disk and be usable after the copy.

A PL can also be copied to tape as backup and copied back to disk without using the UPDATE A and B options. Note, however, that the PL is not usable by UPDATE while it is on the backup tape - the UPDATE B option is required to convert from random to sequential mode.

UPDATE is used to maintain the common products (FTN, COBOL, LOADER, etc.) and NAM. New development is also proceeding on UPDATE e.g. 8 bit support. Its life is assured!

MODIFY has features similar to those in UPDATE. A MODIFY OPL must be on disk in order to be accessed by MODIFY. It will not work on tape. The operating system software is maintained by MODIFY. Operating systems analysts will have to learn MODIFY.

XEDIT is a pointer based text editor that works both in batch and interactive. XEDIT will be covered in more detail after the interactive section of this handout.

## OBJECT STORAGE, USAGE and MAINTENANCE

- STORAGE

  Object modules can be stored on sequential files and random library files usable by the CYBER Loader. There are no restrictions on sequential files - all module types can be stored (relocatable, absolutes, capsules, texts, etc.). Random library files are currently restricted to relocatables and capsules.

- USAGE

  Sequential files of relocatables, absolutes, etc. perform the same way on both systems.

  Capsules are stored in random libraries on both systems. They are accessed in the same way on both systems. Typically, the user program via FDL specifies which library files are to be searched by the loader.

Absolutes currently cannot be stored in and accessed from NOS user libraries. There are at least three alternatives -

a) If the purpose of storing absolutes in a NOS/BE user library was to save disk accesses then FOL (Fast Overlay Loader) should be considered. Access to absolutes in NOS/BE libraries requires 3 disk accesses while FOL only requires one disk access per absolute overlay from a sequential file.

   FOL has been available on both NOS and NOS/BE since PSR Level 472. The only change required in most cases is to place an OV=n parameter on the main overlay OVERLAY directive.

b) A GTR statement can be inserted in the place where the global LIBRARY statement would be on NOS/BE. GTR (get records) is a very powerful copy utility. With it one would be able to copy certain absolutes onto a temporary 'work' file from a large sequential file.

c) Code is available (via the underground) for LIBGEN so that absolutes can be placed in user libraries. The code hopefully will be standard when you convert to NOS.

CCL Procedures (the 'only' kind) currently cannot be stored in NOS user libraries. The simplist and most powerful solution (even better than NOS/BE libraries) is to place your procedures into an indirect permanent file - lets call it PROCS. When the procedure is desired, in interactive enter the statement:

```
-proc,PROCS,p1,p2,,,.
```

NOS (CCL) will do an automatic GET of the procedure file PROCS and execute the appropriate procedure. Batch requires a BEGIN,proc,PROCS,p1,p2,,,.

*Relocatable main* programs can be stored in NOS user libraries. However,only the CYBER loader knows how to read global libraries. NOS does not search global libraries in response to a *name call load*. NOS does not invoke the CYBER loader unless explicitly called. The CYBER loader can be explicitly invoked with the following sequence -

```
LDSET,LIB=libname.
Myprog.
```

CPUMTR (1AJ) code is available via the underground.

- Maintenance

Object modules can be maintained on sequential files with the following utilities -

a) COPYL - Common on both systems.

b) COPYLM - Common on both systems.

c) ITEMIZE - Common on both systems.

d) GTR - Very handy extraction utility. This utility is definitely on the list of 'must-be-learned' utilities.

e) LIBEDIT - Don't get fooled by the name. This is a fancy, directive driven COPYL. It generates a *new* 'clean' file.

Object modules (relocatables and capsules) can be 'maintained' on user library files with the LIBGEN utility. LIBGEN's *only* task in life is to copy a 'clean' sequential file of object modules to user library format. LIBGEN cannot read an existing user library file. There are two CCL procedures in the slide portion of this handout. The procedure uses GTR, LIBEDIT and LIBGEN to form a one-step user library maintenance utility.

## INTERACTIVE

NAM provides the basic interface to interactive, remote batch, transaction and teleprocessing on NOS. Refer to the slides for a pictorial representation of the basic interaction. The terminal user is first validated. This is accomplished by entering the desired family and the validated user-name and password. This interaction is handled by NAM. After the user has been validated, NAM will request a NAM application name to be entered. NAM applications include the following -

IAF - Interactive. This is the access to the compilers, permanent files, editors, etc., in time sharing mode (one program - one terminal).

RBF - Remote Batch. Refer to the RBF reference manual for full details on operation - card readers, line printers, card punches, plotters, etc.

TAF - Transaction. Refer to the TAF reference manual for full details.

xxxxxx - Teleprocessing. The programmer may code his own NAM 'application'. Typical uses would be to code a NOS/BE-like MUJ (Multi User Job). COBOL and FTN programmers should refer to appendix G (QTRM) of the NAM reference manual.

Refer to the slides in this handout for more general information concerning interactive, remote batch and teleprocessing.

## TEXT EDITORS

Many text editors exist on NOS. It is a very sad/funny situation to see people trying to use all of the text editors at the same time. My suggestion is to very carefully choose one of the editors. Learn it well and then use it. Three of the many text editors include -

a) NOS line editor - This editor is great for the operating system. It does not fill the needs or requirements of the sophisticated user. Entire lines can be inserted or deleted.

b) EDIT -

c) XEDIT - A fairly powerful pointer based editor with most of the features that the NOS/BE user is accustomed to.

There is an editor comparison list at the end of this section for benefit. I choose XEDIT. It can do just about everything the NOS/BE EDITOR can do plus some very powerful new commands (COPY, READ, MODIFY and terminal P.F. interface). I had to make two compromises -

a) XEDIT is a pointer based editor. It took me about two weeks to 'forget' line numbers. Currently, I feel that the NOS/BE Editor line numbers get in the way. Don't get tempted to use them!

b) XEDIT commands cannot be intermixed with operating system commands. I.e. get into XEDIT, edit your source and then get out of XEDIT.

I suggest that one learn the XEDIT commands in phases. Use the simple forms of the commands. Don't go on to subsequent phases until the prior phases have been mastered. Suggested phases -

1) INSERT
DELETE
CHANGE
MODIFY
LOCATE
NEXT
TOP
BOTTOM
PRINT
END
STOP

2) COPY
READ
REPLACE
INSERTB
DEFTAB
TABS
LISTAB

3) WEOR
WEOF
DEOR
DEOF
TRIM
RMARGIN
QMOD

Directives can be placed on the XEDIT control statement. Consider inserting your XEDIT control statement into a CCL procedure tailored to your needs - tabs, bells, margins, etc.

XEDIT is very versatile. It can be used in batch. It can be used as the NOS/BE PAGE command. The END command can make both a local and a permanent file at the same time. There is no need to do NOS/BE 'saves' or 'CATALOG's.

COMPARATIVE EDITOR FEATURES

| Feature | NOS Line Editor | Edit | XEDIT |
|---|---|---|---|
| What can be modified. | Edits only entire lines. | Edits lines, portions of lines (strings), and individual characters. | Edits lines, portions of lines (strings), and individual characters. |
| How much can be modified. | Can only modify one line at a time. | Can modify numerous lines with one command | Can modify numerous lines with one command. |
| Number of commands allowed in a single line. | None | One | 39 |
| Easy command syntax. | Does not provide commands. | Has more delimiters than XEDIT. | Simpler command syntax than EDIT because of fewer mandatory delimiters. |
| Verification of user editing entries. | No | No | Automatically verifies how certain commands alter the file. |
| Internal interrupt processing. | Not applicable. | If user interrupts XEDIT, processing stays under control of the editor and the user's previous editing is retained. | If user interrupts XEDIT, processing stays under control of the editor and the user's previous editing is retained. |

| Feature | NOS Line Editor | Edit | XEDIT |
|---|---|---|---|
| Movement through the user's file. | All searches through the file are based solely on line numbers. | Searches through the user's file are based on string criteria. Searching solely on the basis of line numbers is risky since EDIT will treat that number as a string not necessarily as a line number. | Searches through the user's file can unambiguously be based solely on line numbers as well as string criteria. |
| Availability of NOS permanent file commands. | All NOS commands are immediately available because the line editor operates as a part of the normal NOS control language. | User must terminate EDIT processing, issue an appropriate file command under NOS, and then recall EDIT. | NOS file commands are available within the XEDIT command set. Accordingly, file operations can be performed without terminating XEDIT execution. |
| Easy string and character modification. | Individual characters or strings cannot be modified; only entire lines can be edited. | Users employ various string manipulation commands. | Users can employ the MODIFY command which requires less input than comparable EDIT commands. |

## COMPARITIVE EDITOR FUNCTIONS

| NOS Line Editor | EDIT Text Editor | XEDIT Extended Text Editor |
|---|---|---|
| Usable without having to call | Callable by entering:<br>EDIT | Callable by entering:<br>XEDIT |
| Is not designed to process multiple-record files. | Can not edit multiple-record files. | Handles multiple-record files and multiple-file files. |
| No command set. User can only add, delete, and replace lines. | Full set of commands for editing lines and strings. | Full set of commands for editing lines and strings. |
| Only applicable to the primary file. | Applicable to primary and secondary files. | Applicable to primary and secondary files. |
| Only edits lines that are preceeded by line numbers. | Can edit lines with or without line numbers. | Can edit lines with or without line numbers. |
| Not usable with direct access files. | Usable with direct access files but editing changes are written onto the direct access file itself. | Usable with direct access files but editing changes are written onto a copy of the direct access file. |

## EASING THE TRANSITION

The transition from NOS/BE to NOS can be eased by planning ahead. I have listed some suggestions to ease the transition. Some are aimed at users while others are aimed at computer management.

- Use commands that are common on both NOS/BE and NOS. An example is to use ROUTE instead of BATCH or DISPOSE. This process should start today.

- Use the migration as an opportunity to review current systems of programs, operational procedures, etc. Many times, a review of a complex system can expose inefficiences that can easily be improved upon.

- Use CCL procedures for mundane control statement tasks. Implement the procedures on NOS/BE. The conversion process will consist of changing some statements in the body of the procedure in one common place.

- Choose general purpose commands instead of single purpose commands. An example is XEDIT. It can do everything that EDIT and the NOS line editor can do. Another example is the RETURN command. 'RETURN,*' does the same as CLEAR.

- Don't try to use all of the utilities in NOS. A small number will generally fill most users needs. Some 'old' utilities on NOS (like some on NOS/BE) exist for hysterical in addition to historical reasons.

- Learn the utilities a little each day.

- Compare tasks. Don't compare features or parameters. Many times, when the task is viewed objectively from afar, the task can be accomplished with fewer clearer and more efficient statements.

- All sites that have converted said that the operating system change was an order of magnitude easier than the SCOPE 3.3 to SCOPE 3.4 product set change.

- Accounting will change. Get the accounting programs ready as soon as possible. Some jobs will cost more to run while others will cost less. The low cost jobs on NOS/BE were probably not paying their fair share for all systems resources.

- Computer policy will change. Inform users well in advance.

- If possible, trade something. The users will have to change their job control, procedures, etc. New features or new hardware always gives users an ego boost.

- Plan for zero local operating system mods.

- Identify current COMPASS macro use. Plan your attack. The COMMON COMMON DECKS in COMPASS are a valuable source of code to do mundane tasks.

- People still don't like to change. People are still afraid of the unknown - inform them!

This attachment consists of a table of the NOS commands grouped by logical function. There is a brief description of the command and a manual to refer to for more detailed information. New users of NOS have found this to be very useful during the first few months of exposure to NOS.

```
NOS     3ONTROL STATEMENT SUMMARY BY CLASSIFICATION  79/09/07   PAGE 00001

GENERAL CONTROL STATEMENTS        01
PROGRAM  REF DESCRIPTION

BEGIN     A  INITIATE EXECUTION OF A PROCEDURE FILE
COMMENT   A  INSERT A COMMENT INTO THE J3B 3AYFILE
CTIME     A  DISPLAYS ACCUMULATED CPU TIME READING FOR THE JOB IN THE DAYFILE
DISPLAY   A  EVALUATE AN EXPRESSION AND PUT RESULTS IN THE JOB DAYFILE
ELSE      A  INITIATE OR TERMINATE CCL SKIPPING
ENDIF     A  TERMINATE CCL SKIPPING
ENDW      A  TERMINATE CCL REPETITIVE SEQUENCE
EXIT      A  ESTABLISH AN EXIT PATH FOR JOB IN CASE OF AN ERROR
IFE       A  CAUSES CONTROL STATEMENTS TO BE SKIPPED IF THE CCL CONDITION IS FALSE
JOBCARD   A  JOB IDENTIFICATION AND RESOURCE REQUIREMENTS
MFL       A  SETS THE MAXIMUM FIELD LENGTH FOR A JOB STEP
NOEXIT    A  INHIBITS THE EFFECT OF ANY -EXIT- STATEMENTS
NORERUN   A  ENSURES THAT THE JOB IS NOT RERUN IF A FAILURE OCCURRS
OFFSW     A  TURN OFF ONE OF THE 6 SENSE SWITCHES
ONEXIT    A  REVERSES THE EFFECT OF THE -NOEXIT- STATEMENT (EXIT IS OPERATIONAL)
ONSW      A  TURN ON ONE OF THE 6 SENSE SWITCHES
RENAME    A  CHANGE THE NAME OF A LOCAL FILE
RERUN     A  REVERSES THE EFFECT OF THE -NORERUN- STATEMENT
RESOURC   A  RESERVE A MAXIMUM NUMBER OF PERIPHERAL DEVICES (DISK, TAPE ETC.)
REVERT    A  RETURN FROM A CCL PROCEDURE
RFL       A  REQUEST A DIFFERENT CENTRAL MEMORY FIELD LENGTH
RTIME     A  DISPLAY CURRENT REAL-TIME CLOCK READING IN THE JOB DAYFILE
SET       A  ALLOWS THE USER TO ASSOCIATE A VALUE WITH CERTAIN CCL REGISTERS
SETTL     A  SPECIFY A NEW CPU TIME LIMIT FOR SUBSEQUENT JOB STEPS
SKIP      A  CAUSES UNCONDITIONAL STATEMENT SKIPPING - TERMINATED BY AN -ENDIF-
STIME     A  DISPLAY ACCUMULATED SRU VALUE FOR THE JOB IN THE DAYFILE
SWITCH    A  TURN ON ONE OF THE 6 SENSE SWITCHES (REPLACED BY ONSW)
WHILE     A  BEGIN CCL REPETITIVE SEQUENCE

JOB AND RESOURCE AUTHORIZATION  05
PROGRAM  REF DESCRIPTION

ACCOUNT   A  FOR COMPATIBILITY ONLY (REPLACED BY -USER-)
CHARGE    A  SPECIFIES WHICH PROJECT IS TO BE CHARGED FOR THIS COMPUTER RUN
LIMITS    A  LISTS CURRENT VALIDATION INFORMATION FOR CURRENT USER
MODVAL       DEFINES RESOURCE LIMITS FOR EACH INDIVIDUAL USER
PASSWOR   A  ALLOWS A USER TO CHANGE HER SYSTEM ACCESS PASSWORD
USER      A  IDENTIFIES USER TO SYSTEM WHICH THEN ALLOCATES AUTHORIZED RESOURCES

FILE ASSIGNMENT             10
PROGRAM  REF DESCRIPTION

ASSIGN    A  ASSOCIATE A SPECIFIC PHYSICAL DEVICE WITH A LOCAL FILE
CLEAR     A  RELEASE ALL CURRENT WORKING FILES
COMMON    A  CREATE OR ACCESS A LIBRARY-TYPE FILE
EVICT     A  DISCARD DATA BUT RETAIN LOCAL FILE NAME AND RESOURCE RESERVATION
LDI       A  COPY LOCAL FILE TO INPUT QUEUE WITH ID (REPLACED BY ROUTE)
LOCK      A  ENABLE USER TO PREVENT WRITING ON A FILE
OUT       A  RELEASE OUTPUT, PUNCH(B) AND P8 LOCAL FILES TO RESPECTIVE QUEUES (SEE ROUTE)
REQUEST   A  REQUEST ASSIGNMENT OF A DEVICE ORDEVICE ATTRIBUTE TO A LOCAL FILE
SETID     A  SPECIFY AN -ID- FOR LOCAL FILE TO BE SENT TO QUEUE
SUBMIT    A  REFORMAT AND SEND LOCAL FILE TO INPUT QUEUE
```

```
NOS    CONTROL STATEMENT SUMMARY BY CLASSIFICATION  79/09/07   PAGE 00002


FILE ASSIGNMENT              10
PROGRAM  REF DESCRIPTION

UNLOCK    A  RESCINDS THE EFFECTS OF THE -LOCK- COMMAND


FILE COPY UTILITIES          12
PROGRAM  REF DESCRIPTION

COPY      A  COPY RECORDS AND/OR FILES - SEE REFERENCE MANUAL
COPYBF    A  COPY BINARY FILE(S) IN ODD PARITY
COPYBR    A  COPY BINARY SYSTEM RECORD(S) IN ODD PARITY
COPYCF    A  COPY CODED SYSTEM FILES IN EVEN PARITY
COPYCR    A  COPY CODED SYSTEM RECORDS IN EVEN PARITY
COPYEI    A  COPYFILE UNTIL DOUBLE END-OF-FILE OR EOI IS ENCOUNTERED
COPYSBF   A  COPY FILE WITH EACH UNIT RECORD SHIFTED RIGHT ONE CHARACTER
COPYX     A  COPY FILES WITH CONDITIONS APPLIED - SEE REFERENCE MANUAL
COPY8P    H  COPIES IBM PRINT FILES TO SYSTEM COMPATIBLE PRINT FILES
PACK      A  COPY A FILE AND REMOVE SYSTEM EOR-S AND EOF-S
TCOPY     A  COPY E, B, X OR SI-CODED TAPES TO DISK OR I-TAPE OR SI-BINARY TAPE
VERIFY    A  COMPARE ONE OR MORE CONSECUTIVE SYSTEM LOGICAL RECORDS
WRITEF    A  WRITE A SYSTEM END-OF-FILE ON SPECIFIED FILE
WRITER    A  WRITE A SYSTEM END-OF-RECORD ON SPECIFIED FILE


FILE POSITIONING UTILITIES   14
PROGRAM  REF DESCRIPTION

BKSP      A  BYPASS SYSTEM LOGICAL RECORDS IN THE REVERSE DIRECTION
REWIND    A  REWIND SPECIFIED FILE(S)
SKIPEI    A  BYPASS UNTIL THE END-OF-INFORMATION IS ENCOUNTERED ON THE FILE
SKIPF     A  BYPASS SYSTEM LOGICAL FILES IN A FORWARD DIRECTION
SKIPFB    A  BYPASS SYSTEM LOGICAL FILES IN A REVERSE DIRECTION
SKIPR     A  BYPASS SYSTEM LOGICAL RECORDS IN A FORWARD DIRECTION


FILE DISPOSITION UTILITIES   16
PROGRAM  REF DESCRIPTION

DISPOSE   A  RELEASES FILE BEFORE END OF JOB (REPLACED BY ROUTE)
RETURN    A  PERFORM A CLOSE/RETURN ON A FILE (RELEASE DATA AND RESOURCE)
ROUTE     A  SEND A LOCAL FILE TO A QUEUE (INPUT, OUTPUT, PUNCH ETC.)
UNLOAD    A  PERFORM A CLOSE/UNLOAD ON FILE (RELEASE DATA BUT NOT RESOURCE)


PROGRAM MAINTENANCE          18
PROGRAM  REF DESCRIPTION

COPYCL    L  COBOL SOURCE LIBRARY MAINTENANCE UTILITY
COPYL     R  REPLACE NAMED RELOCATABLE PROGRAMS ON FILE
COPYLM    R  REPLACE MULTIPLE OCCURRENCES OF NAMED RELOCATABLE BINARY PROGRAMS
LIBEDIT   A  REPLACE NAMED RELOCATABLE PROGRAMS ON FILE
LIBGEN    A  ALLOW USER TO GENERATE A USER LIBRARY FILE (RELOCATABLE BINARY ONLY)
MODIFY    W  SOURCE PROGRAM (DECK) MAINTENANCE UTILITY FOR NOS OPERATING SYSTEM
OPLEDIT   A  REMOVE DECKS AND IDENTIFIERS FROM A -MODIFY- SOURCE PROGRAM LIBRARY
SYSEDIT   X  MAKE CHANGES TO THE OPERATING SYSTEM MODULES
UPDATE    F  SOURCE PROGRAM (DECK) MAINTENANCE UTILITY
```

```
NOS    CONTROL STATEMENT SUMMARY BY CLASSIFICATION  79/09/07   PAGE 00003

PERMANENT FILE                    20
PROGRAM  REF DESCRIPTION

APPEND    A  ADD MORE INFORMATION TO AN EXISTING INDIRECT ACCESS FILE
ATTACH    A  ASSIGN A DIRECT ACCESS PERMANENT FILE TO A JOB OR TERMINAL
CATLIST   A  LIST CURRENT STATUS OF SELECTED PERMANENT FILES IN OPERATING SYSTEM
CHANGE    A  CHANGE (RENAME) SPECIFIED PARAMETERS OF A PERMANENT FILE(S)
DEFINE    A  -DEFINE- A DIRECT ACCESS PERMANENT FILE
GET       A  FETCH A COPY OF A INDIRECT ACCESS PERMANENT FILE AS A LOCAL FILE
PACKNAM   A  ASSOCIATE A SPECIFIC DISK PACK(S) WITH A JOB
PERMIT    A  ALLOW A SPECIFIC USER TO USE A SPECIFIC PRIVATE PERMANENT FILE
PURGALL   A  REMOVE ALL PERMANENT FILES IN THE USER-S CATALOG
PURGE     A  REMOVE ONE PERMANENT FILE FROM THE USER-S CATALOG
REPLACE   A  PLACE A COPY OF A LOCAL FILE IN THE PF SYSTEM AS AN INDIRECT ACCESS FILE
SAVE      A  ALLOWS A USER TO RETAIN A COPY OF A LOCAL FILE AA AN INDIRECT ACCESS FILE


MAGNETIC TAPE                     25
PROGRAM  REF DESCRIPTION

BLANK     A  WRITE EMPTY -ANSI- LABELS ONTO A MAGNETIC TAPE
LABEL     A  ASSIGN A MAGNETIC TAPE AND READ OR WRITE THE LABEL(S)
LISTLB    A  LIST ALL THE LABELS ON A MULTI-FILE TAPE
STAGE     A  COPY SPECIFIC SECTIONS OF A MAGNETIC TAPE TO DISK
VSN       A  EQUATE A LIST OF VSN(S) TO A LOCAL FILE NAME


CYBER RECORD MANAGER              30
PROGRAM  REF DESCRIPTION

CRMEP     B  POST-PROCESSOR TO PROCESS RECORD MANAGER ERROR FILE
ESTMATE   B  UTILITY FOR CALCULATING INITIAL INDEXED SEQUENTIAL BLOCK SIZES
FILE      B  PROVIDE -FIT- VALUES FOR A PROGRAM
FLBLOK    V  ESTIMATE BLOCK SIZE ON EXTENDED -IS- FILES
FLSTAT    V  LISTS INFORMATION ABOUT EXTENDED AAM FILES
FORM      E  SUPER COPY UTILITY USING CRM
IXGEN     B  DEFINES ALTERNATE KEYS AND CREATES -MIP- FILE
MIPDIS    V  DISASSOCIATES A MIP FILE FROM ITS EXTENDED INDEXED SEQUENTIAL FILE
MIPGEN    V  DEFINES ALTERNATE KEYS AND CREATES -MIP- FILE EXTENDED AAM
SISTAT    B  LISTS STATISTICAL INFORMATION ABOUT AN -IS- FILE


COMPILERS AND ASSEMBLERS          35
PROGRAM  REF DESCRIPTION

ALGOL     M  ALGOL COMPILER
APEX      S  APEX COMPILER
BASIC     N  BASIC COMPILER
COBOL     L  COBOL COMPILER
COBOL5    L  COBOL 5.X COMPILER
COMPASS   J  ASSEMBLER FOR CYBER SYSTEMS
DDL       O  DATA BASE DEFINITION LANGUAGE PROCESSOR
FTN       K  FORTRAN EXTENDED COMPILER
GPSS      U  GPSS COMPILER
PL-I      Y  PL-I COMPILER (ANSI VERSION...)
QU        P  QUERY UPDATE - FILE MANIPULATION AND REPORT WRITER
SORTMRG   D  SORT AND/OR MERGE RECORDS FROM INPUT FILES
```

```
COMPILERS AND ASSEMBLERS         35
PROGRAM  REF DESCRIPTION

SYMPL     Q  SYMPL COMPILER


CYBER LOADER                     40
PROGRAM  REF DESCRIPTION

EXECUTE   C  CAUSES THE COMPLETION OF LOAD AND EXECUTION OF LOADED PROGRAM(S)
LDSET     C  PROVIDES USER CONTROL OF A VARIETY OF LOAD OPERATIONS
LFN       C  CAUSES EXECUTION OF PROGRAM ON NAMED FILE (LFN)
LIBLOAD   C  SPECIFIES LIBRARY (USER OR SYSTEM) TO LOAD PROGRAM(S)
LIBRARY   C  SPECIFIES GLOBAL LIBRARY SET FOR EXTERNAL SATISFACTION
LOAD      C  LOAD RELOCATABLE BINARY MODULES FROM SPECIFIED FILE(S)
MAP       C  SELECT LOAD MAP LIST OPTIONS
NOGO      C  CAUSES COMPLETION OF LOAD OPERATIONS WITHOUT EXECUTION
REDUCE    A  INITIATE DYNAMIC FIELD LENGTH MANAGEMENT
SATISFY   C  SPECIFIES ORDER OF EXTERNAL PROGRAM REFERENCE SATISFACTION
SEGLOAD   C  DIRECTIVE-DRIVEN SEGMENT LOADER
SLOAD     C  PERFORMS PHYSICAL LOAD OF SELECTED MODULES FROM LOCAL FILES


IAF RELATED                      52
PROGRAM  REF DESCRIPTION

ASCII     A  SPECIFIES THAT ALL SUBSEQUENT TERMINAL OPERATIONS ARE TO BE IN ASCII CHAR. SET
CSET      A  SPECIFIES CHARACTER SET FOR SUBSEQUENT TERMINAL OPERATIONS
EDIT      A  CALLS THE -NOS- TEXT EDITOR
ENTER     A  ENABLES USER TO STACK COMMANDS (USEFUL FOR THE CYBER LOADER
NEW       A  CREATES A NEW EMPTY PRIMARY FILE AND RELEASES OLD (IF IT EXISTS)
OLD       A  MAKES A COPY OF A PERMANENT FILE THE PRIMARY FILE
PRIMARY   A  MAKES A LOCAL FILE THE PRIMARY FILE OR CREATES AN EMPTY PRIMARY FILE
RESEQ     A  RESEQUENCE SOURCE FILE OR ADD SEQUENCE NUMBERS TO A SOURCE FILE
SORT      A  SORT A FILE OF TEXT LINES IF SEQUENCE NUMBERS EXIST


MISC. UTILITIES                  60
PROGRAM  REF DESCRIPTION

DAYFILE   A  COPY SPECIFIED PORTIONS OF THE USER-S DAYFILE TO A LOCAL FILE
ENQUIRE   A  LIST SYSTEM STATUS (FILES, USER, RESOURCES, REGISTERS ETC.)
GTR       A  COPY SPECIFIC NAMED RECORDS FROM ONE FILE TO ANOTHER FILE
ITEMIZE   R  LIST INFORMATION ABOUT EACH RECORD IN A FILE AND ITS FILE STRUCTURE
LENGTH    A  DISPLAYS CURRENT LENGTH OF A SPECIFIED FILE IN THE DAYFILE
SUMMARY   A  LIST SYSTEM STATUS (REPLACED BY ENQUIRE)
TDUMP     A  LIST A FILE IN OCTAL AND/OR ALPHANUMERIC FORM(S)
UPMOD     A  CONVERT AN -UPDATE- SOURCE PROGRAM LIBRARY TO -MODIFY- FORMAT
VFYLIB    A  COMPARE THE RELOCATABLE BINARY INFORMATION ON TWO FILES


DEBUGGING AIDS                   65
PROGRAM  REF DESCRIPTION

DEBUG     X  INFORMS LOADER TO INITIATE -CID- ON SUBSEQUENT LOAD SEQUENCES
DMD       A  DUMP PORTIONS OF USER CENTRAL MEMORY (IN DISPLAY CODE AND OCTAL)
DMP       A  DUMP PORTIONS OF USER CENTRAL MEMORY
```

```
DEBUGGING AIDS                 65
PROGRAM  REF DESCRIPTION

DMPECS    A  DUMP PORTION OF USER EXTENDED CENTRAL STORAGE (ECS)
TRAP      C  DEBUGGING TOOL FOR NON-OVERLAYED PROGRAMS


RARE AND/OR STRANGE            90
PROGRAM  REF DESCRIPTION

CKP       A  GENERATE A CHECKPOINT DUMP FOR RECOVERY PURPOSES
MODE      A  DEFINE JOB HALT CONDITIONS
RESTART   A  RESTART A -CKP- CHECKPOINT TAPE
ROLLOUT   A  MOVE AN EXECUTING JOB TO THE ROLLOUT QUEUE
SETCORE   A  SET EACH WORD IN THE FIELD LENGTH TO SPECIFIED VALUE (REPLACED BY LDSET)
USECPU    A  SPECIFY WHICH CPU TO USE IN A MULTI-CPU MACHINE


MANUALS                        99

   A   NOS REFERENCE MANUAL (VOLUME 1 AND 2)
   B   CYBER RECORD MANAGER (CRM) BASIC ACCESS METHODS (BAM) REFERENCE MANUAL
   C   CYBER LOADER REFERENCE MANUAL
   D   SORT/MERGE REFERENCE MANUAL
   E   FORM REFERENCE MANUAL
   F   UPDATE REFERENCE MANUAL
   H   8-BIT REFERENCE MANUAL
   J   COMPASS REFERENCE MANUAL
   K   FORTRAN EXTENDED REFERENCE MANUAL
   L   COBOL REFERENCE MANUAL (VERSION 4.X OR 5.X)
   M   ALGOL REFERENCE MANUAL
   N   BASIC REFERENCE MANUAL
   O   DATA DEFINITION LANGUAGE REFERENCE MANUAL (DDL)
   P   QUERY UPDATE REFERENCE MANUAL
   Q   SYMPL REFERENCE MANUAL
   R   CYBER COMMON UTILITIES REFERENCE MANUAL
   S   APEX REFERENCE MANUAL
   U   GPSS REFERENCE MANUAL
   V   CYBER RECORD MANAGER (CRM) ADVANCED ACCESS METHODS (AAM) REFERENCE MANUAL
   W   MODIFY REFERENCE MANUAL
   X   CYBER INTERACTIVE DEBUG REFERENCE MANUAL
   Y   PL-I REFERENCE MANUAL
```

THIS ATTACHMENT WAS LIFTED FROM MY NOS 'JOB CONTROL' CLASS HANDOUT. IT HAS BEEN MY EXPERIENCE THAT MAGNETIC TAPES ARE VERY MYSTERIOUS TO MANY PEOPLE. THE HANDOUT STARTS WITH THE BASICS AND PROCEEDS UP TO THE USER-OPERATING SYSTEM INTERFACE.

# MAGNETIC TAPES

JOHN S. FOX
CONTROL DATA CORP.
FIELD SUPPORT
SUNNYVALE, CALIF.

The magnetic tape usage notes have been divided into two sections. The first section deals with the magnetic tape medium (reading technique, load point, parity errors, etc.). The second section deals with the operating system interface between programs and tape.

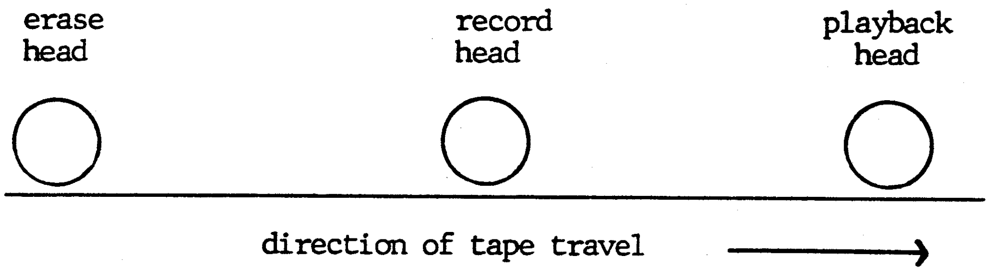
## THE MEDIUM

The magnetic tapes used on our machinery are 2400-foot-long reels of plastic tape, ½ inch wide, with a thin coating of iron oxide on one side. The tape is similar to tapes used for sound recording, but much stronger. It is possible to use shorter lengths of tape. When it is desired to write or read a tape, the tape is mounted on a tape drive (also known as a tape transport).

The tape drive requires a few feet of tape at each end for leaders. Two shiny reflective markers on the uncoated side of the tape are attached to mark the leader areas; they are called the load-point marker and the end-of-tape marker. New tapes are supplied with these tape markers already attached.

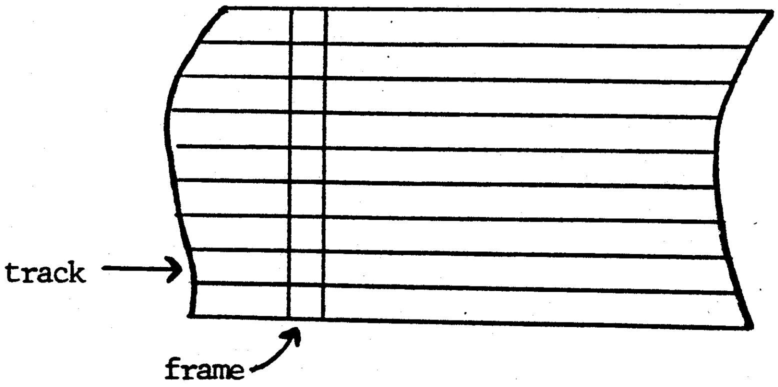
The oxide particles on the tape each have a magnetic field like a tiny magnet. The blank tape, as supplied by the manufacturer, has the magnetic fields of individual particles pointing in random directions, so that the sum of their effects is negligible. Just as a screwdriver or other steel item can be magnetized by stroking with a magnet, the oxide particles can be magnetized by passing them under an electromagnet. When this is done, their electromagnetic fields are all aligned in the same direction, so that their sum is a usable magnetic field.

Most tape drives are built with three electromagnetic heads over (or under) which the tape is passed:



During recording, the tape passes first under the erase head, which erases the tape by magnetizing it completely in one constant direction. (Tape which is randomly magnetized is called blank. Neither erased nor blank tape contains any useful data, and it is hard to tell the difference.) The tape then passes under the record head, which magnetizes the tape alternately in opposite directions, depending on the data. The playback head senses the changes in the direction of magnetization, from which the data are extracted and returned to the computer. When a tape is simply being read, the erase and record heads are disabled.

The area required to represent a single zero or one is very small. On our half-inch-wide tapes, it is common to record nine binary digits side by side. The group of nine side-by-side digits are used to record data; the other is used for a check on the eight and is called a parity bit (to be described later).

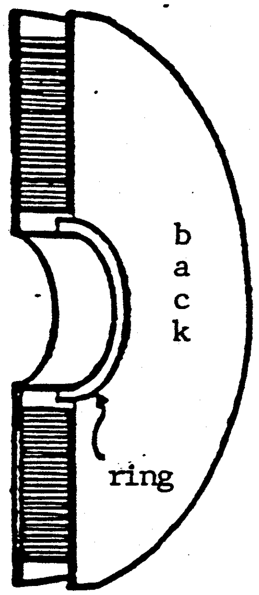


Section of 9 Track Magnetic Tape

When seven digits are recorded side by side, the tape is called a seven-track tape, each digit forming part of a track the length of the reel. Some tape drives use a nine-track format, others use seven-track format. It is very important to know with which you are working, since the tape drives used for one type will not record or play back the other. (The tape itself is the same; it is just the recording format that is different.) At the present time, nine-track tapes are predominant in the industry, so the information in this document refers to nine-track format except where seven-track is specified.

## Write Ring

The diagram to the right shows a cross-section of a reel of tape. On the back side of the reel, around the hole, is a slot. This slot is ordinarily empty. A switch built into the tape drive can sense this. A plastic ring can be inserted into the slot by the computer operator before the tape is mounted. We use this to signify that the tape may be written on. It is called a write ring or write-enable ring.



The operating system requires the operator to insert (or remove) the ring depending on the PO = W or R parameter on the LABEL or REQUEST control statement. If the parameter is omitted then no ring is assumed and enforced by the operating system.

## Density and Speed

Along the length of nine-track tapes, most drives are capable of recording and reading at three different densities: 800, 1600, or 6250 frames per inch (sometimes called bytes per inch, bpi, or characters per inch, cpi).

Tape drives typically operate at a speed of from 100 to 200 inches per second. The number of characters which can be read or written per second at 200 inches per second is a function of the density:

800 frames/inch = 80,000 - 160,000 frames/second
1600 frames/inch = 160,000 - 320,000 frames/second
6250 frames/inch = 625,000 - 1,250,000 frames/second

Since we use 6 bit characters, it is possible (we do) to represent 4 characters in 3 frames on a 9-track tape. The effective transfer rates for 9-track tape drives are:

800 frames/inch = 106,666 characters/second = 9.4 us/character
1600 frames/inch = 213,333 characters/second = 4.7 us/character

The above data transfer rates assume, of course, that you do not have to stop the tape drive for any reason, and that you use every bit of the tape for data (which is actually not possible, as will be explained shortly).

## Record Gaps and Physical Records

What happens if a program which is using a tape is running slower than the tape drive? The drive has to stop now and then to wait for the program. When it stops, it takes some time to brake to a halt, and when it starts, time to get up to 100 or 200 inches per second. The space passing under the heads during that time cannot be used for data, since the speed is changing. This space is composed of approximately 3/8 inch of erased tape. It is called a record gap or sometimes an inter-record gap (abbreviated IRG).

The space between record gaps, which contains the data, is called a physical record, or block. Its length is arbitrary - a physical record could conceivably be as short as one character or as long as the whole tape. Practical considerations, however, exclude both very short and very long physical records.

Very short records result in many record gaps, which are just wasted tape. The following table illustrates this, by comparing the number of 100 character images which can be recorded on a tape using 100, 200, and 3000 characters per physical record (assuming nine-track tape, 2400 usable feet, 4 characters per 3 frames).

| Density | 1 image/rec | 2 images/rec | 30 images/rec | theorectical max |
|---|---|---|---|---|
| 800 | 61,400 | 102,400 | 271,000 | 307,200 |
| 1600 | 68,200 | 122,800 | 485,000 | 614,400 |

At the other extreme, very long records are difficult to handle, because there are so few record gaps that it is hard for the program to keep up with the tape. Also, very long records do not make much difference in the capacity of a tape. For example, notice that 3000-character records provide more than 80% of the theoretical maximum. The "best" blocking, then, is somewhere in the middle, with from 500 to 5000 characters per record.

## Longitudinal Parity Character

In addition to automatically generating a record gap for each physical record, the tape drive automatically generates (after your data and before the record gap) a longitudinal parity character. This is done by taking the sum of all the frames in the physical record. (Actually a Boolean difference is used, not arithmetic sum.) When the tape is read back, the drive automatically generates another character and compares that to the one read at the end of your data; if they match that is OK; if they do not, something has happened to the tape (more about that shortly) and you have a "parity error" to contend with.

## Frame Parity

As mentioned before, the ninth bit of each frame is used for a parity-bit. The usual way of deciding what to write there is to add up the other eight digits, which have values of zero or one, and write a zero if that total is odd, or a one if it is even. The total number of ones in the frame then is always odd (1, 3, 5, 7, or 9). This is called, accordingly, odd parity. The parity bit is generated automatically by the tape drive as it writes each frame. As each frame is read back in, the drive checks to ensure that the number of ones is odd. If it is not, then something has happened to alter your data since it was written, and the drive tells your program that there is a parity error.

Odd-parity recording has the property that each frame of data (including the longitudinal parity character) has at least one digit one. A frame with all nine digits zero is not possible. So that is what we use in the record gap.

Seven-track drives can use another parity scheme called even parity, in which the total number of ones in a frame is 2, 4, or 6. This scheme was very widely used throughout the industry, and there exists a very standard code (corresponding to the Hollerith code for cards) for the numeric and alphabetic characters on even-parity tape.

The disadvantage of even-parity tapes is that if you were allowed to write a 3/4 inch string of characters that were all zero bits, it would look just like a record gap, since the parity bits for such frames would also be zeros. This is undesirable, and the tape drives have been built so that such a character is converted to a different character before being written to tape. With odd parity, you can write any of 64 different characters in one frame, but with even parity you only get the use of 63. There is no logical requirement for the existence of even-parity tapes. They are merely an historical accident that continues as a convention. All nine-track tapes use odd parity.

## Modes and Codes

Because data from the machine's memory may contain any combination of binary digits, the writing of such data must be done in odd parity in order to be read back correctly, and odd parity is thus usually called binary mode.

Just as the Hollerith card code is a common standard for data-processing machinery, there is a standard code for seven-track tapes, the binary coded decimal character code (BCD). It is sometimes called external BCD to differentiate it from a BCD code that was used in the internal memory of some older machines. The conversion between the CDC machine's internal character code (called display code) and BCD code is done automatically during tape reading or writing, when coded mode is specified.

## Parity Errors

Some causes of parity errors sensed by a bad parity bit or a bad longtitudinal parity character, are: dust on the tape getting between the oxide layer and the head; oxide rubbing off onto the head enough to make a layer between the head and your tape; the plastic backing of your tape being crimped, dented, or cut; the oxide layer having either a bald spot or a limp which did not record properly; the tape having been stretched by a maladjusted drive; and storing the tape in a hot, cold, humid, or dry place. Generally speaking, physical damage to the tape, bad tape, or something getting between the tape and the tape drive head can cause a parity error.

Tape drives are equipped with separate heads for recording and playback, set up so that immediately after data is written it passes under the head which reads back the data. The tape drive automatically checks the parity on data read back by the second head, and signals a parity error if it is not correct. When this happens, the operating system reads the tape backwards the same number of characters as it tried to write, then writes again trying to get the information out there correctly. If it succeeds in a reasonable number of tries (usually 10), a message is written to your dayfile to indicate that a write parity error which was encountered and overcome on a retry.

On occasion, the read-backward will encounter the previous record gap before the proper number of characters has been read. This can mean either that the read-backward dropped one or more characters, or that there was enough of a gap in the middle cf the record as it was written so that the read-backward stopped at that gap instead of at the beginning of the record. In either case, you get a dayfile message. If you get the message, it is probably a good idea to verify the data on the tape or recreate the tape.

## Rewind and Unload

The drives will accept commands to rewind or unload a tape. A rewind causes the drive to move the tape at high speed back to the load-point marker, where reading or writing will proceed from the beginning of your data.

An unload causes the tape to be rewound to the load-pointer marker, and then rewound further until the leader comes off the take-up reel. The tape is then ready for the computer operator to dismount, and it cannot be accessed by your program again. All tapes should be unloaded after processing on them is completed, so that the operator may remove your tape and make use of the drive for another job.

During the time that a rewind or unload operation is taking place, the drive will reject other commands sent to it. Thus, if you issue an unload on a tape which is in the process of rewinding, you are wasting computer time, because the unload will not be accepted by the drive until the tape is completely rewound. A better way would be not to issue the rewind at all, since unloading always rewinds first.

## Care of Tapes

Ideal temperature and humidity conditions for storing magnetic tapes are $70 \pm 5^{o}$ F and $40 \pm 10\%$ RH. Ranges of $40\text{-}80^{o}$ and 30-70% RH are allowable; however, variations of temperature or humidity outside of these ranges can result in irreversible damage to tapes.

Tapes should be protected against physical damage and possible contamination by plastic cannisters or wrap-around seals.

Tapes should not be exposed to magnetic fields; do not store tapes near electric motors, generators, permanent magnets, etc.

Tapes should be handled with care; do not touch the tape itself, and do not stack heavy objects on tapes, drop them, or do anything which could deform or damage the tape reel.

## THE OPERATING SYSTEM

This section describes some of the restrictions and capabilities of magnetic tape use that are dictated by NOS in contrast to the previous section which was concerned exclusively with characteristics of the medium pertinent to practically all computers.

### Labels

Each reel of tape is assigned a volume serial number (VSN) by the Tape Librarian. This is strictly an installation-dependent function. The VSN is used to file the tape in a rack so that it can be found when requested.

Many tapes have another kind of label: it is common to write, as the first physical records on the reel, specially-formatted, records followed by a tape mark, which are identifiable as a label. This machine-readable label contains the VSN along with some other information. A tape with such a label is called a labeled tape; one without is called an unlabeled tape, even though all tapes, of course, have a human-readable gummed label.

The use of labeled tapes greatly reduces the possibility of operator error. Labeling will ensure that you do not have the wrong input or output tape mounted. Labeling also reduces the amount of operator intervention required to assign a tape. The operator physically mounts the tape. If the VSN, label, etc., match then the operating system will automatically assign the tape to the job.

The exact format of the label information is described in the NOS Reference Manual. The volume header label (as distinguished from the trailer label written at the end of the information on the tape) contains such information as the VSN, the date on which the label was written, and the expiration data of the label (if a retention period was specified).

The trailer label contains a block count of the number of physical records written on the tape. When the tape is read back, an error message will be issued if the number of blocks read does not agree. This usually happens when numerous parity errors have made it difficult to accurately read individual physical records.

## Stranger Tapes

NOS can read and write tapes in formats other than the standard NOS format (to be described later). Such tapes may have whatever blocking scheme and record length you desire, with few restrictions. These tapes are called stranger tapes.

Stranger tapes are specified by the F=S or F=L parameter on the tape REQUEST card or the LABEL card. The use of S indicates that the maximum physical record length is 5,120 characters or 512 60 bit words (the same length restriction applies to binary-mode tapes). The length of L-tape physical records is restricted only by the size of the buffer in your program. Although it would appear off hand that S tapes are unnecessary with the availability of L tapes, the system programs that handle S tapes are more efficient and less subject to certain kinds of error conditions; it is therefore important to specify S whenever possible.

No data conversion occurs with binary-mode tapes; the binary digits are copied from memory to tape and back exactly as they appear in memory. The tapes are odd parity. There are two choices of character codes for nine-track 'coded' tapes - ASCII or EBCDIC. Stranger tapes can be labeled or unlabeled. ANSI standard format labels are supported. Other formats are usable with appropriate user programming.

NOS Tapes

A special internal format is used for tapes to be written and read by programs using the NOS operating system. If you do not specify stranger format, NOS internal format is the default.

NOS internal tapes always have trailer labels. Whether they have header labels depends on the user specification.

Information can be written to NOS tapes only in multiples of ten characters (60 binary digits).

Tape marks are used in NOS tapes only to delimit the header and trailer labels; they are not used to partition data. Sub-divisions are indicated by the operating system adding an eight-character suffix to the physical record, or by writing an eight-character record all by itself. When the tape is read back, the operating system recognizes the suffix, deletes it, and sets a flag indicating an end-of-subdivision condition. The level number of the subdivision is included in the suffix; level 0 is end-of-section, level 17 is end-of-partition.

The default blocking method for NOS tapes is spanning. If, for example, the logical record is 5½ times as large as the physical record size (block size), there are five maximum-length physical records followed by one short one (which has an eight-character end-of-section suffix added to its multiple-of-10 characters data). Each physical record of the logical record except the last one is maximum length, and the data spans the physical records; a logical record that won't fit in one is continued in the next.

## NOS Binary Tapes

All NOS binary-mode tapes are restricted to 5,120 characters per physical record. This restriction is imposed because a small computer (called a "peripheral processor") is used to control the tape drive, and it is programmed such that the entire physical record must fit into its memory at once. The 5,120-frame restriction for seven-track tapes converts to 3,840 frames for nine-track tapes (which hold the same number of 6-binary-digit characters).

## Control Statements

There are three distinct aspects to tape usage - reservation, assignment and release.

## Tape Reservation

The number of tape drives which will be required simultaneously must be specified if it is more than 1. This is done with the RESOURC control statement. Separate specifications are required for nine-track (the NT=n parameter) and seven-track (MT=n parameter) tapes. However, it is more efficient (and required with 6250 density drives) to specify the requirements by density (GE=n, PE=n, etc.).

## Tape Assignment

Tape drives are logically connected to jobs with a LABEL or REQUEST control statement. With the exception of 'multi-labeled' tapes the LABEL and REQUEST statements are interchangeable. The LABEL statement, however, has more features than the REQUEST statement. The format of the LABEL statement is:

```
LABEL(lfn,p1,p2,,,pn)
```

lfn is the logical file name that is to be associated with the specified tape. The lfn must be the first parameter. pn is a series of optional keyword parameters. Some of the parameters commonly used include:

Processing Options (PO). There are approximately 15 PO parameters that fall into 3 categories; a) Ring enforcement in (W) or out (R); b) parity error processing and; c) Inhibiting unload. PO=R (ring out) and PO=W (ring in) are the most heavily used PO options.

Label read or write:

R - label is to be read and compared with parameters specified on the LABEL control card.

W - label is to be written.

Tape Characteristics:

D=d - density of the tape. d may be:

GE - 6250 } 9 Track
PE - 1600 cpi phase encoded
HD - 800 bpi

HY - 800 bpi } 7 Track
HI - 556 bpi
LO - 200 bpi

F=f - Format of file data. Default is standard NOS (F=I)

S - Stranger tape format
L - Long Stranger
SI - NOS/BE System Internal

CV=cd - Code conversion. Default is no conversion. Has meaning only if the tape is read in coded mode.

AS - ASCII code
EB - EBCDIC code

File header label fields:

L=z - Label name. Z is 1-17 characters. Default value is spaces.

T=t - Number of days file is to be retained. t may be 1-3 digits. Default is 0. 999 is permanent. A value greater than 364 results in a value of 999.

Volume serial number:

VSN=vsn - Volume serial number is 1-6 characters used to identify the tape for automatic assignment. The VSN parameter may be specified on the VSN control statement - more about this later. A VSN of SCRATCH is used to specify a scratch tape.

VSN control statement. There are cases when it is convenient to separate the VSN parameter from the LABEL statement. Two cases occur: (a) continuation reels need to be specified, or; (b) the VSN is changing frequently from job to job. The format of the VSN card is:

```
VSN(lfn=vsn,...)
```

where

lfn is the logical file name to be associated with the lfn on the LABEL statement.

vsn is the volume serial number (1-6 characters).

If the file is a multi-volume set, VSN's should appear separated by slashes in the order that the volumes are to be accessed.

```
VSN(TAPE96=7065/2942/1705)
```

The VSN statement can go any place in the control statement stream before the LABEL statement. The recommended place is near the front of the job. If the statement needs to be changed, it will be easier to find.

## Tape Release - RETURN and UNLOAD

As soon as one is finished processing the data on a tape, the tape drive should be released to the operating system so that it can be used by other jobs. There are two control statements that release tapes to the operating system. One (RETURN) releases both the tape and the tape drive reservation while the other (UNLOAD) releases only the tape and retains the tape drive reservation. If UNLOAD is used then it is possible to use many tapes serially without specifying a large requirement for tape drives on the RESOURC card.

Examples

1. Tape creation. The operator assigns a scratch tape (assume 7026).

```
    JOB1.
    .
    .
    .
    LABEL(FNEW,W,D=PE,PO=W,VSN=SCRATCH,L=MYTAPE)
    .
    .
    .
    RETURN(FNEW)
```

2. Reading the tape created in example 1.

```
    JOB2.
    .
    .
    .
    LABEL(FOLD,R,D=PE,PO=R,VSN=7026,L=MYTAPE)
    .
    .
    .
    RETURN(FOLD)
```

3. Separating the VSN card from the LABEL card.

```
    JOB3.
    .
    .
    .
    VSN(FOLD=7026)
    .
    .
    .
    LABEL(FOLD,R,D=PE,PO=R,L=MYTAPE)
    .
    .
    .
    RETURN(FOLD)
```

4. Using multiple tapes (3) serially with no need for a RESOURCE card.

```
JOB4.
.
.
.
VSN(F1=7026)
VSN(F2=SCRATCH)
VSN(F3=1812)
.
.
.
LABEL(F1,R,D=PE,PO=R,L=MYTAPE)
.
.
.
UNLOAD(F1)
.
.
.
LABEL(F2,W,D=DE,PO=W,L=GEORGE)
.
.
.
UNLOAD(F2)
.
.
.
LABEL(F3,F,D=PE,PO=R,L=ZIGGY)
.
.
.
RETURN(F3)
```

5. Using multiple tapes (3) -- two simultaneously and a third serially. Requires a RESOURC card.

```
JOB5.
.
.
.
RESOURCE(GE=1,PE=1)
VSN(F1=7026)
VSN(F2=SCRATCH)
VSN(F3=1812)
.
.
.
LABEL(F1,R,D=PE,PO=R,L=MYTAPE)
LABEL(F2,R,D=GE,PO=W,L=GEORGE)
.
.
.
UNLOAD(F1)
RETURN(F2)
.
.
.
LABEL(F3,R,D=PE,PO=R,L=ZIGGY)
.
.
.
RETURN(F3)
```