COMP 3400 Mainframe Administration[1]

Christian Grothoff
christian@grothoff.org

http://grothoff.org/christian/

[1]These slides are based in part on materials provided by IBM's Academic Initiative.

CMS

Provides application programming and execution environment:

- Commands, user programs
- EXECs (scripts), REXX
- Pipelines
- Xedit (editor)
- Compilers

CMS is Simple

- Single-user oriented
- Shared CMS Nucleus (one image, many running instances)
- High performance but simplistic file system

CMS and CP integration

Commands can be automatically passed to CP:

- If command is not found in CMS, command is passed to CP
- Also, commands prefixed with `#CP` go to CP

CMS File System

- CMS files generally cannot be read or written by other operating systems
- No directory hierarchy!
- Files are named using a file identifier consisting of:
  1. File name (FN)
  2. File type (FT)
  3. File mode (FM) or Directory name (dirname)

File Modes

- File mode letter (A-Z) identifies where minidisk or directory resides, established by ACCESS command
- File mode number (0-6, default 1) used to identify or operate subset of files

Storage

- Minidisks (DASDs): A/191 – user's disk ("/home"), S/190 – system disk ("/") , Y – installed programms ("/usr")
- Shared File Systems (SFS) – remote storage, for example in a z/OS data set ("SERVER:USER.SSL.C.EXAMPLES")
- Byte File System (BFS) – hierarchical file system
- Network File System (UNIX NFS)

Minidisks

There are 3 types of minidisks:

- Permanent (defined in directory)
- Temporary (T-DISC) – destroyed at logoff, use "CP DEFINE" to create
- Virtual disks (V-DISC) – simulated minidisk in system storage; no underlying DASD; avoids I/O overhead

CMS Commands

- Not case-sensitive (CMS converts to uppercase)
- COPY SRC-FN SRC-FT SRC-VOL DST-FN DST-FT DST-VOL
- You can use "=" to preserve corresponding SRC-field
- Also try: LISTFILE and FILELIST
- Immediate Commands interrupt running commands and execute immediately
- HELP – interactive help system

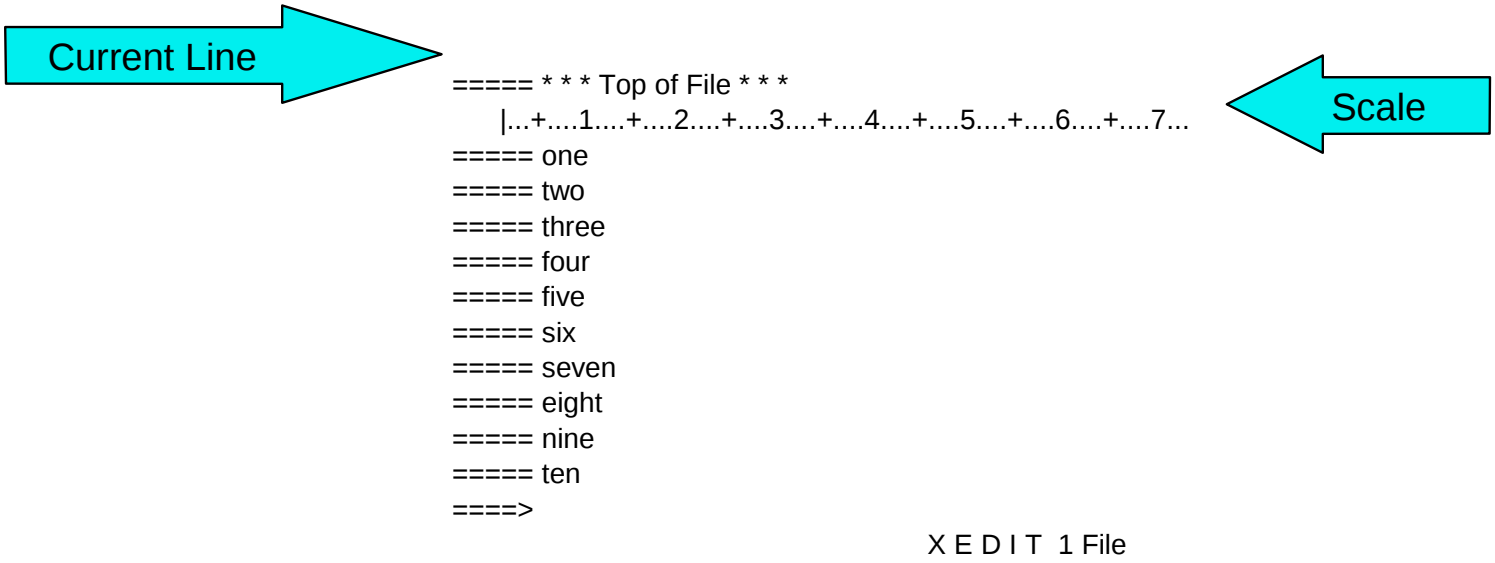
CMS Command Search Order

1. Search for EXEC in storage (A-Z)
2. Search for translation or synonym
3. Search for a module
4. Pass to CP (unless SET IMPCP is OFF)

XEDIT

1. CMS editor
2. Similar to z/OS ISPF editor

XEDIT


TEST FILE A1 F 80 Trunc=80 Size=45 Line=0 Col=1 Alt=0
Current Line
===== * * * Top of File * * *
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7...
Scale
===== one
===== two
===== three
===== four
===== five
===== six
===== seven
===== eight
===== nine
===== ten
====>
X E D I T 1 File

Important XEDIT Prefix Commands

1. m/mm – move
2. c/cc – copy (p to paste)
3. f – following
4. p – preceding
5. a – add
6. si – sequential insert
7. d/dd – delete
8. ”/”” – repeat

Important XEDIT Command Line Commands

1. change /xxx/yyy/ * *
2. /zzz/ – find zzz in text
3. QQuit – leave and don't save changes
4. SAVE – do not leave, but save changes
5. FILE – leave and save changes

EXECs

- Like UNIX shell scripts, just for CMS
- Filetype must be "EXEC" (equivalent to `chmod +x`)
- REXX programs must begin with a comment line
  `/* beginning of REXX program */`
- EXECs do not begin with a REXX-style comment!

PROFILE EXEC

- Like ".profile" or ".bashrc" – executed automatically upon login
- Can suppress execution using "IPL CMS ACCESS (NOPROF"

FORMAT

Use FORMAT to prepare a disk for access:

```
FORMAT 291 c
```

This will format your disk 291 using the mode (label) “c”.

291 is the number assigned using CP’s DEFINE command for the respective DASD.

ACCESS and RELEASE

Use ACCESS to assign a device a mode letter and begin using it (device needs to be DEFINEd or LINKed and possibly FORMATed first):

```
ACCESS 291 c
RELEASE c
```

ACCESS corresponds to `mount`, RELEASE to `umount`.

Exercise

1. LOGON to CMS
2. FORMAT your 191 (A) minidisk
3. ACCESS it
4. (X)EDIT some file

Standard File Operations

- LISTFILE – ls
- FILELIST – mc
- COPY – cp
- RENAME – mv
- ERASE – rm
- PRINT – lpr

SENDFILE

CMS command using the Punch (spool device) to send a file to a reader (possibly belonging to another user):

```
SENDFile FILENAME FILETYPE USERID
```

RECEIVE

To receive a file from the reader (spool device), first start the interactive RDRLIST application:

```
RDRLIST
```

Then type "RECEIVE / FILENAME FILETYPE V" next to the file you want to receive (specifying the target name, type and volume).

Exercise

1. Create a file using XEDIT
2. Send the file to the reader of a student next to you
3. Receive file from another student
4. Leave CMS
5. Query your reader using CP
6. Purge all files in your reader

CMS Pipelines

- Like UNIX pipes in use
- Sligthly different syntax
- NEW: multistream pipelines

CMS Pipelines

```
Pipe < INPUT FILE A % input is a stage!
| drop 4             % like ‘‘eat 4’’
| locate 5.1 /4/     % grep 4 in colum 5
| sort 34-36         % sort by colums 34-36
| > OUTPUT FILE A    % output is a stage!
```

Pipeline Terminology

- Stage – Program that accomplishes a specific task
- Stage Separator – |
- Stream – flow of data into and out of a stage
- Device Driver – stage that interfaces with the environment
- Filter – processes data without interfacing with environment

Common Filters

- locate, find, nlocate, nfind – select records with specified target
- between, inside, outside, ninside – select records between specified targets
- take, drop – select records by counter
- unique, sort unique – select unique records
- sort – sorting
- combine, overlay – combine records
- duplicate – duplicate records

Common Filters

- specs, change, chop, strip, pad – manipulate record data
- block, deblock, split, spill, join, joincont – block and unblock records

Multistream Pipelines

- Multistream pipelines are pipelines that contains stages that have multiple input or output streams

Multistream pipelines introduce a new potential problem: pipeline stalls.

Writing Multistream Pipelines

- Implement primary pipeline; place a label on every stage with multiple input or output streams
- Use the endchar "?" to indicate the end of the primary pipeline
- Write the next pipeline, using the labels to refer to streams from the primary pipeline

CMS Pipelines

```
Pipe < INPUT FILE A
| d:drop 4    % label data with dropped data ‘‘d’
| sort 34-36 % sort primary stream
| i:faninany % merge with input ‘‘i’’
| > OUTPUT FILE A
?              % end of primary pipeline
d: | i:       % connect ‘‘d’’ to ‘‘i’’
```

Pipeline Stalls

- Every stage is waiting for some other stage to perform some function (read or write)
- Cause is usually stage that reads multiple inputs in a particular order (or multiple records)
- Preceding stages may not be able to deliver order or quantity required

When a stall occurs, you receive a return code of "-4095".

Questions

?