Documentation

- This document is only the briefest overview for CP-V.
- Additional documentation can be found in:
- The Andrews University User's Guide, 7th Edition, 1987.
- Xerox CP-V Time-sharing Reference Manual (900907F) and Time-sharing Users Guide (901692D).
- There is a Reference Manual and Operations Manual for most languages.
- More technical details for museum staff is overviewed in another set of slides.

CP-V History

- When SDS delivered the first Sigma 7 in 1966 there was only limited diagnostic software available.
- Users wrote and shared their own software.
- In 1967–71 versions of BPM (Batch Processing Monitor) and BTM (Batch Time-sharing Monitor) were being freely distributed by SDS/XDS to support Batch, Real-time, Remote Batch, and Online (time-sharing) users.
- Although UTS (Universal Time-Sharing system) was announced in 1966, it wasn't delivered until 1971.
- This delay resulted in the name change in 1973 to CP-V or Control Program fiVe due to bad press.

CP-V and Real-Time

- SDS had a long history of providing real-time computers to the scientific community before the Sigma line was developed.
- The Sigma line expanded that line into the 16- and 32-bit world (from the 12- and 24-bit).
- The SDS940 was well-known as a time-sharing system.
- These real-time characteristics made Sigma ideal for time-sharing.
- However, management badly underestimated how long it would take to deliver the product.
- CP-V is event driven with the clocks and I/O completion being the major hardware events.

CP-V Overview

- The V or five in CP-V stand for the five modes of operation.
- These include: batch, online, transaction processing, real-time, and remote batch.
- Ghost can be considered another mode used internally for the operating system or sometimes for real-time support.
- During the 1960s batch processing was the rule.
- Early time-sharing systems were developed but access was generally limited until the 1970s.

CP-V Versions–1

- CP-V A00 was released in 1973 as a successor to UTS D00 and supported the Xerox 560 in addition to the Sigma 6/7/9 line.
- CP-V B00 was released in early 1974 and supported real-time.
- CP-V C00 was released in late 1974 and support transaction processing. It also supported the 7277 disk drives (100 Mb).
- CP-V D00 was released in Dec. 1975 and supported multiprocessing Sigma 9s.
- With Xerox's departure from the mainframe computer business Honeywell started requiring a signed license agreement for software, even if it was provided at no charge.
- Some copies of D00 apparently were released without this formality.
- CP-V operating system releases generally included complete source.

CP-V Versions—2

- Honeywell released CP-V E00 late in 1976. Maximum users was doubled to 254 and multiprocessing 560s were supported.
- CP-V F00 was released by 1978 and supported the Honeywell MPC (MicroProcessor Controlled) tapes and disk. It also extended multi-processor support to the Sigma 6/7s, supported large Sigma 6/7s, and mapped Sigma 5s.
- Honeywell then developed CP-6 to run on 36-bit Honeywell CPUs to provide users an upgrade path.
- Telefile provided Sigma-compatible CPUs (T85) and released TCP-V versions G00, H00, and I00 during the 1980s to support their hardware developments.

AP

- `!AP` is the Assembly Language processor or cross assembler. It replaced `SYMBOL/METASYMBOL` and is MUCH faster.
- It prompts with `WITH>` for options and then `>` if no input file was specified.
- You might enter this sample program after `!AP ME OVER #`:

```
        SYSTEM    SIG7
START   CAL1,9    1
        END       START
```

- If you are lucky it will end with

```
* NO UNDEFINED SYMBOLS
* ERROR SEVERITY LEVEL: 0
* NUMBER OF ERROR LINES: 0
```

APL – A Programming Language

- APL is an old, cryptic, but powerful, generally interpretive language: A Programming Language.
- `!APL` will come back with
  ```
  TELEFILE APL
  CLEAR WS
  ```
- It will then prompt with 6 spaces.
- You might use)TERM 13 if on a Datamedia 3045 or other APL equipped keyboard (lots of strange symbols).
- Otherwise you will use mnemonics: `$I` for ι.
- Example: `$I 10` should give you: `1 2 3 4 5 6 7 8 9 10`
- `)OFF HOLD` will return you to `!TEL`. `)OFF` will log you off.

BASIC

- `!BASIC` will come back with
  `VER. C02`
- The CP-V Basic prompt is `>`
- You might enter the following program:
- ```
  >10 FOR K=1 to 10
  >20 PRINT K,K*K,K*K*K
  >30 NEXT K
  ```
- You might follow that with `>RUN`, `>LIST`, or `>SAVE ON TESTB` commands.
- `>SYSTEM` will return you to `!TEL`; `>BYE` will log you off.
- `>ACC GAMEB` will allow you to load various games there.
  ```

COBOL – COmmon Business Oriented Language

- COBOL is available.
- At AU `COBOL` programs did everything from student records, accounting, alumni, to play `OTHELLO`.
- The COBOL compiler was enhanced beyond the 1968 ANSI standard and supplemented by many subroutines.
- Screen programs were developed under George's guidance by 1975, specifically NIDA for Name, IDentification, and Address.
- By 1978 George had Dan develop a `COBOL` shared library which cost several vendors (Honeywell, Telefile, Western) many millions of dollars in lost memory sales.
- Soon we also had COBOL pure procedure and shared processors.
- The compiler has self-modifying code!

FORT and F77

- `!FORT` is the powerful extended FORTRAN IV version.
- `!F77` is based on the FORTRAN 77 standard which Xerox/Honeywell only completed to the 1976 draft standard.
- However, this version does pass the certification test suite due to work done at Andrews University sponsored by Telefile.
- `!FORT ME OVER #` will prompt for input.
- For a good joke type in GOTO JAIL after column 6.
- You will need to enter "esc F" (end of file) to exit. Or try:
-
```
      DO 100 I=1,100
100   OUTPUT I
      STOP
      END
```

Other Langauges

- During the 1980s Andrews University ported the `C` langauge onto Sigma/CP-V.
- Various implementations of `LISP`, `SNOBOL`, and `PASCAL` also kicked around.
- These are all in various states of resurrection.
- Other common processors of the time included: `RPG`, `IDP`, `CIRC-AC/DC/TR`, and `EDMS`.
- There may also have been `FORTH`, `SL/1`, and `ALGOL`. Only time will tell what has survived.

BATCH

- Batch jobs could be submitted via punch cards and the card reader or as a file via the `!BATCH` processor.
- `!BATCH fid` submits the file `fid` as if it came in via the card reader.
- You will receive information regarding job number and job status.
- Job status may be `WAITING n TO RUN` or `RUNNING`.
- There are 8 batch partitions ranging in size from small to large to encourage a good job mix (high throughput).
- Getting good production (batch job flow) while keeping good online response was the holy grail of the systems programmer.

EDIT, Line Editor—1

- The system editor is entered via `!E fid` or `!EDIT fid` where `fid` is a file identification.
- It may also be entered via `!B fid` or `!BUILD fid` if the `fid` is a new file; or even `!E B fid`.
- There is both a line edit and screen edit mode.
- Typical line edit commands are `IN, TY, SE, MK, DE`, typically followed by a line range.
- Line ranges are from 0.001 to 9999.999 (and beyond).
- Edit files are keyed files with a key max of 3.
- Edit keys are displayed as value/1000, thus key 1.000 corresponds with X'0003E8'.

EDIT, Line Editor—2

- `*TA [A, F, C]` sets tab simulator for the appropriate language. A programmer typically spent a lot of time within the editor!
- `*HX [ON/OFF]` toggles output of subsequent lines in hexadecimal.
- `*FT range,/string/` will find and type `string` in `range`.
- `*AD range` will add to the end of existing records.
- A rich assortment of intrarecord commands to substitute, delete, precede, follow, or overwrite is available and sadly missed in most DOS/Windows editors. Use the `HELP` command for more info.
- The concept of a record is foreign to modern editors.
- Example: `*SE10-20,/str/;3/xyz/P/abc/` would select all records between 10 and 20 with the string `str` and precede the third occurrance in each record of `xyz` with `abc`.

EDIT, Line Editor—3

- `*TY-` will type all lines.
- `*TY0+` will type a screen full of lines starting with 0
- `*+` will print a screen full of lines starting at the end of the current selection.
- `*=` is equivalent to `*+` due to common typographic errors.

EDIT – Display

- Display is the screen edit side of the editor.
- Logically it is implemented as a shared library.
- This shared library `:PDD` is also used by the word processor `TXT`.
- In fact, it was implemented there first and programmers would convert their programs into TXT and edit them there.
- `DI` invokes it; the arrow keys and various control sequences move around. `esc H` brings up a help screen which is terminal type sensitive.

LYNX

- The CP-V post-C00 access to the `!LOADER` is through `!LYNX`.
- Thus PO-Boot tape CPCU doesn't have it.
- There it is only available through `!LOAD` in batch, `!LOAD.X` if the X files are restored, or `!PASS2` (which is generally used to generate the operating system).
- `!LYNX fid` takes its options from file `fid`.
- `!LYNX # over *` will generate `lmn` * from the object #.
- `!LYNX (MO) LYNX.:SYS` will give you a map of `LYNX` (but only if you have access to it).
- PO-Boot tape CPCU does have `!LINK # OVER *`

PCL – Peripheral Conversion Language—1

- `!PCL` is often invoked indirectly via a `!COPY`, `!LIST`, or `!DELETE` command in `!TEL`.
- Of course, these are often shortened to `!C`, `!L`, or `!D`.
- The `!PCL` prompt is < after the salutation of `PCL HERE`.
- Within `!PCL` you can use the full range of commands including `<DELETEALL`, `<COPYALL`, `<WEOF`, *etc.*
- `!PCL` commands can come from a file: `!PCL fid`
- `FL` lists files faster (`FL,.:SYS` or `FL Q.:SYS`).
- The wild card character is `@`: `!FL MONDMP@.:SYS`.
- `LIST.X` and `LISTR.X X.:SYS` were commonly used.

PCL – Peripheral Conversion Language—2

- Additional useful PCL (TEL) commands are as follows:
- `!COPY fid to ME(X)` [displays file in hex]
- `!COPY fid to ME(K)` [shows the file keys]
- `!COPY fid OVER fid(LN)` [adds edit keys]
- `!COPY fid(C) OVER fid(LN)` [uncompressed file]
- `!COPY fid1 INTO fid2` [copies one file into another]
- `!COPY fid1 TO fid2` [same as ON?]

SORT/MERGE

- The `!SORT` processor would sort files on multiple keys, various key lengths, and in various directions.
- `!SET F:SORTIN/fid1; !SET F:SORTOUT/fid2`
- In batch use `!ASSIGN` commands.
- `!SORT`
- `.REC(reclen)`
- `.KEY(beginpt1,len1),(beginpt2,len2)`
- esc F (in batch use a `!EOD`)
- There are also subroutines available to pass a card image of information to `!SORT` for internal program use.

SPLIST - Symbiont Lister

- AU developed `SPLIST` to manage the print symbiont.
- Aka `SPOOL` and `UNSPOOL` are just copies but will roll symbionts to/from tape.
- `SPLIST`'s main prompt is $<$. For symbionts the prompt is -.
- Without `<LOOK` it justs lists everything when you enter a blank line. With `LOOK` it stops and lets you enter a page number and then displays the current symbiont starting at that page.
- It has a powerful parser so that you can select by `<FORM=0 AND ID>10 AND PAGE<99` for example.
- `-DELETE` lets you purge the output queue (75 slots).
- `-FILE=fid` will direct subsequent output to file `fid`.

TELL (and TELLG)

- `!TELL` was George Plue's early attempt to make the system friendly or perhaps add some artificial intelligence.
- `!TELL ME 'HI' AT 16:34` should give you the message 'HI' at that given time.
- `!TELL ME WHEN JOB 111 IS COMPLETED` should send you a message when job 111 has printed.
- `!TELL ME WHEN JOB 111 IS DONE` should tell you when job 111 has finished running.
- `!TELL` communicates with the ghost job `TELLG` who keeps track of the various requests and does the behind the scenes work.

TXT, word processor

- The Xerox provided word processor was known as `TEXT`.
- As was typical of that era it was not `WYSIWYG` (what you see is what you get), meaning control codes were not rendered on the screen, but rather performed when a printout was requested.
- In 1978 Frank Clark at AU started writing `TXT` and it was used for many school papers, even dissertations.
- It supported proportional fonts on the `Diablo` printing terminals and `HP` laser printers.
- Although internally it used non-edit files, its command structure and file display was much like `EDIT`.
- Its prompt was `-?-` and it has an extensive `-?- HELP` command.

CCI – Control Command Interpreter

- Batch uses a different command processor than online.
- `CCI` interprets the `JCL` submitted for batch jobs.
- `JCL` means Job Control Language.
- Commands such as `!JOB`, `!LIMIT`, `!ASSIGN`, and `!RUN` are used.
- Later versions of the operating system replaced the `!ASSIGN` with `!SET`, `!RUN` with `!lmn`, and `!LOAD` with `!LYNX` to make it similar to the online environment.

DELTA—1

- Technically `DELTA` is not a command processor.
- However, it resides in the special processor region (like `!TEL`).
- `DELTA` is the system level debugger.
- Delta command are cryptic. Useful commands are below.
- ;I= (tab) will tell you the current instrution and go to it.
- There you can use) to single step.
- You can also linefeed (cntrl J) and ^ to go one location down (higher addresss) or up (lower address). This is useful for skipping instructions.

DELTA—2

- . refers to the current location. .;B sets a breakpoint there.
- ;S will retrive symbols. .;BT is a tracing breakpoint. 1;B resets breakpoint 1.
- .A000;1 will set the lower search bound (;2 for upper).
- value;w will search for value; .FFFFEF0F;M would set the search mask.
- 999;R will up the default from 80 for using relative symbols.
- 'ABCD'= will convert (up to 4 bytes) to EBCDIC.
- location/ will open location. (I=/(S= will interpret the value as integer/short floating.
- `!RUN * UNDER DELTA;`, `!U n/l *.`, and esc esc `!DELTA` are common ways to invoke `!DELTA`.

LOGON

- If you enter the `!TEL` command `!LOGON` it will log you off!
- `!LOGON` handles logging users onto and off the system.
- `HI TCP-V HERE - L. C. M. COF`
  `08:15 DEC 18,'14 ON LN E  USER# 17  LINE# E`
  `LOGON PLEASE:`
  is its prompt.
- Echoplex (`esc E`) will be turned off so you won't be able to see what you are typing.
- If the museum Sigma 9 is up and running, try entering `LCM GUEST SIGMA9`
- When you log off it will give you the CPU time uses (in minutes), connect time (in HH:MM), interactions, and "charge" in $.

TEL – Terminal Executive Language

- `TEL` is the normal command processor used by online users.
- Its command prompt is `!` lovingly known as the "bang."
- Historically, there was also `!EASY` intended for `BASIC` users.
- From `!TEL` you can invoke various languages and utilities.
- You can also inquire about system, user, or terminal status: `!DISPLAY`, `!STATUS`/`!SHOW`, and `!TERMINAL`
- You can control device and file streams via `!LDEV`/`!LIST` and `!SET` commands.
- You can also log off the system: `!OFF`

Online Usage

- Many escape sequences will be encountered online.
- esc Q will return with !! if the system is up/running.
- esc R will reprint the current input line.
- esc X will delete the current input line.
- esc Y, cntrl Y, and esc esc generally return you to `!TEL`. From there you can continue (`GO`) or abort (`Q`) the current job step.
- esc E toggles echoplex (seeing what you type).
- Many other terminal modes can be listed via the `!T` command.

Online Modes

- esc T toggles tab simulation. `*TA A/F/C` in `!EDIT` will set your tabs for the corresponding language (`AP`, `FORTRAN`, `COBOL`). If tab simulation is off and `!EDIT` encounters a tab it will tell you.
- esc U toggles upper case restrict.
- esc P toggles paper tape.
- esc S toggles space insertion (so tabs are not inserted).
- esc (/) forces lower case shift off/on.
- esc C toggles relative tabbing.
- esc O toggles backspace edit mode.
- Usually on screen terminals rubout will do a backspace space backspace sequence.
- esc lf and esc cr do local linefeeds/carriage returns.

More TEL commands

- `!TABS` will list what tabs are set, `!TABS 10,19,37` will set those tab stops.
- `!T 37` sets the terminal type to 37. `!T 13` may be valid on an APL terminal (Datamedia 3045).
-

KEYIN

- KEYIN processes some but not all operator commands.
- KEYIN is loaded as a monitor overlay but runs as a ghost.
- As such it has its own dedicated ghost table slot as user 1.
- Device retries do not get processed by KEYIN.
- KEYIN does process DATE, TIME, ZAP, GJOB, ON, ONB, DELE, MOUN, REQU, DISP, and X commands.
- When the system is sick console input commands might give the message LATER.

RECOVER/RVGHOST

- `RECOVER` lives in limbo land of being a monitor overlay and ghost.
- Technically it is not an overlay since it doesn't get loaded with nor run with the running monitor.
- However, it gets loaded into location X'4000' and thus overlays monitor procedure while processing monitor data.
- `RECOVER` is associated with crash recovery, closing files, saving symbionts, logging off users, and otherwise cleaning things up.
- `RVGHOST` processes a special kind of crash file, `DUMPFILE`, which is used in a Single User Abort.
- `DUMPFILE` is copied onto a normal `MONDMPx` file, where $0 \leq x \leq 7$, for normal processing by `ANLZ`. `DUMPFILE` is a random file with 1 granule per genned system memory page + users.

GHOST1

- GHOST1 also runs as user 1 and is instrumental in bringing the system up.
- After certain monitor initialization the monitor exits with interrupts on to the scheduler with two users, GHOST1 and ALLOCAT.
- GHOST1 has to determine what type of boot is occurring: tape, disk, crash recovery, and act accordingly.
- GHOST1 tries to retain the input/output symbionts and device partitioning information.
- GHOST1 exits by transforming into FILL and the system is up.
- FIX sometimes exits by transforming into GHOST1.
- GHOST1 also kicks off MOOSE, but at AU we made that DAY.

ALLOCAT

- `ALLOCAT` is the granule allocation ghost and user 2.
- His data area contains the `HGP`s or Head Granule Pointers.
- `ALLOCAT` gets called to run when either the "in-core" stacks of granules are too high or too low.
- These "stacks" are actually queues for faster processing.
- These queues contain word entries with generalized disk addresses.
- Whereas, `ALLOCAT`'s data is maintained as a bit table.
- A generalized disk address contains a DCT index in bits 11–15, and the relative sector number in bits 8–10, 16–31.
- Bits 8–10 were interpretted differently by different vendors/users/releases (big/little endian).

RBBAT

- RBBAT is the symbiont processing ghost and user 3.
- Unlike ALLOCAT, RBBAT does not have a dedicated swap area for his data so it is often lost.
- Hence the common message by GHOST1: Batch Queue Lost.
- SPLIST/SPOOL/UNSPOOL makes a copy of RBBAT's data in RBDATA.:SYS
- DAY may copy it periodically onto RBDATA1 for processing by SPLIST/SPOOL via the <CRASH option.
- RBBAT will occasionally experience "morning sickness" and will need to be revived via a crash.

FIX/FILL/DAY

- `FIX` is an essential part of the operating system but was not working well in CP-V C00.
- As such we typically had it exit (except on PO boots).
- The main idea was to process bad files (750x type errors).
- `FIX` sometimes exitted to `GHOST1`.
- `GHOST1` exits to `FILL` and starts `MOOSE/DAY`.
- The `DAY` ghost prints the date and uptime on the console every 30 minutes, and will nag the operator about allowing online/batch users, and low granules.

ERR:FIL

- The monitor logs errors into “in-core” error buffers.
- When full these error buffers get written to disk.
- The `ERR:FIL` ghost is responsible for moving these error log entries into `ERRFILE.:SYS`.
- Typically, every week, after the weekly save, this file would be summarized via `ELLA` and deleted.

9DIAG

- Intermittant hardware errors are the bane of technical support.
- Collecting information on intermittant errors is key to solving them.
- `9DIAG` was written with this in mind.
- Typically it uses 1% or 10% of the CPU time to run the main 4 CPU diagnostics: `9AUTO`, `9SUFFIX`, `9FLOAT`, and `9DECIMAL`.
- Some tests have to be skipped to be compatible with CP-V, such as `LPSD`, `XPSD`, `MMC`.
- However, a vast part of the complex SIGMA 9 CPU can be tested every few seconds.