Programming Languages for the Commodore 64:

PROMAL Revisited

Glenn Holmer/VCFMW 2016

speaker bio

- a.k.a. "Cenbe", "ShadowM"
- web site at www.lyonlabs.org
- long-time collector of Commodore 64 compilers/interpreters/assemblers
- a neglected area for Commodore 64 enthusiasts and collectors
- this is part of a series of talks on some of the more interesting programming languages available

PROMAL revisited

- spoke about PROMAL here in 2014
- had just acquired the full version after decades of searching
- the beer goggles are off!
- let's talk about some of PROMAL's shortcomings... and how they might be addressed

PROMAL shortcomings

- only two drives supported (not uncommon for software of that time)
- drives named 0: and 1: (confusing for Commodore users)
- can't easily change which device 0: and 1: point to
- can't issue disk commands to arbitrary devices (just 0: and 1:)
- lack of good disk utilities!

PROMAL disk utilities: getcmddate

- takes device number as argument
- sends T-RD (time read decimal), then sets PROMAL system variables
- 31,SYNTAX ERROR means no clock on device
- uIEC response inconsistent w/CMD
- will run from **bootscript.j** (autoexec batch script)

"diskutils" module

- getlfn (get first free file number)
- drvquery (query attached drives)
- drvdesc (get drive type desc.)
- getparttype (get partition type)
- partdesc (get part. type desc.)
- sendcmd (send disk command)
- readcmd (read cmd. channel output)
- getdirhead (get directory header)
- readsect (read disk sector)

PROMAL disk utilities: drives

- calls drvquery, which loads an exported 23-byte array holding the drives list (8-30)
- if drives[dev-8] is 0, no device attached at that address
- can look up the drive type with drvdesc to get human-readable type

PROMAL disk utilities: pwd

- easy on a CMD drive, which provides a track and sector pointer to the parent directory header
- more of a kludge on uIEC; you have to use CD← until it fails, then CD back to the constructed path (but it works within a DNP image)
- both can use G-P command for info
- consider path to be // if you're in a 15x1 emulation partition on CMD (or a disk image on the uIEC)
- 1581 "directories" not supported

PROMAL disk utilities: ls

- what's the PROMAL equivalent of "malloc some dirent structs"?
- can't just fake a struct with a series of variables because arrays and scalars are stored differently
- solution: use byte array w/offsets
- memory allocation done manually based on system variables
- options: drive number, wildcards, -n (sort by name), -t (sort by timestamp on SD or CMD drive)

PROMAL disk utilities: cp

- supports copying across devices, partitions, and directories
- -s and -d required for device nos.
- CMD syntax
- proper wildcards (not like C=)
- can use "." as destination
- prompts for replace
- only supports PRG and SEQ

bottom line: pros

- good high-level language features (looping, local variables)
- good low-level language features (pointers!)
- good multi-module support, sophisticated loader
- support for assembly modules (with jsr keyword or jump table)
- shell with batch scripts and command recall
- excellent documentation, access to internals

bottom line: cons

- ~~limited drive support~~
- ~~poor disk utilities~~
- no linkage editor, have to manually load library modules (and dependent code has to be recompiled if the library module changes)
- keeping modules loaded wherever possible requires frequent use of "unload" command

conclusion

When all is said and done, still (currently) my favorite language to use on the Commodore 64.

PROMAL RULEZ OK!

resources

- visit my site (www.lyonlabs.org) to get disk images and documentation:

  /commodore/onrequest/PROMAL/index.html

- there's also a cheat sheet there with commonly used commands &c.

- I can demo PROMAL and other programming languages at my table

DEMO

(slides and sample code
are on my web site)