VirtualCpm
CP/M Emulator
December 3, 2022

Table of Contents

Introduction

VirtualCpm is a JAVA program that amulates a CP/M CCP, BDOS, and BIOS running on a Z80 with 64K RAM, with CP/M 3 extensions. It is normally used to execute a single CP/M command, but is capable of handling “submit” files as well as native (on the host PC) text files filled with commands.

Native Files

Native files must have lower-case only names. Mixed-case filenames will cause unpredictable results. All files created by CP/M will be in lower-case.

The file’s write permission is used to reflect the CP/M RO attribute. CP/M programs that change a file’s RO attribute will change the native file’s write permission.

The file’s execute permission is used to reflect the CP/M SYS attribute. Note that Windows will always show files as executable, and thus files on a Windows host will always have the SYS attribute set. Also remember that CP/M normally hides files that have the SYS attribute set. There is a server configuration setting that disables the SYS attribute, to avoid these issues on Windows hosts.

The CP/M ARCHIVE attribute is not supported.

Files that are not an even multiple of 128 bytes in size will be padded to a 128-byte multiple, using Ctrl-Z (EOF, 0x1a), when reading. Writing to a file from CP/M always involves a full 128-byte record, so no additional padding is performed. The CP/M 3 feature “Set File Byte Count” will truncate a file to a specific, arbitrary, number of bytes, after which the file may no longer be an even multiple of 128 bytes.

Configuration

The emulator is configured using a “configuration file”, which is plan text formatted as “property = value” lines. The configuration file to be used is will look in the current directory for “**vcpm.rc**” and then the user’s home directory for “**.vcpmrc**”. Comments in the file start with ‘#’ as the first non-space character of the line. Property values may use backslash (‘\’) to extend long values to the following line.

The following properties are recognized:

vcpm_dso = *string*
: Specifies the Drive Search Order used to locate commands. Accepts drive letters and the keyword “def”, separated by commas. Only four items are allowed. Default is “def,a”.

silent
: Suppresses informative messages about configuration, etc.

vcpm_root_dir = *path*
: Specifies the top-level (root) directory to be exported to CP/NET. Subdirectories named “a” through “p” are assumed, but not created automatically. Default will be “**~/HostFileBdos**”.

vcpm_nosys
: Disable the CP/M SYS attribute, so files will not be hidden on Windows.

vcpm_drive_*X* **=** *path*
: Where “*X*” is one of “a” through “p”. Specify the path to use instead of “root_dir/X” for the CP/M drive.

Environment Variables

The following environment variables are recognized:

CPMDrives = *path-list*
: The *path-list* is a list of (up to) 16 paths separated by commas. Each path is positional dependent, i.e. the first path is for drive A:, second for drive B:, etc. If a path is empty (“,,”), then the default is used. All non-empty paths are translated into **vcpm_drive_***X* properties.

CPMDrive_*X* = *path*
: Specifies the path to use for drive *X*. This is translated into a **vcpm_drive_***X* properties.

VCPMCoreDump = *anything*
: Triggers a core dump of RAM when exiting the emulation. The dump file is xxxx and contains the raw (binary) contents of RAM.

CPMDefault = *usr-drv*
: Specifies the default/initial user number and/or drive. Default is “0A:”.

Running the Emulator

Typical startup command:

```
java -jar VirtualCpm.jar cpm-command
```

Typically, a wrapper script or batch file will be created, to simplify running commands. On Linux, this might look like:

```
#!/bin/bash
exec java -jar /path/to/VirtualCpm.jar "${@}"
```

If this script were named “vcpm”, and is on the user’s PATH, then one would run CP/M commands using:

```
vcpm cpm-command
```

Makefile Example

Here is an example Makefile that uses vcpm to assemble using RMAC and link the result into an SPR file, using source files “prog1a.asm” and “prog1b.asm”. This makefile and the source file reside in the same directory. The default VirtualCpm A: drive must contain RMAC.COM and LINK.COM, and any other necessary files (like Z80.LIB).

```
export CPMDrive_D = $(PWD)
export CPMDefault = d:

all: prog1.spr

%.rel: %.asm
	vcpm rmac "$?" '$$SZLA'

prog1.spr: prog1a.rel prog1b.rel
	vcpm link "prog1=prog1a,prog1b[os,nr]"
```