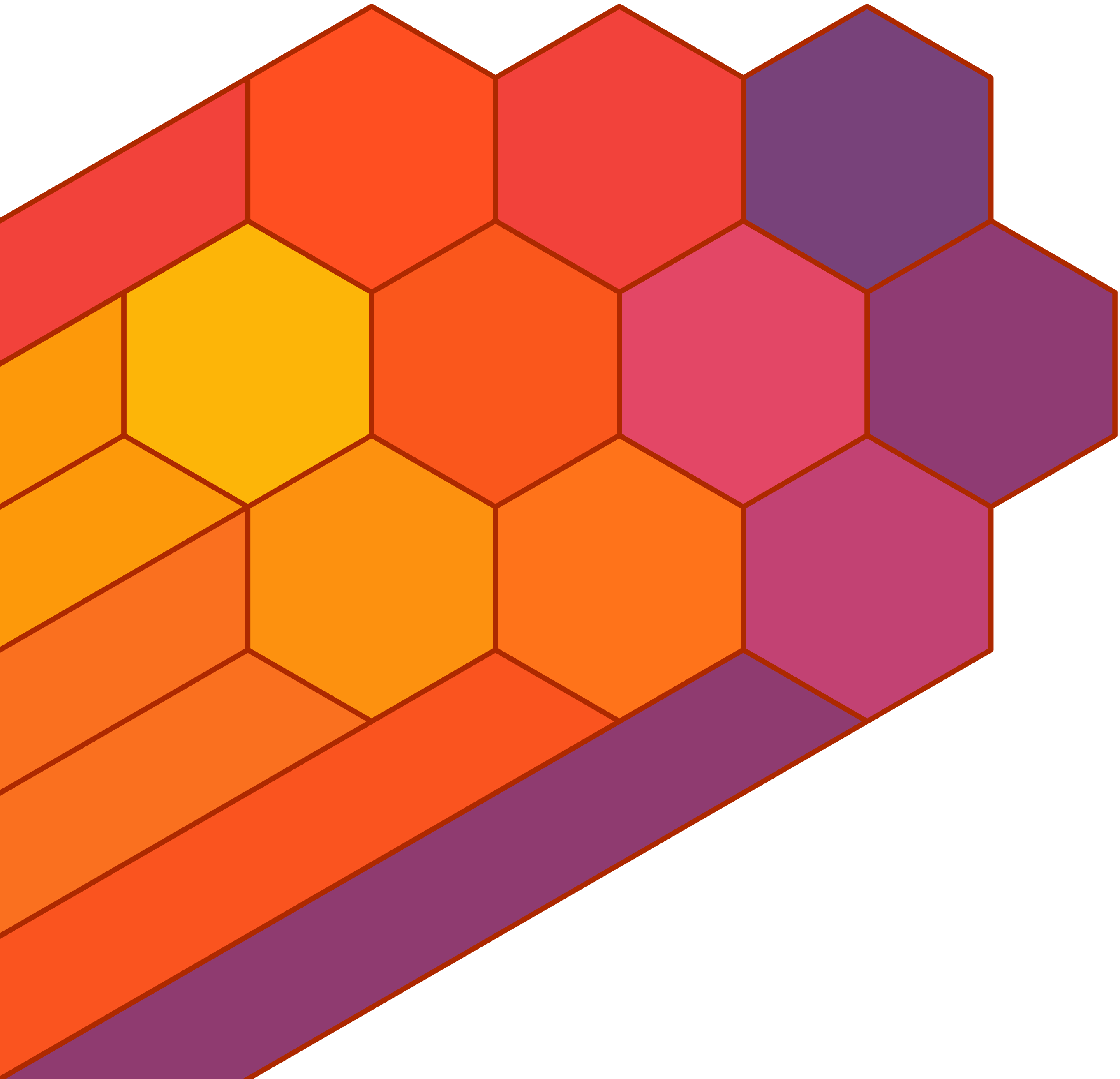
ucsd-psystem-vm
UCSD p-System Virtual Machine

Reference Manual

Mario Klebsch
mario@klebsch.de



.

This document describes ucsd-psystem-vm version 0.7
and was prepared 6 April 2010.

NAME

ucsd-psystem-vm − UCSD p-System virtual machine

DESCRIPTION

The *ucsd-psystem-vm* package is a virtual machine (p-code interpreter) for the UCSD p-System.

The *ucsd-psystem-vm* package makes it possible to run Apple Pascal (an ancient programming environment) on today’s Linux machines, and also *BSD. The p-code runs very much faster on modern hardware.

This p-code interpreter ws developed using the 64K system of Apple Pascal Version 1.3. Versions 1.2 and 1.1 also work, are not as excessivly tested.

This is still experimental code. If you are in trouble, “Use the source, Luke!”

Running the Virtual Machine

For instructions on how to run the virtual machine, includeing details of how to fetch and manipulate disk images, see the *ucsdpsys_vm*(1) man page.

Nostalgia

If you are still reading, prepare to enter the nostalgic world of Apple Pascal...

```
Command: E(dit, R(un, F(ile, C(omp, L(ink, X(ecute, A(ssem, D(ebug,? [1.1]

Welcome APPLE1, to Apple II Pascal 1.1

Based on UCSD Pascal II.1

Current date is 27-Sep-86

(C) Apple Computer Inc. 1979, 1980
(C) U.C. Regents 1979
```

BUILD

You will find complete instructions for building the *ucsd-psystem-vm* package in the BUILDING file in the source tarball.

COPYRIGHT

ucsd-psystem-vm version 0.7
Copyright © 2000, 2001 Mario Klebsch
Copyright © 2006, 2009, 2010 Peter Miller

The *ucsd-psystem-vm* programs come with ABSOLUTELY NO WARRANTY. This is free software and you are welcome to redistribute it under certain conditions. For details see the *ucsdpsys_vm_license*(1) man page.

AUTHOR

Mario Klebsch	E-Mail:	mario@klebsch.de
	WWW:	http://www.klebsch.de/

MAINTAINER

Peter Miller	E-Mail:	pmiller@opensource.org.au
/\/*	WWW:	http://miller.emu.id.au/pmiller/

RELEASE NOTES

This section details the various features and bug fixes of the various releases.

Version 0.7 (2010-Apr-06)

The project now uses GNU Autoconf to configure. The tarball compiles “out of the box”. Pre-built Debian packages are available on the web site.

All of the commands consistently use getopt_long for argument parsing. All commands now come with a man page.

The project uses libexplain for better error reporting of Linux and Unix errors.

Version 0.5 (2001-Aug-02)

Long integer support has been added.

Version 0.4 (2001-Jun-08)

This time, protability has been improved. Disk access now functions even when *mmap*(2) is not available, using *lseek*(2), *read*(2) and *write*(2) instread.

The code to implement the Turtle Graphics (and all dependencies on X11) has been moved to a separate process. This should make porting to other platforms much easier, because this is simply a dedicated server that reads turtle commands. (For historical reasons, it is called at the moment but still *ucsdpsys_xturtleserver*(1), his background, he can not even hide it completely).

The PRINTER: Device is now also implemented, otherwise faulty are a few P-commands have been corrected.

Anyone who had a problem with the F(iler the eX(amine command crashing, should try again now.

Version 0.3 (2001-May-20)

The build instructions have been simplified. The Apple Pascal 1.1 demo now runs.

Version 0.2 (2001-May-10)

Things have improved. I now have Apple Pascal 1.3 and 1.1 version working. Running is perhaps slightly exaggerated, but at least the compiler seems to work.

Version 0.1 (2000-Sep-21)

Many years ago I learned programming at school using Apple-Pascal. From nostalgic feelings out (and because it was the best Pascal system, the more I could use), I’ve started to code interpreter for a p-Machine. Actually, not yet ripe for a publication, but in a discussion in de.alt.folklore.computer was considerable interest expressed in this software.

Although this program is the status of best-Alpha, it’s here to download ready. Documentation non-existent, unfortunately. Use the source, Luke! Use the source, Luke!

NAME

How to build ucsd-psystem-vm

BEFORE YOU START

There are a few pieces of software you may want to fetch and install before you proceed with your installation of ucsd-psystem-vm.

libexplain

The *ucsd-psystem-vm* package depends on the libexplain package, a library of system-call-specific *strerror*(3) replacements.
http://libexplain.sourceforge.net/

X11

The *ucsd-psystem-vm* package depends on the X11 libraries, in order to emulate Turtle Graphics. Only the core functionality and Xt widgets are used.

GNU Groff

The documentation for the *ucsd-psystem-vm* package was prepared using the GNU Groff package (version 1.14 or later). This distribution includes full documentation, which may be processed into PostScript or DVI files at install time − if GNU Groff has been installed.

SITE CONFIGURATION

The **ucsd-psystem-vm** package is configured using the *configure* program included in this distribution.

The *configure* shell script attempts to guess correct values for various system-dependent variables used during compilation, and creates the *Makefile* and *lib/config.h* files. It also creates a shell script *config.status* that you can run in the future to recreate the current configuration.

Normally, you just *cd* to the directory containing *ucsd-psystem-vm*'s source code and then type

```
% ./configure
...lots of output...
%
```

Running *configure* takes a minute or two. While it is running, it prints some messages that tell what it is doing. If you don't want to see the messages, run *configure* using the quiet option; for example,

```
% ./configure --quiet
%
```

To compile the **ucsd-psystem-vm** package in a different directory from the one containing the source code, you must use a version of *make* that supports the *VPATH* variable, such as *GNU make*. Change directory to the directory where you want the object files and executables to go and run the *configure* script. The *configure* script automatically checks for the source code in the directory that *configure* is in and in .. (the parent directory). If for some reason *configure* is not in the source code directory that you are configuring, then it will report that it can't find the source code. In that case, run *configure* with the option `--srcdir=`*DIR*, where *DIR* is the directory that contains the source code.

By default, *configure* will arrange for the *make install* command to install the **ucsd-psystem-vm** package's files in */usr/local/bin*, and */usr/local/man*. There are options which allow you to control the placement of these files.

`--prefix=`*PATH*

This specifies the path prefix to be used in the installation. Defaults to */usr/local* unless otherwise specified.

`--exec-prefix=`*PATH*

You can specify separate installation prefixes for architecture-specific files files. Defaults to *${prefix}* unless otherwise specified.

`--bindir=`*PATH*

This directory contains executable programs. On a network, this directory may be shared between machines with identical hardware and operating systems; it may be mounted read-only. Defaults to *${exec_prefix}/bin* unless otherwise specified.

`--mandir=`*PATH*

This directory contains the on-line manual entries. On a network, this directory may be shared between all machines; it may be mounted read-only. Defaults to *${prefix}/man* unless otherwise specified.

The *configure* script ignores most other arguments that you give it; use the `--help` option for a complete list.

On systems that require unusual options for compilation or linking that the *ucsd-psystem-vm* package's *configure* script does not know about, you can give *configure* initial values for variables by setting them in the environment. In Bourne-compatible shells, you can do that on the command line like this:

```
$ CC='gcc -traditional' LIBS=-lposix ./configure
...lots of output...
$
```

Here are the *make* variables that you might want to override with environment variables when running *configure*.

Variable: CC

C compiler program. The default is *c++*.

Variable: CPPFLAGS

Preprocessor flags, commonly defines and include search paths. Defaults to empty. It is common to use `CPPFLAGS=-I/usr/local/include` to access other installed packages.

Variable: INSTALL

Program to use to install files. The default is *install* if you have it, *cp* otherwise.

Variable: LIBS

Libraries to link with, in the form `-l`*foo* `-l`*bar*. The *configure* script will append to this, rather than replace it. It is common to use `LIBS=-L/usr/local/lib` to access other installed packages.

If you need to do unusual things to compile the package, the author encourages you to figure out how *configure* could check whether to do them, and mail diffs or instructions to the author so that they can be included in the next release.

BUILDING UCSD-PSYSTEM-VM

All you should need to do is use the

```
% make
...lots of output...
%
```

command and wait. When this finishes you should see a directory called *bin* containing several programs.

If you have GNU Groff installed, the build will also create a *etc/reference.ps* file. This contains the README file, this BUILDING file, and all of the man pages.

You can remove the program binaries and object files from the source directory by using the

```
% make clean
...lots of output...
%
```

command. To remove all of the above files, and also remove the *Makefile* and *lib/config.h* and *config.status* files, use the

```
% make distclean
...lots of output...
%
```

command.

The file *etc/configure.in* is used to create *configure* by a GNU program called *autoconf*. You only need to know this if you want to regenerate *configure* using a newer version of *autoconf*.

INSTALLING UCSD-PSYSTEM-VM

As explained in the *SITE CONFIGURATION* section, above, the *ucsd-psystem-vm* package is installed under the */usr/local* tree by default. Use the `--prefix=`*PATH* option to *configure* if you want some other path. More specific installation locations are assignable, use the `--help` option to *configure* for details.

All that is required to install the *ucsd-psystem-vm* package is to use the

```
% make install
...lots of output...
%
```

command. Control of the directories used may be found in the first few lines of the *Makefile* file and the other files written by the *configure* script; it is best to reconfigure using the *configure* script, rather than attempting to do this by hand.

GETTING HELP

If you need assistance with the *ucsd-psystem-vm* package, please do not hesitate to contact the author at

```
Peter Miller <pmiller@opensource.org.au>
```

Any and all feedback is welcome.

When reporting problems, please include the version number given by the

```
% ucsdpsys_mount -V
ucsdpsys_mount version 0.7.D001
...warranty disclaimer...
%
```

command. Please do not send this example; run the program for the exact version number.

COPYRIGHT

ucsd-psystem-vm version 0.7
Copyright © 2000, 2001 Mario Klebsch
Copyright © 2006, 2009, 2010 Peter Miller

The *ucsd-psystem-vm* programs come with ABSOLUTELY NO WARRANTY. This is free software and you are welcome to redistribute it under certain conditions. For details see the *ucsdpsys_vm_license*(1) man page.

AUTHOR

Mario Klebsch	E-Mail:	mario@klebsch.de
	WWW:	http://www.klebsch.de/

MAINTAINER

Peter Miller	E-Mail:	pmiller@opensource.org.au
/\/*	WWW:	http://miller.emu.id.au/pmiller/

NAME

ucsdpsys_svolcvt − de-interleave Apple Pascal disk images

SYNOPSIS

ucsdpsys_svolcvt < *svol-file* > *dsk-file*
ucsdpsys_svolcvt < *dsk-file* > *svol-file*

DESCRIPTION

The *ucsdpsys_svolcvt* program is a utility to convert USCD volume images from `.svol` format to `.dsk` format and vice versa. The conversion is symetric, no options are needed to specify which conversion is needed.

OPTIONS

The following options are understood:

−V Print the version number and exit.

SEE ALSO

ucsdpsys_vm(1)
UCSD p-System virtual machine

ucsdpsys_svolio(1)
manipulate UCSD p-System disk images

BUGS

The size of the volume file must be multiple of 4KB (the Apple][track size)

EXIT STATUS

The *ucsdpsys_svolcvt* command will exit with a status of 1 on any error. The *ucsdpsys_svolcvt* command will only exit with a status of 0 if there are no errors.

COPYRIGHT

ucsdpsys_svolcvt version 0.7
Copyright © 2000, 2001 Mario Klebsch
Copyright © 2006, 2009, 2010 Peter Miller

The *ucsd-psystem-vm* programs come with ABSOLUTELY NO WARRANTY. This is free software and you are welcome to redistribute it under certain conditions. For details see the *ucsdpsys_vm_license*(1) man page.

AUTHOR

Mario Klebsch	E-Mail:	mario@klebsch.de
	WWW:	http://www.klebsch.de/

MAINTAINER

Peter Miller	E-Mail:	pmiller@opensource.org.au
/\/*	WWW:	http://miller.emu.id.au/pmiller/

NAME

ucsdpsys_svolio − UCSD p-System volume I/O

SYNOPSIS

ucsdpsys_svolio −e *volume-file*
ucsdpsys_svolio −l *volume-file*
ucsdpsys_svolio [**−t**] *volume-file content-file*

DESCRIPTION

The *ucsdpsys_svolio* program is used to read UCSD volume images. It can list the directory and read files in binary mode as well as in text mode. When reading files, the data is written to stdout.

The *volume-file* can be either in `.svol` format or in `.dsk` format, used by Apple][emulators. The `.svol` format does not specify a byte order. The *ucsdpsys_svolio* program tries to automagically determine the correct byte order. The following options are understood:

−e *volume-file*
show extended directory listing of *volume-file*.

−l *volume-file*
show the directory of *volume-file*.

−t Read the file using text mode. In text mode, the first two blocks are skiped (they do contain UCSD editor internal data), DLEs and line endings are converted.

−V Print the version and exit.

EXAMPLES

To read the contents of the volume file `apple1.svol`, execute

```
ucsdpsys_svolio -l apple1.dsk
```

To extract `SYSTEM.PASCAL` you can use

```
ucsdpsys_svolio apple1.dsk system.pascal > system.pascal
```

To take (yet another) view of `GRAFDEMO.TEXT`, execute

```
ucsdpsys_svolio -t apple3.dsk grafdemo.text | less
```

SEE ALSO

ucsdpsys_vm(1)
UCSD p-System virtual machine

ucsdpsys_volcvt(1)
convert Apple][Pascal disk images

Apple Pascal Operating System Reference Manual (1980)
Appendix C, File Formats/Text files

BUGS

Can not write to *volume-file*.

EXIT STATUS

The *ucsdpsys_svolio* command will exit with a status of 1 on any error. The *ucsdpsys_svolio* command will only exit with a status of 0 if there are no errors.

COPYRIGHT

ucsdpsys_svolio version 0.7

AUTHOR

Mario Klebsch	E-Mail:	mario@klebsch.de
	WWW:	http://www.klebsch.de/

MAINTAINER

Peter Miller	E-Mail:	pmiller@opensource.org.au
/\/*	WWW:	http://miller.emu.id.au/pmiller/

NAME

ucsdpsys_vm − UCSD p-System virtual machine

SYNOPSIS

ucsdpsys_vm [*option*...]

DESCRIPTION

The *ucsdpsys_vm* program is a UCSD p-System virtual machine. It is able to run UCSD Pascal II.0 and Apple][Pascal (an ancient programming environment) on todays computer systems.

OPTIONS

The following options are understood:

−a

−−apple execute the system in Apple compatibility mode. In Apple compatibility mode, the P-system is initialized using the same addresses as the original Apple][p-System.

−b *batch-file*

−−batch=*filename*

Starts the system in batch mode. Input is read from *batch-file*, output is written to stdout (unless **−x** is used). If '−' is specified as *batch-file*, stdin is used.

−d

−−dump

This options causes the memory images to be written out to a file even when there is no panic.

−f *volume-file*

−−forget=*filename*

Uses *volume-file* for I/O on disk volumes in *forget-changes* mode, which discards all changes to the volume, when *ucsdpsys_vm* exits. This option may be given more than once.

−g

−−trace-max

Enter P-code debug mode.

−n *system-name*

−−name=*system-name*

Uses *system-name* instead of `SYSTEM.PASCAL`. In contrast to the original Apple system, the system file is found on any of the volumes.

−r *volume-file*

−−read=*volume-file*

Uses *volume-file* for I/O on disk volumes in *read-only* mode. This option may be given more than once.

−t *filename*

−−trace-file=*file-name*

Write the p-instructions executed to *trace-file*. This recorded instructions can be compared to a recodring made on an Apple][emulator. If '−' is specified as *trace-file*, the data is written to stdout.

−V

−−version

prints the version number and exits

−w *volume-file*

−−write=*filename*
Uses *volume-file* for I/O on disk volumes in *read-write* mode. This option may be given more than once.

−x

−−xterm
Start an *xterm*(1) for `CONSOLE:` and `SYSTERM:`. Especially usefull when using the debugger and its messages are output to stdout and stderr.

EXAMPLES

A simple method of working with the UCSD p-System is to use *ucsdpsys_vm* with `work.svol` as #4: in read-write mode and `system.svol` as #5: in read-only mode.

```
ucsdpsys_vm -w work.svol -r system.svol
```

To use the original Apple Pascal images, `APPLE1:` on #4:, `APPLE2:` on #5: and `APPLE3:` on #9: you can execute

```
ucsdpsys_vm -w apple1.dsk -r apple2.dsk -r apple3.dsk
```

system.charset

You need SYSTEM.CHARSET from your Apple Pascal system disk in order to use Turtle Graphics. This file contains the bitmapped front for characters. A utility called *ucsdpsys_svolio*(1) can be used to extract system.charset from an UCSD disk image.

```
ucsdpsys_svolio apple0.dsk system.charset > system.charset
```

Alternatively, from the *ucsd-psystem-fs* package, you could use the *ucsdpsys_disk*(1) command

```
ucsdpsys_disk -f apple0.dsk -g system.charset
```

There is an alternative that allows you to create a system.charset from an X11 font, or even create your own. The *bigtext*(1) command comes from the *bigtext* package.

```
bigtext --apple-pascal fixed > system.charset.text
ucsdpsys_charset -e system.charset.text system.charset
```

You can also turn a system.charset file into text so that you can edit it by using

```
ucsdpsys_charset -d system.charset system.charset.text
```

Disk Images

You will need UCSD disk images. You may find disk images on

ftp.apple.asimov.net:/pub/apple_II/images/programming/pascal/ucsd

You can use *wget*(1) to fetc the files

```
wget ftp://ftp.apple.asimov.net/pub/apple_II/images/programming/\
pascal/ucsd/UCSD%20Pascal%201.2_0.DSK apple0.dsk
wget ftp://ftp.apple.asimov.net/pub/apple_II/images/programming/\
pascal/ucsd/UCSD%20Pascal%201.2_1.DSK apple2.dsk
wget ftp://ftp.apple.asimov.net/pub/apple_II/images/programming/\
pascal/ucsd/UCSD%20Pascal%201.2_2.DSK apple2.dsk
wget ftp://ftp.apple.asimov.net/pub/apple_II/images/programming/\
pascal/ucsd/UCSD%20Pascal%201.2_3.DSK apple3.dsk
```

These images are stored in a format used by Apple][emulators. The UCSD p-System has a different view of its disks. Th *ucsdpsys_vm* command is able to work with images in both formats, and automatically recognises which format a disk image is in. Images in the Apple][emulators format are recognized by their `.dsk` extension. The conversion is symmetric and the utility program *ucsdpsys_svolcvt*(1) converts between the two file formates.

```
ucsdpsys_svolcvt example.dsk example.svol
```

If no files are named on the command line, it reads the image on stdin and writes the converted image to stdout.

RUNNING

To boot the Apple Pascal system, you do not need SYSTEM.APPLE (it is replaced by this p-code interpreter). So you just can boot APPLE0: without using APPLE3:. Of course, APPLE1: still is able to boot.

To build your first set of working images, execute 'make system.svol' or 'make work.svol' (each target builds both images). Piping the output through a pager (e.g. less) is a good idea. The files for the images are taken from `ftp.apple.asimov.net`. This build-target also does modify SYSTEM.MISCINFO and rebuilds SYSTEM.LIBRARY using the turtlegrafics replacement library. After building these images, you should be able to boot your system:

```
ucsdpsys_vm -w work.svol -r system.svol
```

As a simple test case, you can try to X(ecute system:grafdemo.

Volumes cannot be mounted or unmounted when the system is running. You have to specify which images are used on the units when starting the virtual machine. The first image will be unit #4, the second one will be unit #5. Units #6, #7 and #8 are reserved for other, non-disk devices (PRINTER:, REMIN: and REMOUT:). So, the third image will be unit #9 and so on.

The system has three modes for the volumes: Read-Only, Read-Write and ForgetChanges. The ForgetChanges mode is read-write, but the data will not make it to the (UNIX) volume image. If the virtual machine is exited, all changes to “forget volumes are lost.

TURTLE GRAPHICS

The original Apple TURTLEGRAPHICS is written partly in Pascal, partly in assembly. Since (hopefully) nothing depends on the internal interfaces between these two parts, I decided to replace the entire library. A volume (library.svol) is included, which contains the source of this replacement library.

```
LIBRARY:
COPYING            40    7-Nov-84     6   512   Textfile
TURTLEGR.TEXT       8    7-Nov-84    46   512   Textfile
TURTLEGR.CODE       7    7-Nov-84    54   512   Codefile
TURT.ASM.TEXT       8    7-Nov-84    61   512   Textfile
TURT.ASM.CODE       5    7-Nov-84    69   512   Codefile
TURT.LIB.CODE       7    7-Nov-84    74   512   Codefile
< UNUSED >        431                81
6/6 files <listed/in dir>, 81 blocks used, 431 unused, 431 in largest
```

It consists of two parts, one (the interface) written in Pascal, the other written in assembly. All assembly routines (they will not work on the original Apple pascal) will be intercepted by the interpreter. You must C(ompile and A(ssemble the two source files and L(ink them together. The code files are included on this volumes, TURT.LIB.CODE is the final code of the library. You must X(ecute LIBRARY (on APPLE3:) to build a new library containing the new TURTLEGRAPHICS and all other segments from the old SYSTEM.LIBRARY.

DISK IMAGES

You can make your own set of system disk images (for example, called `work.svol` and `system.svol`). To create a 1024 block volume in a file *my.svol*, execute

```
ucsdpsys_mkfs -B 1024 my.svol
```

It is suggest that you put the following files on your volumes:

```
WORK:
SYSTEM.MISCINFO     1    7-Nov-84     6   192   Datafile
SYSTEM.LIBRARY     28    7-Nov-84     7   512   Datafile
< UNUSED >        989                35
2/2 files <listed/in dir>, 35 blocks used, 989 unused, 989 in largest
```

```
SYSTEM:
SYSTEM.PASCAL       44    3-Sep-85     6   512   Codefile
SYSTEM.FILER        30    3-Sep-85    50   512   Codefile
SYSTEM.EDITOR       50    3-Sep-85    80   512   Codefile
SYSTEM.SYNTAX       12    1-Jul-85   130   512   Textfile
SYSTEM.COMPILER     78    3-Sep-85   142   512   Codefile
SYSTEM.LINKER       25    3-Sep-85   220   512   Codefile
SYSTEM.ASSMBLER     50    3-Sep-85   245   512   Codefile
6502.OPCODES         2   20-Dec-78   295   208   Datafile
6502.ERRORS          7   25-Dec-83   297   498   Datafile
LIBRARY.CODE         8    3-Sep-85   304   512   Codefile
LIBMAP.CODE         11    3-Sep-85   312   512   Codefile
... etc
```

You will note, there is no `SYSTEM.APPLE` and no `SYSTEM.CHARSET` while `SYSTEM.PASCAL` is on the second volume. This virtual machine looks for `SYSTEM.PASCAL` on all volumes available, so it does not have to be located on the system volume. Only `SYSTEM.LIBRARY`, `SYSTEM.MISCINFO` and `SYSTEM.WRK.=` are stored on the system volume.

DEBUGGER COMMANDS

p Print evaluation stack

d *from to*
Dump memory in HEX and ASCII

l *SegNo ProcNo*
Disassemble P-code

t Dump call stack including local variable and arguments

v Dump local varable aera of the current procedure/function

g Go (start execution)

n Execute next instructon, do not trace subroutines.

f finish current procedure/function, execution stops at the first instruction after returning.

r Show registers

q Quit interpreter

SEE ALSO

ucsdpsys_svolio(1)
read p-System disk images

ucsdpsys_disk(1)
read and write p-System disk images

ucsdpsys_svolcvt(1)
convert Apple][Pascal disk images

ucsdpsys_charset(1)
encode and decode system.charset files.

Apple Pascal Operating System Reference Manual (1980)
Appendix A and B, Architecture and Operation of the P-machine

The UCSD P-System Museum
http://www.threedee.com/jcm/psystem/index.html

HISTORY

The first P-system (version I.3), written by was released by UCSD in august 1977. In 1980, Apple Computers released its Apple Pascal, based on UCSD Pascal version II.

WISH LIST

- more working APPLESTUFF
- native TRANCENDENT
- better volume handling (especially mounting/unmounting disks)
- REMIN: and REMOUT:
- resizing of CONSOLE:
- BREAK-handling
- removing (or hiding) the debug stuff
- Getting the 64KWord (instead of 64KByte) memory model working
- resizeing the TURTLEGRAPHICS screen
- adding more colors to TURTLEGRAPHICS
- better documentations (in the source code as well as in separate files)
- Porting the device drivers (mainly the console and turtlegraphics) to Windows and MacOS (could be obsolete with MacOS X)

BUGS

Lots of functions still missing, e.g. native functions of the Apple Pascal System (UNIT APPLESTUFF).

EXIT STATUS

The *ucsdpsys_vm* command will exit with a status of 1 on any error. The *ucsdpsys_vm* command will only exit with a status of 0 if there are no errors.

COPYRIGHT

ucsdpsys_vm version 0.7
Copyright © 2000, 2001 Mario Klebsch
Copyright © 2006, 2009, 2010 Peter Miller

The *ucsd-psystem-vm* programs come with ABSOLUTELY NO WARRANTY. This is free software and you are welcome to redistribute it under certain conditions. For details see the *ucsdpsys_vm_license*(1) man page.

AUTHOR

Mario Klebsch E-Mail: mario@klebsch.de
WWW: http://www.klebsch.de/

MAINTAINER

Peter Miller E-Mail: pmiller@opensource.org.au
/\/* WWW: http://miller.emu.id.au/pmiller/

GNU GENERAL PUBLIC LICENSE

Version 2, June 1991

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software -- to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation’s software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author’s protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors’ reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone’s free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

GNU GENERAL PUBLIC LICENSE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program’s source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients’ exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

Appendix: How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

> *one line to give the program’s name and a brief idea of what it does.*
> Copyright (C) 19yy *name of author*
>
> This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.
>
> This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.
>
> You should have received a copy of the GNU General Public License along with this program. If not, see <http://www.gnu.org/licenses/>.

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

> Gnomovision version 69, Copyright (C) 19yy name of author
> Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type ‘show w’. This is free software, and you are welcome to redistribute it under certain conditions; type ‘show c’ for details.

The hypothetical commands ‘show w’ and ‘show c’ should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than ‘show w’ and ‘show c’; they could even be mouse-clicks or menu items -- whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

> Yoyodyne, Inc., hereby disclaims all copyright interest in the program ‘Gnomovision’ (which makes passes at compilers) written by James Hacker.
>
> *signature of Ty Coon,* 1 April 1989
> Ty Coon, President of Vice

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Library General Public License instead of this License.

NAME

ucsdpsys_xturtleserver − X11 Turtle Graphics server

SYNOPSIS

ucsdpsys_xturtleserver
ucsdpsys_xturtleserver -V

DESCRIPTION

The *ucsdpsys_xturtleserver* program is used by the *ucsdpsys_vm*(1) program to draw graphics, via the specially crafted TURTLEGRAPHICS unit. This command is not usually invoked directly by the user.

OPTIONS

The following options are understood:

−V Print the version of the *ucsdpsys_xturtleserver* program being executed.

All other options will produce a diagnostic error.

EXIT STATUS

The *ucsdpsys_xturtleserver* command will exit with a status of 1 on any error. The *ucsdpsys_xturtleserver* command will only exit with a status of 0 if there are no errors.

COPYRIGHT

ucsdpsys_xturtleserver version 0.7

The *ucsd-psystem-vm* programs come with ABSOLUTELY NO WARRANTY. This is free software and you are welcome to redistribute it under certain conditions. For details see the *ucsdpsys_vm_license*(1) man page.

AUTHOR

Mario Klebsch	E-Mail:	mario@klebsch.de
	WWW:	http://www.klebsch.de/

MAINTAINER

Peter Miller	E-Mail:	pmiller@opensource.org.au
/\/*	WWW:	http://miller.emu.id.au/pmiller/