NAME

envwipe – run a program with a specified environment

SYNOPSIS

envwipe [**−e** *envfile*] [**−V**] *program arg0* [*argument(s) ...*]

DESCRIPTION

Introduction

The *envwipe* program is used on a UNIX system to run another program with a controlled environment. This is sometimes necessary in order to get a precise and repeatable instruction execution trace of the program. Precise instruction execution traces are useful for comparison with the execution of the same program on other machine platforms, like simulators. This program is essentially the same as the *execname* program that is commonly used for starting programs in multi-architecture computer environments. The difference is that no environment variables are passed to the target program by default.

Two arguments to the program are always required. These are the program to execute (its file path) and the first (or zeroth) argument to the program. This second argument forms part of the program environment so care should be used if it needs to be matched up with some specific value.

Options

There is only one important argument option for the program.

-e *envfile*   This option allows for the specification of a file that holds the environment that should be setup for the target program. The file should contains lines of the form:

```
key=value
```

These should look like environment variables being set by shell programs like Bourne Shell and Korn Shell. Comments and blank lines are allowed in this file. Comments are introduced using the pound ('#') character. Note that full shell-like quoting is not allowed for the value portion of the variable assignment. Quoting with the double quote character is allowed.

-V   This option causes the program to print its program version to standard error and then the program exits.

Operation

The default behavior of the program, when invoked with no optional arguments, is to execute the target program with no environment variables and only the arguments that are specified in addition to the zeroth argument to the program. If some optional environment is specified, then that environment is setup for the target program in addition to the arguments that are specified. The combination of the program arguments and its environment form the full user specified environment for the target program.

Environment File

An option file can be specified at program invocation so that some environment variables can be setup for the target program. By default, no environment variables are setup for the target program. If this file is given with the −e option, then any environment variables given in the file will be passed to the target program. The format of this file consists of key-value pairs. This is similar to how environment variables are assigned in the Bourne and Korn shell programs. Entries are of the form:

```
key=value
```

The value can be quoted with double quotes if it contains any white space. Blank lines in this file are ignored as well as all lines that are introduced with the pound ('#') character. Also, the value is taken to only be the first white-space terminated string after the equals-sign character.

EXAMPLES

☞   run the SpecInt program *compress* with its first argument being the string `compress`

```
envwipe /proj/levo/benchmarks/bin/compress compress
```

☞   run the SpecInt program *compress* with its first argument being the string `compress`
    ; its environment should be setup according to that which is found in the file `envfile` located in
the current directory

```
envwipe -e envfile /proj/levo/benchmarks/bin/compress compress
```

CAVEATS

This program is useful for executing a program with precise environment. This can be used to compare execution traces of the same program run on different platforms. The goal is to have the program traces be precisely identical. Note that other factors that cannot be controlled by this program can come into play such that the instruction execution traces are not the same. Care should be used that the stack area and heap are setup the same on all machines that are used for comparison purposes. Also, the exact arrangement of program environment data needs to be the same across platforms that are expected to be compared. Of course, this is not usually possible when running programs using a vendor supplied OS. It is left to the reader to figure out where remaining differences in OS setup for program execution occurs and to deal with it in some manner (if possible).

SEE ALSO

execv(1), execname(1), suexec(1), exec(1)

PATH TO

This program is currently located in
`/usr/add-on/local/bin` or possibly where ever *local* programs are stored on your system. This is often at
`"${LOCAL}/bin"` on some systems.

WARNINGS

Be careful to put at one space between all option key letters and the associated key letter parameter.

AUTHOR

Dave Morano