NAME

environ − user environment

SYNOPSIS

extern char *****environ*;**

DESCRIPTION

The variable *environ* points to an array of pointers to strings called the "environment". The last pointer in this array has the value NULL. (This variable must be declared in the user program, but is declared in the header file <*unistd.h*> in case the header files came from libc4 or libc5, and in case they came from glibc and **_GNU_SOURCE** was defined.) This array of strings is made available to the process by the **exec**(3) call that started the process.

By convention the strings in *environ* have the form "*name=value*". Common examples are:

USER The name of the logged-in user (used by some BSD-derived programs).

LOGNAME
The name of the logged-in user (used by some System-V derived programs).

HOME
A user's login directory, set by **login**(1) from the password file **passwd**(5).

LANG The name of a locale to use for locale categories when not overridden by **LC_ALL** or more specific environment variables like **LC_COLLATE**, **LC_CTYPE**, **LC_MESSAGES**, **LC_MONETARY**, **LC_NUMERIC**, **LC_TIME**, cf. **locale**(5).

PATH The sequence of directory prefixes that **sh**(1) and many other programs apply in searching for a file known by an incomplete pathname. The prefixes are separated by '**:**'. (Similarly one has **CDPATH** used by some shells to find the target of a change directory command, **MANPATH** used by **man**(1) to find manual pages, etc.)

PWD The current working directory. Set by some shells.

SHELL
The pathname of the user's login shell.

TERM The terminal type for which output is to be prepared.

PAGER
The user's preferred utility to display text files.

EDITOR/VISUAL
The user's preferred utility to edit text files.

Further names may be placed in the environment by the *export* command and "name=value" in **sh**(1), or by the *setenv* command if you use **csh**(1). Arguments may also be placed in the environment at the point of an **exec**(3). A C program can manipulate its environment using the functions **getenv**(3), **putenv**(3), **setenv**(3), and **unsetenv**(3).

Note that the behavior of many programs and library routines is influenced by the presence or value of certain environment variables. A random collection:

The variables **LANG**, **LANGUAGE**, **NLSPATH**, **LOCPATH**, **LC_ALL**, **LC_MESSAGES**, etc. influence locale handling, cf. **locale**(5).

TMPDIR influences the path prefix of names created by **tmpnam**(3) and other routines, the temporary directory used by **sort**(1) and other programs, etc.

LD_LIBRARY_PATH, **LD_PRELOAD** and other LD_* variables influence the behavior of the dynamic loader/linker.

POSIXLY_CORRECT makes certain programs and library routines follow the prescriptions of POSIX.

The behavior of **malloc**(3) is influenced by **MALLOC_*** variables.

The variable **HOSTALIASES** gives the name of a file containing aliases to be used with **gethostbyname**(3).

TZ and **TZDIR** give timezone information used by **tzset**(3) and through that by functions like **ctime**(3), **localtime**(3), **mktime**(3), **strftime**(3). See also **tzselect**(8).

TERMCAP gives information on how to address a given terminal (or gives the name of a file containing such information).

COLUMNS and **LINES** tell applications about the window size, possibly overriding the actual size.

PRINTER or **LPDEST** may specify the desired printer to use. See **lpr**(1).

Etc.

BUGS

Clearly there is a security risk here. Many a system command has been tricked into mischief by a user who specified unusual values for **IFS** or **LD_LIBRARY_PATH**.

There is also the risk of name space pollution. Programs like *make* and *autoconf* allow overriding of default utility names from the environment with similarly named variables in all caps. Thus one uses **CC** to select the desired C compiler (and similarly **MAKE**, **AR**, **AS**, **FC**, **LD**, **LEX**, **RM**, **YACC**, etc.). However, in some traditional uses such an environment variable gives options for the program instead of a pathname. Thus, one has **MORE**, **LESS**, and **GZIP**. Such usage is considered mistaken, and to be avoided in new programs. The authors of *gzip* should consider renaming their option to **GZIP_OPT**.

SEE ALSO

bash(1), **csh**(1), **login**(1), **sh**(1), **tcsh**(1), **execve**(2), **clearenv**(3), **exec**(3), **getenv**(3), **putenv**(3), **setenv**(3), **unsetenv**(3), **locale**(5)

COLOPHON

This page is part of release 3.22 of the Linux *man-pages* project. A description of the project, and information about reporting bugs, can be found at http://www.kernel.org/doc/man-pages/.