IEEE P1003.2a Draft 8 – December 1991

3.5.3 Variables

⇒ **3.5.3 Variables.** *Modify the contents of subclause 3.5.3, Variables, to add the following environment variables in the correct sorted order.*

ENV This variable, when the shell is invoked, shall be subjected to parameter expansion (see 3.6.2) by the shell and the resulting value shall be used as a pathname of a file containing shell commands to execute in the current environment. The file need not be executable. If the expanded value of **ENV** is not an absolute pathname, the results are unspecified. **ENV** shall be ignored if the user's real and effective user IDs or real and effective group IDs are different. This standard specifies the effects of this variable only for systems supporting the User Portability Utilities Option.

LINENO This variable shall be set by the shell to a decimal number representing the current sequential line number (numbered starting with 1) within a script or function before it executes each command. If the user unsets or resets **LINENO** , the variable may lose its special meaning for the life of the shell. If the shell is not currently executing a script or function, the value of **LINENO** is unspecified. This standard specifies the effects of this variable only for systems supporting the User Portability Utilities Option.

PPID This variable shall be set by the shell to the decimal process ID of the process that invoked this shell. In a subshell (see 3.12), **PPID** shall be set to the same value as that of parent of the current shell. For example, `echo $PPID` and `(echo $PPID)` would produce the same value. This standard specifies the effects of this variable only for systems supporting the User Portability Utilities Option.

PS 1 Each time an interactive shell is ready to read a command, the value of this variable shall be subjected to parameter expansion and written to standard error. The default value shall be `"$ "`. For users who have specific additional implementation-defined privileges (see 2.2.2.6) the default may be another, implementation-defined, value. The shell shall replace each instance of the character `!` in **PS 1** with the history file number of the next command to be typed. (See 5.12.) Escaping the `!` with another `!` (i.e., `!!`) shall place the literal character `!` in the prompt. This standard specifies the effects of this variable only for systems supporting the User Portability Utilities Option.

PS 2 Each time the user enters a <newline> prior to completing a command line in an interactive shell, the value of this variable shall be subjected to parameter expansion and written to stan-
dard error. The default value shall be `"> "`. This standard 7
specifies the effects of this variable only for systems supporting 7
the User Portability Utilities Option. 7

PS 4 When an execution trace (`set -x`) is being performed in an interactive shell, before each line in the execution trace, the value of this variable shall be subjected to parameter expansion and
written to standard error. The default value shall be `"+ "`. This 7
standard specifies the effects of this variable only for systems 7
supporting the User Portability Utilities Option. 7

Rationale. *(This subclause is not a part of P1003.2a)*

The following entries represent additional information about variables included in the standard, or rationale for common variables in use by shells that have been excluded.

_ (underscore)
While underscore is historic practice, its overloaded usage in the KornShell is confusing and it has been omitted from POSIX.2.

ENV This can be used to set aliases and other items local to the invocation of a shell. The file referred to by **ENV** differs from `$HOME/.profile` in that `.profile` is typically executed at session startup, whereas the **ENV** file is executed at the beginning of each shell invocation.

The **ENV** value is interpreted in a manner similar to a dot script, in that the commands are executed in the current environment and the file needs to be readable, but not executable. However, unlike dot scripts, no **PATH** searching is performed. This is used as a guard against Trojan Horse security breaches.

ERRNO Omitted from the standard because the values of POSIX.1 {8} error numbers are not defined in a portable manner.

FCEDIT In an earlier draft, **FCEDIT** was included in this subclause. However, since it affects only the `fc` utility, it has been removed from this more global place. The value of **FCEDIT** does not affect the
command-line editing mode in the shell; see the description of 8
`set -o vi` in 3.14.11. 8

LINENO This variable is frequently used in **PS 4** during trace mode (`set -x`).

PPID Without this variable, there is no way for a utility to signal its parent or to find its parent process. This is also useful to know if the shell has been ‘‘orphaned.’’

PS 1 Used for interactive prompts. Historically, the ‘‘super-user’’ has had a prompt of `"# "`. Since privileges are not required to be monolithic, it is difficult to define which privileges should cause the alternate prompt. However, a sufficiently powerful user should be reminded of that power by having an alternate prompt.

PS 2 Used for interactive prompts.

PS 3 Used by the KornShell for the `select` command. Since the POSIX. 2 shell does not support `select`, **PS 3** was omitted.

PS 4 Used for shell debugging. For example, the following script:

```
PS4='[${LINENO}]+ '
set -x
echo Hello
```

writes the following to standard error:

```
[3]+ echo Hello
```

RANDOM This pseudo-random number generator was not seen as being very useful to interactive users.

SECONDS Although this variable is sometimes used with **PS 1** to allow the display of the current time in the user's prompt, it is not one that would be manipulated frequently enough by an interactive user to include in POSIX.

3.12 Shell Execution Environment

⇒ **3.12 Shell Execution Environment.** *Add another item to the list of the contents of the shell execution environment:*

— Shell aliases (see 3.3.1).