IEEE P1003.2a Draft 8 – December 1991

The author of the original `emacs` program also expressed his desire to omit the program. Furthermore, there were a number of historical systems that did not include `emacs`, or included it without supporting it, but there were very few that
did not include and support `vi`. The shell `emacs` command-line editing mode was 8
finally omitted from this standard because it became apparent that the KornShell 8
version and the editor being distributed with the GNU system had diverged in 8
some respects. The author of `emacs` requested that the POSIX `emacs` mode either 8
be deleted or have a significant number of unspecified conditions. Although the 8
KornShell author agreed to consider changes to bring the shell into alignment, 8
the developers of the standard decided to defer specification at this time, rather 8
than attempting to agree on a specific subset of `emacs` late within balloting of the 8
standard. (Furthermore, the standard developers included few individuals who 8
had direct experience with `emacs` mode and the specification did not achieve the 8
level of review afforded to the `vi` mode.) It is assumed that the `emacs` and Korn- 8
Shell developers will converge on a definition acceptable to both groups and this 8
may be used as a model for a future version of this standard. In the interim, 8
implementations are free to offer additional command-line editing modes based on 8
the exact models of editors their users are most comfortable with. 8

4.70 `wait` — Await Process Completion

⇒ **4.70 `wait` Operands.** *In subclause 4.70, modify the description of the pid operand in the Operands subclause:*

pid One of the following:

(1) *<current contents>*

(2) On systems supporting both the POSIX.1 {8} job control option and the User Portability Utilities Option, a job control job ID (see 2.2.2.206) that identifies a background process group to be waited for. The job control job ID notation is applicable only for invocations of `wait` in the current shell execution environment; see 3.12. The exit status of `wait` shall be determined by the last command in the pipeline.

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

This additional syntax for *pid* is applicable only to systems having the job control features of the User Portability Extension. Thus, it is described in this supplement document, rather than in the base POSIX.2 document. See the rationale for `kill` for additional information on the effects of the job control job ID notation outside the current shell execution environment.

⇒ **4.70.10 `wait` Rationale.** *In subclause 4.70, add the following example to the rationale.*

Example: If the following sequence of commands is run in less than 31 seconds:

```
sleep 257 | sleep 31 &
jobs -l %%
```

either of the following commands will return the exit status of the second `sleep` in the pipeline:

```
wait <pid of sleep 31>
wait %%
```

Section 5: User Portability Utilities Option

This section describes the utilities that shall be implemented on all systems that claim conformance to the User Portability Utilities Option. Certain utilities are noted as having features that cannot be implemented on all terminal types; if the {POSIX2_CHAR_TERM} option is supported, the system shall support all such features on at least one terminal type. See 2.14.

Some of the utilities are only required on systems that also support the Software Development Utilities Option, the character-at-a-time terminal option (see 2.14) or that support the POSIX.1 {8} job control option; such utilities are so marked, directly before their Synopsis subclauses. All of the other utilities in this section are required only on systems that claim conformance to the User Portability Utilities Option.

User Portability Utilities Option Rationale. *(This subclause is not a part of P1003.2a)*.

All of the new utilities defined in the UPE are collected into a single large section. This was done for the practical reason that empty sections (chapters) were being held open for this work in the base POSIX.2 document. It was considered impractical to have enough empty chapters to handle all of the permutations of UPE, software development, and special terminal characteristics.

A summary of the combinations of options required for certain utilities in this section (where • indicates that the entire utility is optional unless the feature is provided and text indicates the portions that are optional):

Utility	{_POSIX2_UPE}	{_POSIX2_SW_DEV}	{_POSIX_JOB_CONTROL}	{_POSIX2_CHAR_TERM}
`bg`	•		•	
`ex`	•	`-t` option		see NOTE
`ctags`	•	•		
`fg`	•		•	
`more`	•	`-t` option		see NOTE
`nm`	•	•		
`sh`	•			see NOTE
`talk`	•			see NOTE
`vi`	•	`-t` option		•

NOTE: The `ex`, `more`, `sh`, and `talk` utilities also have terminal-related requirements that are allowed to result in error messages on some terminals (or all terminals, if {POSIX2_CHAR_TERM} is