# IEEE P1003.2a Draft 8 – December 1991

The `−w` option in previous drafts was changed to `−l` to match historical practice.

The `−N` option was included because without it, a noninteractive application cannot reject previously applied patches. For example, if a user is piping the output of `diff` into the `patch` utility, and the user only wants to patch a file to a newer version noninteractively, the `−N` option is required.

Changes to the `−l` option description were proposed to allow matching across 8
<newline>s in addition to just <blank>s. Since this is not historical practice, 8
and since some ambiguities could result, it is suggested that future developments 8
in this area be triggered by another option letter, such as `−L`. 8

## 5.23 ps — Report process status

### 5.23.1 Synopsis

```
ps [−aA][−G grouplist ][−o format ]... [−p proclist ][−t termlist ]      7
    [−U userlist ]
```

### 5.23.2 Description

The `ps` utility shall write information about processes, subject to having the appropriate privileges to obtain information about those processes.

When the `−o` option is not specified, information about processes selected shall be
written in an implementation-defined manner. By default, `ps` shall select all 7
processes with the same effective user ID as the current user and the same con- 7
trolling terminal as the invoker. In all cases, the selection of processes to be 7
reported shall be subject to the constraints of appropriate privileges. 7

### 5.23.3 Options

The `ps` utility shall conform to the utility argument syntax guidelines described in 2.10.2.

The following options shall be supported by the implementation.

`−a` Write information for all processes associated with terminals. 7
Implementations may omit session leaders from this list. 7

`−A` Write information for all processes.

`−G` *grouplist*
Write information for processes whose real group ID numbers are given in *grouplist* . The *grouplist* shall be a single argument in the form of a <blank>- or comma-separated list.

`−o` *format* Write information according to the format specification given in *format* . This is fully described in 5.23.6.1. Multiple `−o` options can be specified; the format specification shall be interpreted as the `<space>`-separated concatenation of all the *format* option-arguments.

`−p` *proclist* Write information for processes whose process ID numbers are given in *proclist* . The *proclist* shall be a single argument in the form of a `<blank>`- or comma-separated list.

`−t` *termlist* Write information for processes associated with terminals given in *termlist* . The *termlist* shall be a single argument in the form of a `<blank>`- or comma-separated list. Terminal identifiers shall be given in an implementation-defined format.

`−U` *userlist* Write information for processes whose real user ID numbers or login names are given in *userlist* . The *userlist* shall be a single argument in the form of a `<blank>`- or comma-separated list.

With the exception of `−o` *format* , all of the options shown are used to select 7
processes. If any are specified, the default list shall be ignored and `ps` shall select 7
the processes represented by the inclusive OR of all the selection-criteria options. 7

### 5.23.4 Operands

None.

### 5.23.5 External Influences

#### 5.23.5.1 Standard Input

None.

#### 5.23.5.2 Input Files

None.

#### 5.23.5.3 Environment Variables

The following environment variables shall affect the execution of `ps`:

**COLUMNS** This variable shall override the system-selected horizontal 7
screen size, used to determine the number of text columns 7
to display. See 2.6 for valid values and results when it is 7
unset or null. 7

**LANG** This variable shall determine the locale to use for the locale categories when both **LC_ALL** and the corresponding environment variable (beginning with **LC_**) do not specify a locale. See 2.6.

**LC_ALL** This variable shall determine the locale to be used to override any values for locale categories specified by the settings of **LANG** or any environment variables beginning with **LC_**.

**LC_CTYPE** This variable shall determine the locale for the interpretation of sequences of bytes of text data as characters (e.g., single- versus multibyte characters in arguments).

**LC_MESSAGES** This variable shall determine the language in which messages should be written.

**LC_TIME** This variable shall determine the format and contents of the date and time strings displayed.

#### 5.23.5.4 Asynchronous Events

Default.

### 5.23.6 External Effects

#### 5.23.6.1 Standard Output

When the `−o` option is not specified, the standard output format is implementation defined. The `−o` option allows the output format to be specified under user control.

The format specification shall be a list of names presented as a single argument, `<blank>`- or comma-separated. Each variable has a default header. The default header can be overridden by appending an equals sign and the new text of the header. The rest of the characters in the argument shall be used as the header text. The fields specified shall be written in the order specified on the command line, and should be arranged in columns in the output. The field widths shall be selected by the system to be at least as wide as the header text (default or overridden value). If the header text is null, such as `−o user=`, the field width shall be at least as wide as the default header text. If all header text fields are null, no header line shall be written.

The following names shall be recognized in the POSIX Locale: 8

`ruser` The real user ID of the process. This shall be the textual user ID, if it can be obtained and the field width permits, or a decimal representation otherwise.

`user` The effective user ID of the process. This shall be the textual user ID, if it can be obtained and the field width permits, or a decimal representation otherwise.

`rgroup` The real group ID of the process. This shall be the textual group ID, if it can be obtained and the field width permits, or a decimal representation otherwise.

`group` The effective group ID of the process. This shall be the textual group ID, if it can be obtained and the field width permits, or a decimal representation otherwise.

`pid` The decimal value of the process ID.

`ppid` The decimal value of the parent process ID.

`pgid` The decimal value of the process group ID.

`pcpu` The ratio of CPU time used recently to CPU time available in the 8
same period, expressed as a percentage. The meaning of 8
''recently'' in this context is unspecified. The CPU time available 8
is determined in an unspecified manner.

`vsz` The size of the process in (virtual) memory in kilobytes as a decimal integer.

`nice` The decimal value of the system scheduling priority of the process. See 5.20.

`etime` In the POSIX Locale, the elapsed time since the process was started, in the form:

[[*dd*`-`]*hh*`:`]*mm*`:`*ss*

where *dd* shall represent the number of days, *hh* the number of hours, *mm* the number of minutes, and *ss* the number of seconds. The *dd* field shall be a decimal integer. The *hh*, *mm*, and *ss* fields shall be two-digit decimal integers padded on the left with zeroes.

`time` In the POSIX Locale, the cumulative CPU time of the process in the form:

[*dd*`-`]*hh*`:`*mm*`:`*ss*

The *dd*, *hh*, *mm*, and *ss* fields shall be as described in the `etime` specifier.

`tty` The name of the controlling terminal of the process (if any) in the 7
same format used by the `who` utility. 7

`comm` The name of the command being executed (*argv*[0] value) as a string.

`args` The command with all its arguments as a string. The implementation may truncate this value to the field width; it is implementation defined whether any further truncation occurs. It is unspecified whether the string represented is a version of the argument list as it was passed to the command when it started, or is a version of the arguments as they may have been modified by the application. Applications cannot depend on being able to modify their argument list and having that modification be reflected in the output of `ps`.

Any field need not be meaningful in all implementations. In such a case a hyphen (`-`) should be output in place of the field value.

Only `comm` and `args` shall be allowed to contain `<blank>`s; all others shall not. Any implementation-defined variables shall specify in the conformance document if the field may contain `<blank>`s, as well as the default header.

Table 5-101 specifies the default header to be used in the POSIX Locale 7
corresponding to each format specifier. 7

**Table 5-101 – ps Variable Names and Default Headers**

| Format Specifier | Default Header | Format Specifier | Default Header |
|---|---|---|---|
| args | COMMAND | ppid | PPID |
| comm | COMMAND | rgroup | RGROUP |
| etime | ELAPSED | ruser | RUSER |
| group | GROUP | time | TIME |
| nice | NI | tty | TT |
| pcpu | %CPU | user | USER |
| pgid | PGID | vsz | VSZ |
| pid | PID | | |

#### 5.23.6.2 Standard Error

Used only for diagnostic messages.

#### 5.23.6.3 Output Files

None.

### 5.23.7 Extended Description

None.

### 5.23.8 Exit Status

The `ps` utility shall exit with one of the following values:

0 Successful completion.

>0 An error occurred.

### 5.23.9 Consequences of Errors

Default.

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

**Usage, Examples**

The command:

```
ps -o user,pid,ppid=MOM -o args
```

writes the following in the POSIX Locale. The contents of the COMMAND field need not be the same in all implementations, due to possible truncation.

```
  USER    PID    MOM    COMMAND
helene     34     12    ps -o uid,pid,ppid=MOM -o args
```

The args format specifier is allowed to produce a truncated version of the command arguments. In some implementations, this information is no longer available when the ps utility is executed.

If the field width is too narrow to display a textual ID, the system may use a numeric version. Normally, the system would be expected to choose large enough field widths, but if a large number of fields were selected to write, it might squeeze fields to their minimum sizes to fit on one line. One way to ensure adequate width for the textual IDs is to override the default header for a field to make it larger than most or all user or group names.

There is no special quoting mechanism for header text. The header text is the rest of the argument. If multiple header changes are needed, multiple −o options can be used, such as

```
ps -o "user=User Name" -o pid=Process\ ID
```

**History of Decisions Made**

There is very little commonality between BSD and System V implementations of ps. Many options conflict or have subtly different usages. The working group has attempted to select a set of options that seem useful on a wide range of systems and has selected options that either can be implemented on both BSD and System V based systems without breaking the current implementations or where the options are sufficiently similar that any changes would not be unduly painful for users or implementors.

It is recognized that on some systems, especially multilevel secure systems, ps may be nearly useless. The default output has therefore been chosen both such that it does not break existing implementations and also that it is likely to provide at least some useful information on most systems.

The major change is the addition of the format specification capability. The motivation for this invention is to provide a mechanism for users to access a wider range of system information, if the system permits it, in a portable manner. The fields chosen to appear in the specification were arrived at after considering what concepts were likely to be both reasonably useful to the ‘‘average’’ user and stood a reasonable chance of being implemented on a wide range of systems. Again it is recognized that not all systems will be able to provide all the information and, conversely, some may wish to provide more. It is hoped that the approach adopted will be sufficiently flexible and extensible to accommodate most systems. Implementations may be expected to introduce new format specifiers.

The default output should consist of a short listing containing the process ID, terminal name, cumulative execution time, and command name of each process.

The preference of the group would have been to make the format specification an operand of the `ps` command. Unfortunately, BSD usage precluded this.

At one time a format was included to display the process's environment array. This was deleted because there is no portable way to display it.

The `−o nice` field is named from historical precedent. An implementation may provide an extension to name it something based on ‘‘system scheduling priority.’’

The historical `−l` option has been omitted because it could not be specified more precisely. Since the default output is ‘‘written in an implementation-defined manner,’’ having a longer version of it required is not useful in a standard. However, mentioning `−l` here in the rationale serves to reserve its meaning and implementors are encouraged to provide it as an extension.

The historical options `−g` and `−u` have had different meanings on various systems. 7
They were changed to their uppercase equivalents to achieve a common definition in POSIX.2.

The `−A` option is equivalent to the BSD `−g` and the *SVID* `−e`. Because the two sys- 7
tems differed, a mnemonic compromise was selected. 7

The `−a` option is described with some optional behavior because the *SVID* omits 7
session leaders, but BSD does not. 7

In an earlier draft, format specifiers appeared for priority and start time. The former was not defined adequately in POSIX.2 and was removed in deference to the defined system scheduling priority; the latter because elapsed time was considered to be more useful.

The following categories of information were omitted because they were too implementation-specific:

| | |
|---|---|
| `STATE` | Process state (`R` for ‘‘Run’’,etc.) |
| `FLAGS` | Octal scheduling flags. |
| `ADDR` | Process address. |
| `WCHAN` | Wakeup channel or event being slept on. |

In a new BSD version of `ps`, a `−O` option can be used to write all of the default information followed by additional format specifiers. This was not adopted by POSIX.2 because the default output is implementation-defined. Nevertheless, this is a useful option that should be reserved for that purpose. In the `−o` option for the POSIX.2 `ps`, the format is the concatenation of each `−o`. Therefore, the user can have an alias or function that defines the beginning of her desired format and add more fields to the end of the output in certain cases where that would be useful.

The format of the terminal name is unspecified, but the descriptions of `ps`, `talk`, 7
`who`, and `write` require that they all use the same format. 7

The `pcpu` field indicates that the CPU time available is determined in an 8
unspecified manner. This is because it is difficult to express an algorithm that is 8
useful across all possible machine architectures. Historical counterparts to this 8
value have attempted to show percentage of use in the recent past, such as the 8
preceding minute. Frequently, these values for all processed did not add up to 8
100 percent. Implementations are encouraged to provide data in this field to 8
users that will help them identify processes currently affecting the performance of 8
the system. 8

## 5.24 `renice` — Set system scheduling priorities of running processes

### 5.24.1 Synopsis

`renice` [`−n` *increment* ] [ `−g` | `−p` | `−u` ] *ID* . . .

*Obsolescent Versions* :

`renice` *nice_value* [`−p`] *pid* . . . [`−g` *gid* . . . ] [`−p` *pid* . . . ] [`−u` *user* . . . ]

`renice` *nice_value* `−g` *gid* . . . [`−g` *gid* . . . ] [`−p` *pid* . . . ] [`−u` *user* . . . ]

`renice` *nice_value* `−u` *user* . . . [`−g` *gid* . . . ] [`−p` *pid* . . . ] [`−u` *user* . . . ]

### 5.24.2 Description

The `renice` utility shall request that the system scheduling priorities (see 2.2.2.212) of one or more running processes be changed. When a process ID is `renice`d (the default), the applicable processes are specified by their process IDs. When a process group is `renice`d, the request applies to all processes in the process group.

The system scheduling priority shall be bounded in an implementation-defined manner. If the requested *increment* (or *nice_value* in the obsolescent versions) would raise or lower the system scheduling priority of the executed utility beyond implementation-defined limits, then the limit whose value was exceeded shall be used.

When a user is `renice`d and {_POSIX_SAVED_IDS} (see 2.9.3 in POSIX.1 {8}) is defined, the request applies to all processes whose saved set-user-ID matches the user ID corresponding to the user. When a user is `renice`d and {_POSIX_SAVED_IDS} is not defined, the request applies to all processes whose real user ID matches the user ID corresponding to the user.

Regardless of which options are supplied or any other factor, `renice` shall not alter the system scheduling priorities of any process unless the user requesting such a change has appropriate privileges to do so for the specified process. If the user lacks appropriate privileges to perform the requested action, the utility shall return an error status.

On systems where {_POSIX_SAVED_IDS} is defined, the saved set-user-ID of the user's process shall be checked instead of its effective user ID when `renice` attempts to determine the user ID of the process in order to determine whether the user has appropriate privileges. On systems where {_POSIX_SAVED_IDS} is not defined, `renice` shall check the real or effective user ID when attempting to determine the user ID of the process and the user ID of the requester as part of the determination of whether the user has appropriate privileges to perform the requested action.