Full credit is given to the above companies including the OS that this PDF file was generated!

Red Hat Enterprise Linux Release 9.2 Manual Pages on 'time.1p' command

$ man time.1p

TIME(1P) POSIX Programmer's Manual TIME(1P)

PROLOG

This manual page is part of the POSIX Programmer's Manual. The Linux implementation of this interface may differ (consult the corresponding Linux manual page for details of Linux behavior), or the interface may not be implemented on Linux.

NAME

time ? time a simple command

SYNOPSIS

time [-p] utility [argument...]

DESCRIPTION

The time utility shall invoke the utility named by the utility operand with arguments supplied as the argument operands and write a message to standard error that lists timing statistics for the utility. The mes? sage shall include the following information:

* The elapsed (real) time between invocation of utility and its ter? mination.
* The User CPU time, equivalent to the sum of the tms_utime and tms_cutime fields returned by the times() function defined in the System Interfaces volume of POSIX.1?2017 for the process in which utility is executed.
* The System CPU time, equivalent to the sum of the tms_stime and tms_cstime fields returned by the times() function for the process

in which utility is executed.

The precision of the timing shall be no less than the granularity de? fined for the size of the clock tick unit on the system, but the re? sults shall be reported in terms of standard time units (for example, 0.02 seconds, 00:00:00.02, 1m33.75s, 365.21 seconds), not numbers of clock ticks.

When time is used as part of a pipeline, the times reported are unspec? ified, except when it is the sole command within a grouping command (see Section 2.9.4.1, Grouping Commands) in that pipeline. For example, the commands on the left are unspecified; those on the right report on utilities a and c, respectively:

```
time a | b | c    { time a; } | b | c
a | b | time c    a | b | (time c)
```

OPTIONS

The time utility shall conform to the Base Definitions volume of POSIX.1?2017, Section 12.2, Utility Syntax Guidelines.

The following option shall be supported:

-p Write the timing output to standard error in the format shown in the STDERR section.

OPERANDS

The following operands shall be supported:

utility The name of a utility that is to be invoked. If the utility operand names any of the special built-in utilities in Sec? tion 2.14, Special Built-In Utilities, the results are unde? fined.

argument Any string to be supplied as an argument when invoking the utility named by the utility operand.

STDIN

Not used.

INPUT FILES

None.

ENVIRONMENT VARIABLES

The following environment variables shall affect the execution of time:

LANG Provide a default value for the internationalization vari?
ables that are unset or null. (See the Base Definitions vol?
ume of POSIX.1?2017, Section 8.2, Internationalization Vari?
ables for the precedence of internationalization variables
used to determine the values of locale categories.)

LC_ALL If set to a non-empty string value, override the values of
all the other internationalization variables.

LC_CTYPE Determine the locale for the interpretation of sequences of
bytes of text data as characters (for example, single-byte as
opposed to multi-byte characters in arguments).

LC_MESSAGES
Determine the locale that should be used to affect the format
and contents of diagnostic and informative messages written
to standard error.

LC_NUMERIC
Determine the locale for numeric formatting.

NLSPATH Determine the location of message catalogs for the processing
of LC_MESSAGES.

PATH Determine the search path that shall be used to locate the
utility to be invoked; see the Base Definitions volume of
POSIX.1?2017, Chapter 8, Environment Variables.

ASYNCHRONOUS EVENTS

Default.

STDOUT

Not used.

STDERR

If the utility utility is invoked, the standard error shall be used to
write the timing statistics and may be used to write a diagnostic mes?
sage if the utility terminates abnormally; otherwise, the standard er?
ror shall be used to write diagnostic messages and may also be used to
write the timing statistics.

If -p is specified, the following format shall be used for the timing
statistics in the POSIX locale:

"real %f\nuser %f\nsys %f\n", <real seconds>, <user seconds>,
<system seconds>

where each floating-point number shall be expressed in seconds. The precision used may be less than the default six digits of %f, but shall be sufficiently precise to accommodate the size of the clock tick on the system (for example, if there were 60 clock ticks per second, at least two digits shall follow the radix character). The number of dig? its following the radix character shall be no less than one, even if this always results in a trailing zero. The implementation may append white space and additional information following the format shown here. The implementation may also prepend a single empty line before the for? mat shown here.

OUTPUT FILES

None.

EXTENDED DESCRIPTION

None.

EXIT STATUS

If the utility utility is invoked, the exit status of time shall be the exit status of utility; otherwise, the time utility shall exit with one of the following values:

1?125 An error occurred in the time utility.

126 The utility specified by utility was found but could not be in? voked.

127 The utility specified by utility could not be found.

CONSEQUENCES OF ERRORS

Default.

The following sections are informative.

APPLICATION USAGE

The command, env, nice, nohup, time, and xargs utilities have been specified to use exit code 127 if an error occurs so that applications can distinguish ``failure to find a utility'' from ``invoked utility exited with an error indication''. The value 127 was chosen because it is not commonly used for other meanings; most utilities use small val?

ues for ``normal error conditions'' and the values above 128 can be confused with termination due to receipt of a signal. The value 126 was chosen in a similar manner to indicate that the utility could be found, but not invoked. Some scripts produce meaningful error messages differ? entiating the 126 and 127 cases. The distinction between exit codes 126 and 127 is based on KornShell practice that uses 127 when all attempts to exec the utility fail with [ENOENT], and uses 126 when any attempt to exec the utility fails for any other reason.

EXAMPLES

It is frequently desirable to apply time to pipelines or lists of com? mands. This can be done by placing pipelines and command lists in a single file; this file can then be invoked as a utility, and the time applies to everything in the file.

Alternatively, the following command can be used to apply time to a complex command:

```
time sh -c 'complex-command-line'
```

RATIONALE

When the time utility was originally proposed to be included in the ISO POSIX?2:1993 standard, questions were raised about its suitability for inclusion on the grounds that it was not useful for conforming ap? plications, specifically:

* The underlying CPU definitions from the System Interfaces volume of POSIX.1?2017 are vague, so the numeric output could not be compared accurately between systems or even between invocations.
* The creation of portable benchmark programs was outside the scope this volume of POSIX.1?2017.

However, time does fit in the scope of user portability. Human judge? ment can be applied to the analysis of the output, and it could be very useful in hands-on debugging of applications or in providing subjective measures of system performance. Hence it has been included in this vol? ume of POSIX.1?2017.

The default output format has been left unspecified because historical implementations differ greatly in their style of depicting this numeric

output. The -p option was invented to provide scripts with a common means of obtaining this information.

In the KornShell, time is a shell reserved word that can be used to time an entire pipeline, rather than just a simple command. The POSIX definition has been worded to allow this implementation. Consideration was given to invalidating this approach because of the historical model from the C shell and System V shell. However, since the System V time utility historically has not produced accurate results in pipeline tim? ing (because the constituent processes are not all owned by the same parent process, as allowed by POSIX), it did not seem worthwhile to break historical KornShell usage.

The term utility is used, rather than command, to highlight the fact that shell compound commands, pipelines, special built-ins, and so on, cannot be used directly. However, utility includes user application programs and shell scripts, not just the standard utilities.

FUTURE DIRECTIONS

None.

SEE ALSO

Chapter 2, Shell Command Language, sh

The Base Definitions volume of POSIX.1?2017, Chapter 8, Environment Variables, Section 12.2, Utility Syntax Guidelines

The System Interfaces volume of POSIX.1?2017, times()

COPYRIGHT

most likely to have been introduced during the conversion of the source files to man page format. To report such errors, see https://www.kernel.org/doc/man-pages/reporting_bugs.html .