



*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 'puts.3p' command***

***\$ man puts.3p***

PUTS(3P)                    POSIX Programmer's Manual                    PUTS(3P)

### 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

puts ? put a string on standard output

### SYNOPSIS

```
#include <stdio.h>

int puts(const char *s);
```

### DESCRIPTION

The functionality described on this reference page is aligned with the ISO C standard. Any conflict between the requirements described here and the ISO C standard is unintentional. This volume of POSIX.1?2017 defers to the ISO C standard.

The puts() function shall write the string pointed to by s, followed by a <newline>, to the standard output stream stdout. The terminating null byte shall not be written.

The last data modification and last file status change timestamps of the file shall be marked for update between the successful execution of puts() and the next successful completion of a call to fflush() or fclose() on the same stream or a call to exit() or abort().

## RETURN VALUE

Upon successful completion, puts() shall return a non-negative number.

Otherwise, it shall return EOF, shall set an error indicator for the stream, and errno shall be set to indicate the error.

## ERRORS

Refer to fputc().

The following sections are informative.

## EXAMPLES

### Printing to Standard Output

The following example gets the current time, converts it to a string using localtime() and asctime(), and prints it to standard output using puts(). It then prints the number of minutes to an event for which it is waiting.

```
#include <time.h>
#include <stdio.h>
...
time_t now;
int minutes_to_event;
...
time(&now);
printf("The time is ");
puts(asctime(localtime(&now)));
printf("There are %d minutes to the event.\n",
      minutes_to_event);
...
```

## APPLICATION USAGE

The puts() function appends a <newline>, while fputs() does not.

This volume of POSIX.1?2017 requires that successful completion simply return a non-negative integer. There are at least three known different implementation conventions for this requirement:

- \* Return a constant value.
- \* Return the last character written.
- \* Return the number of bytes written. Note that this implementation

convention cannot be adhered to for strings longer than `{INT_MAX}` bytes as the value would not be representable in the return type of the function. For backwards compatibility, implementations can return the number of bytes for strings of up to `{INT_MAX}` bytes, and return `{INT_MAX}` for all longer strings.

#### RATIONALE

None.

#### FUTURE DIRECTIONS

None.

#### SEE ALSO

Section 2.5, Standard I/O Streams, `fopen()`, `fputs()`, `putc()`

The Base Definitions volume of POSIX.1-2017, `<stdio.h>`

#### COPYRIGHT

Portions of this text are reprinted and reproduced in electronic form from IEEE Std 1003.1-2017, Standard for Information Technology -- Portable Operating System Interface (POSIX), The Open Group Base Specifications Issue 7, 2018 Edition, Copyright (C) 2018 by the Institute of Electrical and Electronics Engineers, Inc and The Open Group. In the event of any discrepancy between this version and the original IEEE and The Open Group Standard, the original IEEE and The Open Group Standard is the referee document. The original Standard can be obtained online at <http://www.opengroup.org/unix/online.html>.

Any typographical or formatting errors that appear in this page are 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](https://www.kernel.org/doc/man-pages/reporting_bugs.html).

IEEE/The Open Group

2017

PUTS(3P)