

**NAME**

cat – concatenate and display files

**SYNOPSIS**

**cat** [-nbsuvet] [*file*...]

**DESCRIPTION**

The **cat** utility reads each *file* in sequence and writes it on the standard output. Thus:

example% **cat file**

prints *file* on your terminal, and:

example% **cat file1 file2 >file3**

concatenates *file1* and *file2*, and writes the results in *file3*. If no input file is given, **cat** reads from the standard input file.

**OPTIONS**

The following options are supported:

- n**        Precede each line output with its line number.
  
- b**        Number the lines, as **-n**, but omit the line numbers from blank lines.
  
- u**        The output is not buffered. (The default is buffered output.)
  
- s**        **cat** is silent about non-existent files.
  
- v**        Non-printing characters (with the exception of tabs, new-lines and form-feeds) are printed visibly. ASCII control characters (octal 000 – 037) are printed as  $\hat{n}$ , where *n* is the corresponding ASCII character in the range octal 100 – 137 (@, A, B, C, . . . , X, Y, Z, [, \, ], ^, and \_); the **DEL** character (octal 0177) is printed  $\hat{?}$ . Other non-printable characters are printed as **M-x**, where *x* is the ASCII character specified by the low-order seven bits.

When used with the **-v** option, the following options may be used:

- e**        A \$ character will be printed at the end of each line (prior to the new-line).
  
- t**        Tabs will be printed as  $\hat{T}$ 's and formfeeds to be printed as  $\hat{L}$ 's.

The **-e** and **-t** options are ignored if the **-v** option is not specified.

**OPERANDS**

The following operand is supported:

*file*                    A path name of an input file. If no *file* is specified, the standard input is used. If *file* is '-', **cat** will read from the standard input at that point in the sequence. **cat** will not close and reopen standard input when it is referenced in this way, but will accept multiple occurrences of '-' as *file*.

#### USAGE

See **largefile(5)** for the description of the behavior of **cat** when encountering files greater than or equal to 2 Gbyte (  $2^{31}$  bytes).

#### EXAMPLES

##### Example 1: Concatenating a file

The following command:

```
example% cat myfile
```

writes the contents of the file **myfile** to standard output.

##### Example 2: Concatenating two files into one

The following command:

```
example% cat doc1 doc2 > doc.all
```

concatenates the files **doc1** and **doc2** and writes the result to **doc.all**.

##### Example 3: Concatenating two arbitrary pieces of input with a single invocation

The command:

```
example% cat start - middle - end > file
```

when standard input is a terminal, gets two arbitrary pieces of input from the terminal with a single invocation of **cat**. Note, however, that if standard input is a regular file, this would be equivalent to the command:

```
cat start - middle /dev/null end > file
```

because the entire contents of the file would be consumed by **cat** the first time '-' was used as a *file* operand and an end-of-file condition would be detected immediately when '-' was referenced the second time.

#### ENVIRONMENT VARIABLES

See **environ(5)** for descriptions of the following environment variables that affect the execution of **cat**: **LANG**, **LC\_ALL**, **LC\_CTYPE**, **LC\_MESSAGES**, and **NLSPATH**.

#### EXIT STATUS

The following exit values are returned:

**0**                    All input files were output successfully.

**>0**                  An error occurred.

#### ATTRIBUTES

See **attributes(5)** for descriptions of the following attributes:

```
tab() allbox; cw(2.750000i)| cw(2.750000i) lw(2.750000i)| lw(2.750000i). ATTRIBUTE
TYPEATTRIBUTE VALUE AvailabilitySUNWcsu CSIenabled Interface StabilityStandard
```

**SEE ALSO**

**touch(1), attributes(5), environ(5), largefile(5), standards(5)**

**NOTES**

Redirecting the output of **cat** onto one of the files being read will cause the loss of the data originally in the file being read. For example,

```
example% cat filename1 filename2 >filename1
causes the original data in filename1 to be lost.
```