

NAME

stdio — standard buffered input/output package

SYNOPSIS

```
#include <stdio.h>
```

```
FILE *stdin;
```

```
FILE *stdout;
```

```
FILE *stderr;
```

DESCRIPTION

The functions described under subheading 3S constitute an efficient user-level buffering scheme. The in-line macros *getc*(3S) and *putc*(3S) handle characters quickly. The higher level routines *gets*, *fgets*, *scanf*, *fscanf*, *fread*, *puts*, *fputs*, *printf*, *sprintf*, *fwrite* all use *getc* and *putc*; they can be freely intermixed.

A file with associated buffering is called a *stream*, and is declared to be a pointer to a defined type **FILE**. *Fopen*(3S) creates certain descriptive data for a stream and returns a pointer to designate the stream in all further transactions. There are three normally open streams with constant pointers declared in the include file and associated with the standard open files:

stdin standard input file

stdout standard output file

stderr standard error file

In addition, there is a constant 'pointer' **NULL** (0) that designates the null stream.

An integer constant **EOF** (-1) is returned upon end of file or error by most integer functions that deal with streams.

Any routine using *occ* that uses the standard input/output package must include the header file of pertinent macro definitions this way:

```
#include <stdio.h>
```

and must be loaded with a special library, obtained this way:

```
cc ... -lS
```

(Note that *occ* will convert any invocation of **-IS** to **-lS** automatically.) The functions and constants mentioned in subheading 3S are appropriately declared in the include file, and need no further declaration. The following 'functions' are implemented as macros; redeclaration of these names is perilous: *getc*, *getchar*, *putc*, *putchar*, *feof*, *ferror*, *fileno*.

SEE ALSO

open(2), *close*(2), *read*(2), *write*(2)

DIAGNOSTICS

Invalid *stream* pointers will usually cause grave disorder, possibly including program termination. See individual function descriptions for possible error conditions.

Typical error conditions to watch for are a **FILE** pointer which has not been initialized with *fopen*, input (output) being attempted on an output (input) stream, or a **FILE** pointer which designates corrupt or otherwise unintelligible **FILE** data.