function

perror

<cstdio>

```
void perror ( const char * str );
```

Print error message

Interprets the value of errno as an error message, and prints it to stderr (the standard error output stream, usually the console), optionally preceding it with the custom message specified in *str*.

errno is an integral variable whose value describes the error condition or diagnostic information produced by a call to a library function (any function of the C standard library may set a value for errno, even if not explicitly specified in this reference, and even if no error happened), see errno for more info.

The error message produced by `perror` is platform-depend.

If the parameter *str* is not a null pointer, *str* is printed followed by a colon (:) and a space. Then, whether *str* was a null pointer or not, the generated error description is printed followed by a newline character (`'\n'`).

`perror` should be called right after the error was produced, otherwise it can be overwritten by calls to other functions.

Parameters.

`str`

C string containing a custom message to be printed before the error message itself.
If it is a null pointer, no preceding custom message is printed, but the error message is still printed.
By convention, the name of the application itself is generally used as parameter.

Return Value

none

Example

```
1  /* perror example */
2  #include <stdio.h>
3
4  int main ()
5  {
6    FILE * pFile;
7    pFile=fopen ("unexist.ent","rb");
8    if (pFile==NULL)
9      perror ("The following error occurred");
10   else
11     fclose (pFile);
12   return 0;
13 }
```

If the file `unexist.ent` does not exist, something similar to this may be expected as program output:

```
The following error occurred: No such file or directory
```

See also

clearerr	Clear error indicators (function)
ferror	Check error indicator (function)

tmpnam
ungetc
vfprintf
vfscanf
vprintf
vscanf
vsnprintf
vsprintf
vsscanf

objects:

stderr
stdin
stdout

types:

FILE
fpos_t
size_t

macro constants:

BUFSIZ
EOF
FILENAME_MAX
FOPEN_MAX
L_tmpnam
NULL
TMP_MAX