

NAME

elftc_demangle - demangle a C++ name

LIBRARY

library "libelftc"

SYNOPSIS

```
#include <libelftc.h>
```

int

```
elftc_demangle(const char *encodedname, char *buffer, size_t bufsize, unsigned int flags);
```

DESCRIPTION

Function **elftc_demangle**() decodes a symbol name encoded according to the type encoding rules for the C++ language and returns a string denoting an equivalent C++ prototype.

Argument *encodedname* specifies the encoded symbol name. Argument *buffer* denotes a programmer-specified area to place the prototype string in. Argument *bufsize* specifies the size of the programmer-specified area. Argument *flags* specifies the encoding style in use for argument *encodedname*.

Supported encoding styles are:

ELFTC_DEM_ARM The encoding style used by compilers adhering to the conventions of the C++ Annotated Reference Manual.

ELFTC_DEM_GNU2 The encoding style by GNU C++ version 2.

ELFTC_DEM_GNU3 The encoding style by GNU C++ version 3 and later.

Argument *flags* may be zero, in which case the function will attempt to guess the encoding scheme from the contents of *encodedname*.

RETURN VALUES

Function **elftc_demangle**() returns 0 on success. In case of an error it returns -1 and sets the *errno* variable.

EXAMPLES

To decode a name that uses an unknown encoding style use:

```
char buffer[1024];  
const char *funcname;
```

```
funcname = ...; /* points to string to be demangled */
if (elftc_demangle(funcname, buffer, sizeof(buffer), 0) == 0)
    printf("Demangled name: %\n", buffer);
else
    perror("Cannot demangle %s", funcname);
```

ERRORS

Function **elftc_demangle()** may fail with the following errors:

[EINVAL] Argument *encodedname* was not a valid encoded name.

[ENAMETOOLONG] The output buffer specified by arguments *buffer* and *bufsize* was too small to hold the decoded function prototype.

SEE ALSO

elf(3), elf_strptr(3)