

NAME

re_comp, **re_exec** - regular expression handler

LIBRARY

Compatibility Library (libcompat, -lcompat)

SYNOPSIS

```
#include <unistd.h>
```

```
char *
```

```
re_comp(const char *s);
```

```
int
```

```
re_exec(const char *s);
```

DESCRIPTION

This interface is made obsolete by `regex(3)`.

The **re_comp()** function compiles a string into an internal form suitable for pattern matching. The **re_exec()** function checks the argument string against the last string passed to **re_comp()**.

The **re_comp()** function returns 0 if the string *s* was compiled successfully; otherwise a string containing an error message is returned. If **re_comp()** is passed 0 or a null string, it returns without changing the currently compiled regular expression.

The **re_exec()** function returns 1 if the string *s* matches the last compiled regular expression, 0 if the string *s* failed to match the last compiled regular expression, and -1 if the compiled regular expression was invalid (indicating an internal error).

The strings passed to both **re_comp()** and **re_exec()** may have trailing or embedded newline characters; they are terminated by NULs. The regular expressions recognized are described in the manual entry for `ed(1)`, given the above difference.

DIAGNOSTICS

The **re_exec()** function returns -1 for an internal error.

The **re_comp()** function returns "no previous regular expression" or one of the strings generated by `regerror(3)`.

SEE ALSO

ed(1), egrep(1), ex(1), fgrep(1), grep(1), regex(3)

HISTORY

The `re_comp()` and `re_exec()` functions appeared in 4.0BSD.