NAME

e_stdio - error generating versions of standard I/O routines

SYNOPSIS

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

char * e_cuserid(s, inhflag)
    char *s;
int  e_fclose(stream, inhflag)
    FILE *stream;
FILE * e_fdopen(fildes, type, inhflag)
    int fildes;
    char *type;
int  e_fflush(stream, inhflag)
    FILE *stream;
int  e_fgetc(stream, inhflag)
    FILE *stream;
char * e_fgets(s, n, stream, inhflag)
    char *s;
    int n;
    FILE *stream;
FILE * e_fopen(filename, type, inhflag)
    char *filename;
    char *type;
int  e_fprintf(stream, format, inhflag, arg1, arg2, ... arg10)
    FILE *stream;
    char *format;
    int arg1;
    /*  etc. -- NB only 10 args are effective */
int  e_fputc(c, stream, inhflag)
    int c;
    FILE *stream;
int  e_fputs(s, stream, inhflag)
    int s;
    FILE *stream;
int  e_fputw(w, stream, inhflag)
    int w;
    FILE *stream;
int  e_fread(ptr, siz, nitems, stream, inhflag)
    char *ptr;
    int siz;
    int nitems;
    FILE *stream;
FILE * e_freopen(filename, type, stream, inhflag)
    char *filename;
    char *type;
    FILE *stream;
int  e_fscanf(stream, format, inhflag, ptr1, ptr2, ... ptr10)
    FILE *stream;
    char *format;
    int *ptr1;
    /*  etc. -- NB only 10 args are effective */
```

```
int  e_fseek(stream, offset, ptrname, inhflag)
    FILE *stream;
    long offset;
    int ptrname;
long  e_ftell(stream, inhflag)
    FILE *stream;
int  e_fwrite(ptr, siz, nitems, stream, inhflag)
    char *ptr;
    int siz;
    int nitems;
    FILE *stream;
int  e_getc(stream, inhflag)
    FILE *stream;
int  e_getchar( inhflag)
char * e_gets(s, inhflag)
    char *s;
int  e_getw(stream, inhflag)
    FILE *stream;
int  e_pclose(stream, inhflag)
    FILE *stream;
FILE * e_popen(command, type, inhflag)
    char *command;
    char *type;
int  e_printf(format, inhflag, arg1, arg2, ... arg10)
    char *format;
    int arg1;
    /*  etc. -- NB only 10 args are effective */
int  e_putc(c, stream, inhflag)
    int c;
    FILE *stream;
int  e_putchar(c, inhflag)
    int c;
int  e_puts(s, inhflag)
    int s;
int  e_rewind(stream, inhflag)
    FILE *stream;
int  e_scanf(format, inhflag, ptr1, ptr2, ... ptr10)
    char *format;
    int *ptr1;
    /*  etc. -- NB only 10 args are effective */
int  e_setbuf(stream, buf, inhflag)
    FILE *stream;
    char *buf;
int  e_system(string, inhflag)
    char *string;
int  e_ungetc(c, stream, inhflag)
    int c;
    FILE *stream;
```

DESCRIPTION

These routines are analagous to those described in e_syscall(3L). The corresponding standard I/O routine - Section 3S - is called and its return value is in turn returned (like that?). See e_syscall(3L) for details of how errors are processed.

Note that no extra "name" arguments are required as in some e_syscall(3L) routines. Instead the file name passed to e_open or e_reopen is remembered and stdin, stdout and stderr are special cased. If a file is opened be other means or if you don't like the saved name, it may be changed by calling e_savename(3L).

LIBRARY

/lib/lib1.a

SEE ALSO

e_syscall(3L) and the sections referenced there
e_savename(3L)

DIAGNOSTICS

Same as corresponding routines in Section 3S.

BUGS

Attempts to use e_fprintf (and possibly a few others) with a stream opened for reading will not be detected since the standard I/O routine leaves no trace of such an error.

E_printf, e_fprintf, e_scanf and e_fscanf are limited to 10 arguments besides the format, stream, and innflag arguments.