

NAME

fexecve() - executes a program

SYNOPSIS

fexecve (name, in, out, err, argptr, envptr)

char *name;

FILE *in, *out, *err;

char **argptr, **envptr;

DESCRIPTION

fexecve() causes the current process to execute a new program. The text, data, and stack segments of the process are discarded and replaced with the text and data sections of the new object file and a new stack. The argument list and environment list are copied into the new address space where they become the arguments to the entry point of the new program.

name is a string containing the name of the object file to be executed. If the name contains a slash (/), it is assumed to be a pathname to the file. If there is no slash, fexecve() searches for the file in a list of directories contained in the environment variable PATH, or if there is no such variable, it searches the working directory first, then /bin. The PATH variable (which is taken from the new environment list envptr) has a value which is a series of directory names separated by colons. The working directory is represented in this list by omitting a name (before the first colon, between two colons, or after the last colon). Thus the default search rules may be expressed as

PATH=:/bin

in, out, and err are files which are to be substituted for stdin, stdout, and stderr, respectively, when the new program is executed.

argptr and envptr are pointers to arrays of pointers to strings, with a NULL pointer as the last element of the array. By convention, argptr[0] is the name of the program.

RETURNS

Returns a standard system error code; fexecve() does not return if it succeeds, as the program that calls it is no longer in this process's memory.

NOTES

If a program needs to run another program without destroying itself, it can use fork(), some variation of fexecve(), and cwait(). These three functions are combined in spawnl() and spawnv().