

idea to subtract values returned by `ftell` to see how far apart two file positions are.

rewind The `rewind` function sets the file position at the beginning. The call `rewind(fp)` is nearly equivalent to `fseek(fp, 0L, SEEK_SET)`. The difference? `rewind` doesn't return a value but does clear the error indicator for `fp`.

fgetpos **fseek** and **ftell** have one problem: they're limited to files whose positions can be stored in a long integer. For working with very large files, C provides two additional functions: `fgetpos` and `fsetpos`. These functions can handle large files because they use values of type `fpos_t` to represent file positions. An `fpos_t` value isn't necessarily an integer: it could be a structure, for instance.

Q&A

The call `fgetpos(fp, &file_pos)` stores the file position associated with `fp` in the `file_pos` variable. The call `fsetpos(fp, &file_pos)` sets the file position for `fp` to be the value stored in `file_pos`. (This value must have been obtained by a previous call of `fgetpos`.) If a call of `fgetpos` or `fsetpos` fails, it stores an error code in `errno`. Both functions return zero when they succeed and a nonzero value when they fail.

Here's how we might use `fgetpos` and `fsetpos` to save a file position and return to it later:

```
fpos_t file_pos;
...
fgetpos(fp, &file_pos);    /* saves current position */
...
fsetpos(fp, &file_pos);    /* returns to old position */
```

PROGRAM Modifying a File of Part Records

The following program opens a binary file containing part structures, reads the structures into an array, sets the `on_hand` member of each structure to 0, and then writes the structures back to the file. Note that the program opens the file in `"rb+"` mode, allowing both reading and writing.

```
invclear.c /* Modifies a file of part records by setting the quantity
              on hand to zero for all records */

#include <stdio.h>
#include <stdlib.h>

#define NAME_LEN 25
#define MAX_PARTS 100

struct part {
    int number;
    char name[NAME_LEN+1];
    int on_hand;
} inventory[MAX_PARTS];
```