Note: The following example, taken from pages 75–76 of the text by Haviland et al., shows how the `ftw` system call is used. For each file/directory encountered during the recursive tree walk, the program prints the name and the permission bits. If the name represents a directory or a symbolic link, an asterisk ('*') is also printed.

```
/* A simple example to illustrate the use of ftw system call. */

#include  <stdio.h>
#include  <sys/types.h>
#include  <sys/stat.h>
#include  <fcntl.h>
#include  <unistd.h>
#include  <ftw.h>

/* The following mask will be used to print only the last nine */
/* permission bits for each file.                              */

#define  MODE_MASK    0777

int  main(int argc, char *argv[]) {

  /* Prototype for the function to be used with ftw. */
  int list (const char *, const struct stat *, int);

  /* If no command line arguments are specified, then the      */
  /* recursive tree walk starts at the current working directory. */
  /* For simplicity, the depth parameter is chosen as 1.       */

  if (argc == 1) {
    ftw(".", list, 1);
  }
  else {
    /* Otherwise, the recursive tree walk begins with the */
    /* directory specified by argv[1].                    */
    ftw(argv[1], list, 1);
  }

  return 0;
} /* End of main. */
```

```c
int list (const char *name, const struct stat *status, int type) {

  /* This function is passed to ftw by main. */

  /* If the call to stat failed, then just return. */

  if (type == FTW_NS)
     return  0;

  /* Otherwise, print the name of the file, permissions and the '*' suffix */
  /* if the file is a directory or a symbolic link.                        */

  /* The "-30" parameter used in printf call ensures that the name is   */
  /* left justified in a field of length 30 characters. The "3o" format */
  /* prints the permissions in octal.                                   */

  if (type == FTW_F)
     printf("%-30s\t0%3o\n", name, status->st_mode & MODE_MASK);
  else
     printf("%-30s*\t0%3o\n", name, status->st_mode & MODE_MASK);

  return 0;

} /* End of list. */
```