myfind.c works by taking two arguments, path to search in and string to search for, and then it prints out what it is searching for and where, then prints out how many occurances of the string it found by using %d and setting it to the return value of the recursive_search function, which takes parameters path, string, and count (which is passed as 0 initially).

The recursive_search function opens the directory it was given and checks every directory (besides “.” and “..”) to see if it contains the char* str which was passed to the function. If it contains it then the count parameter which was initially 0 is incremented, and the print_finfo(fpath, finfo) function is called. print_finfo simply uses the lstat finfo variable it was passed and prints out all the necessary information, then frees the memory allocated by the fpath parameter it was passed before returning. Now, back in recursive_search after print_finfo returns, if the current entry is a directory then the count variable is set equal to another recursive call of the recursive_search function:

count = recursive_search(fpath, str, count);

at the end of each recursive call of this function, the directory is closed and the count variable is returned, so that by the last recursive call the total count has been returned to main in order to be printed, all the directories that had to be opened were closed, and all the memory set aside by malloc for the

fpath variable has been freed after printing the file info in print_finfo.