Linux deep tree file renamer

Language: `bash`

Please use either command line or script format to satisfy this problem.

Notable state:

```
$ pwd
/webroot

$ df -k
Filesystem     1K-blocks     Used Available Use% Mounted on
/dev/xvda1       8123812  1599332   6424232  20% /
/dev/xvda2      24371436 19427724   4843464  80% /webroot
...
```

Summary

- From the web root, rename all files that end in *$path*`.htm` or *$path*`.html` to *$path*`.en.htm` and *$path*`.en.html`, respectively.
- Do not rename files that are already in this format.
- Try to ensure that there is never a time of interim state. Either all original files are present or all updated files are present. There is never a time when some file updates have been completed (such that the original no longer exists) and some have not.
- Filenames can contain arbitrary characters. Any character that can be present in a filename might be present somewhere beneath this tree. Some of these filenames may have been created based on user input from a web form, meaning that anything goes.
- The solution should be as portable as possible.
- Explain your methods and your assumptions, including assumptions that might impact portability.

Scenario

You operate a web server for a large and technically diverse company, and there is a massive localization effort underway. Under the web root, there are hundreds of thousands of pages. Some of them have well-constructed filenames, and some do not. Some were created by people who do not understand “web” technologies, perhaps exported to HTML via Microsoft Word. Many filenames are riddled with special characters, spaces, parentheses, brackets, etc.. You suspect that some filenames may even be based on internet user input from web forms.

Your web server supports a URL rewriting feature, such that, when configured, all requests for pages that end in `.html` (but not `.en.html`) will instead look for files with the extension `.en.html`, and requests for pages that end in `.htm` (but not `.en.htm`) will instead look for files with extension `.en.htm`. Later, there will be an effort to adjust links accordingly.

Examples:

```
http://mrcompany.com/team%20bob/api%20-%20BobRun().htm
    -> $WEBROOT/team bob/api - BobRun().en.htm

http://mrcompany.com/test%20wiki/using%20<stdio_wrapper.h>.html
    -> $WEBROOT/test wiki/using <stdio_wrapper.h>.en.html
```

Without downtime of the web server or any web pages, you would like to prepare the file tree for this.

Order of operations: 1. configure the filesystem such that whether the browser uses `open("/webroot/some/file/data.html")` or `open("/webroot/some/file/data.en.html")`, the result is identical. 2. configure the web server software to rewrite the URLs and open the `.en.html` file instead of the original file. 3. expunge the original files, as they are no longer necessary.

Your tasks are parts **#1** and **#3**. **#2** is out of scope for you.

Send the results by e-mail, in the form of a text file or shell script (packaging the file(s) in a compressed archive, e.g. zip or tar.gz, is also okay).