Workshop: Smart Shell, Part IV

Dr. Mirela Damian
Assigned: February 11, 2014
Due: February 20, 2014

Description

In this programming workshop you will **finalize** the Smart Shell you created so far. You may work alone or with one other classmate, and each person is expected to contribute equally to the effort. Note: if needed, a complete working version of the second Smart Shell project will be provided as a starting point.

Project Specification

- Make a copy of your smartshell3.c file, and name it "mysmartshell.c"
- Using good program design practices, implement the following:

1. "copy src_filename dest_filename" – copies the source file to the destination file
2. "delete filename" - deletes the file named "filename" (moves it to the trash)
3. "trash" - lists any recently deleted files that your smart shell knows about
4. "undelete filename" - restores the deleted file named "filename" if it exists
5. "empty" - removes any files from the trash

Implementation Suggestions

1. Create a directory named "trash" in your smartshell directory (use mkdir). This will serve as a repository of all files you will be deleting. For the purpose of this project you may assume that all deleted files have unique names (meaning that you will never remove a file with the same name from different locations).
2. In the trash directory, maintain a text file – say, trash.txt – that contains a listing of the original location of each file in the trash.
3. In your code maintain a structure – say, trash_files – that stores the contents of trash.txt. This is similar to your history_cmd structure, which stores the contents of history.txt.
4. Implement functions similar to ones with prototypes described below:

```
void handle_copy(char * line);
/* copy source file into destination file (both embedded in line) */

void add_to_trash(char * filepath);
/* add the filepath to the trash_files array (similar to  save_history_cmd);
   the trash_files structure should be initialized in the startup function, and
   should be saved into trash.txt just before quitting the shell.*/
```

```
void handle_delete(char * filepath);
/* add the filepath to the trash_files array, then move the file physically into the trash
   directory (invoke mv using the system command). Note that the original location of
   the file would be lost at this point if we did not save it into the trash_files array. */

void handle_undelete(char * filename);
/* search for the name of the file into the trash_files array (use strstr) to extract the
   original location of the file, then move the file physically from the trash into its
   original location */

void handle_trash();
/* print out each string from the trash_files array (similar to handle_history) */

void handle_empty();
/* remove each file from the trash directory, including the trash.txt file */
```

You will also need to add code to the shutdown method (provided in your solution to the smartshell III assignment) to save and then free the entries of the trash_files structure (which have been dynamically allocated).

Readme

Create a text file named "readme4" (not "readme.txt", or "README", or "Readme", etc.) that contains:

- Your name and the workshop number.
- A description of whatever help (if any) you received from others outside of class.
- An indication of how much time you spent doing the assignment outside of class.
- Your assessment of the workshop: Did it help you to learn? What did it help you to learn? Do you have any suggestions for improvement? Etc.
- Any information that will help us to grade your work in the most favorable light. In particular you should describe all known bugs.

Descriptions of your code should not be in the readme file. Instead they should be integrated into your code as comments.

Your readme file should be a plain text file. Don't create your readme file using Microsoft Word or any other word processor.

What to Hand in

Hand in printed copies of the following:

- Your readme file
- Your C source code
- Sample output showing how each of the commands implemented in this part of the smartshell project works

Leave the readme file and your C source code in your account. Do not make any changes to these files after the deadline for the assignment.

Grading

50	Total points possible
10	Program compiles successfully and runs
15	Each command listed does something resembling the specified action
15	Each command works correctly and completely
5	You have added some extra "bells and whistles", made output "prettier", or otherwise demonstrated excellence in crafting your smart shell
5	Your readme file is comprehensive