Design Choice:

Compares to make the safe removed file hidden, using environment variable TRASH, and put all the safe removed file together in one folder can speed things up. For example, if we want to lsrm/durm/trash all the files or directories that were safe removed, then we need to loop through entire file system to be able to find all the hidden files and directories, not to say some of the files and directories are intentionally hidden but are safe removed. So in that case not only will make the process slow, but also make the process no longer accurate by counting/listing/removing more files and directories that are supposed to be counted/listed/removed.

For the files and/or directories already safely removed, and the srm on them again will make the files and/or directories are deleted. Since user already “deletes” them once, the second time might really want to remove them and recover the disk space. So srm will delete all the files and/or directories that are srm-ed twice.

Also for those non-fetal errors user might make, will still proceed the command, but with error message given. So hopefully can help user to learn how to correctly use the command, and still do the command’s job.

Why not unrm \*

Because the directory where we run the script is not necessary the same directory where we store all the safe removed files and/or directories. So if user want to unrm all the safe removed files and/or directories they should run unrm “$TRASH”/\*. By just running unrm \*, means user want to restore all the files and/or directories in current directory. Which does not make sense, if user want to unrm something, that means they already were moved to trashcan which means they are not in currently directory. So unrm \* will do nothing, since the file and/or directory in current directory is not yet srm-ed.

Command: srm

Description:

Safe delete files by moving all files to a “trashcan” folder (where system should have an environment variable “TRASH” pre-defined, or the command will fail instantly) The “TRASH” folder may or may not exist already, if not, the first time user using srm, will automatically create one for user based on the value of TRASH. Note, if "srm" is invoked with a directory, then the entire directory hierarchy should be safe-rm’d.

Usage:

Under: when the variable TRASH already defined

1. Without option: if user run the scripts only with a list of files, srm will simply move all the valid files to the “TRASH” folder, unless they are already in the “TRASH” folder, then srm will delete the files and/or directories for user. If some path that user provide are invalid, srm will prompt with error message “files and/or directories cannot locate”

2) With option: if user decide to run the script with options, then srm will call command “rm” and append those additional options that user passed in, and remove valid files and/or directories as user desire. If some path that user provide are invalid, srm will prompt with error message “files and/or directories cannot locate”

Command: unrm

Description:

Un-removes files and/or directories if it hasn't been trashed (still in trashcan folder) into user’s current directory. The “TRASH” folder has to be existed already, if not, the process will exist immediately. Simply because if there is no trashcan folder, then nothing is safe-removed nor the user is ready to use this whole safe-delete feature. Note, since srm is not remembering the paths where files and/or directories came from, so unrm can only recover files and/or directories back to user’s current directory.

Usage:

Under: when the variable TRASH already defined and trashcan folder already is built, or will exit immediately.

Unrm does not take any options, unrm only take file and/or directory names, if there are options passed in, will just take it as invalid file and/or directories names that are not in the trash can. If there are invalid names, unrm will print error messages: file/directory does not exist in trash can.

Command: lsrm

Description:

"ls -l" on all files and directories that are currently in the trash

Usage:

Under: when the variable TRASH already defined and trashcan folder already is built, or will exit immediately.

Lsrm does not take any arguments. If there is, lsrm will print error messages, but still ls -l the trash can.

Command: durm

Description:

Rans "du" on the trashcan to let user know how much disk space would be recovered by running "trash"

Usage:

Under: when the variable TRASH already defined and trashcan folder already is built, or will exit immediately.

Durm does not take any arguments. If there is, durm will print error messages, but still du -s the trash can.

Command: trash

Description:

Removes all files that are currently in trash can folder, recovering the disk space.

Usage:

Under: when the variable TRASH already defined and trashcan folder already is built, or will exit immediately.

Trash does not take any arguments. If there is, trash will print error messages, but still removes all the files and/or directories in the trash can.