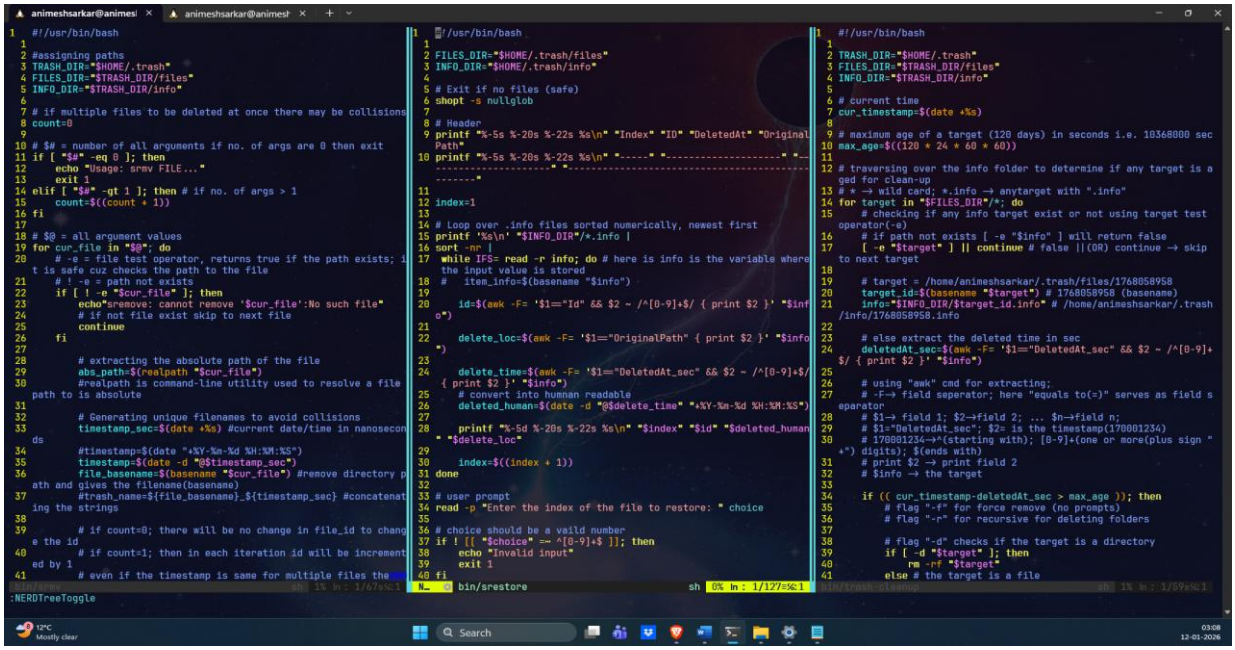
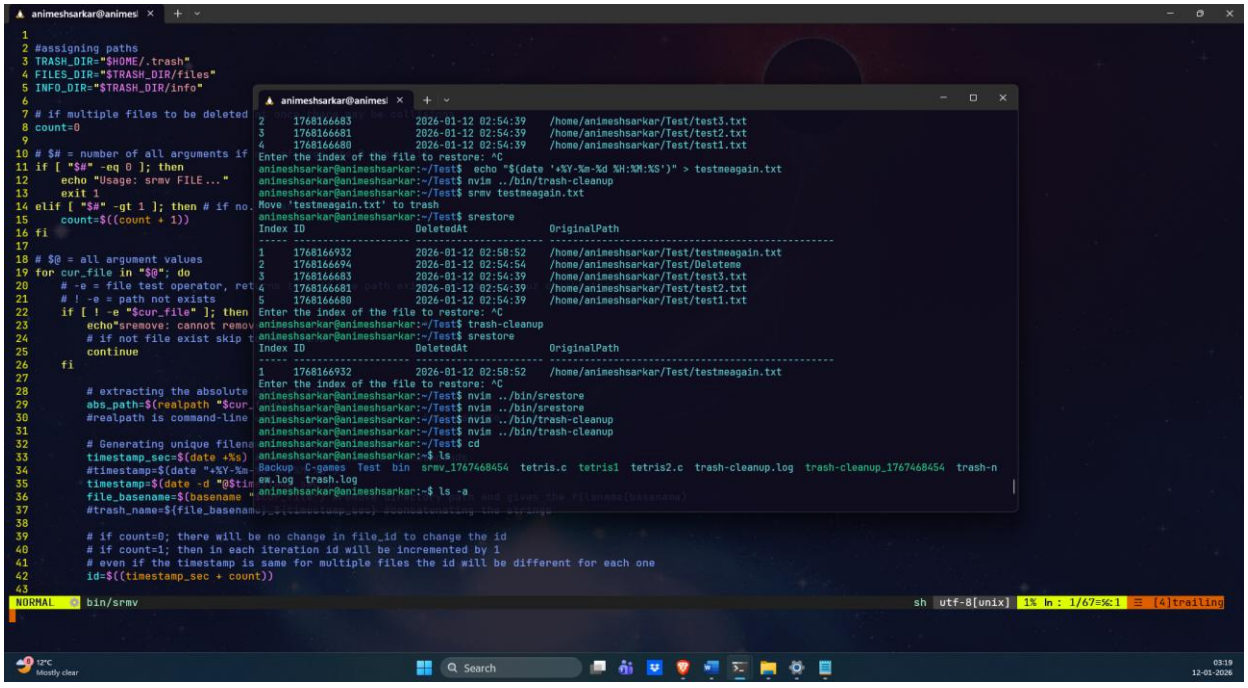
Creating a Custom Secure Linux Remove Command with a Trash Bin





presented by

```
animeshsarkar@animeshsarkar:~$ whoami
animeshsarkar
animeshsarkar@animeshsarkar:~$
```

High-level design

1. Create a trash directory ~/.trash/
2. ├── files/ # actual deleted files
3. └── info/ # metadata (original path, deletion time)
4. Create a **trash-cleanup** cmd
5. Demonstration of the trash cleanup script
6. Create a restore cmd (**srestore**)
7. Replace rm with our script (**srmv**)
8. Use a create a cron job for automatic trash clean-up

---

1. Create trash directories

```
animeshsarkar@animeshsarkar:~$ mkdir -p ~/.trash/files ~/.trash/info
```

Here, ***-p*** represents ***"parents"***: creates parent directories if not exist; if they do exist it doesn't give errors.

```
animeshsarkar@animeshsarkar:~/.trash$ ls -l
total 8
drwxr-xr-x 2 animeshsarkar animeshsarkar 4096 Jan  2 00:34 files
drwxr-xr-x 2 animeshsarkar animeshsarkar 4096 Jan  2 00:34 info
animeshsarkar@animeshsarkar:~/.trash$
```

2. Ensuring that ***~/bin*** is in present in ***$PATH***
   - Run ***echo $PATH*** to print the value of ***PATH*** variable. Check for ***"/home/username/bin" or "/Users/username/bin".*** Since mine is set, so I have no reason to worry for now.

     ```
     animeshsarkar@animeshsarkar:~/.trash$ echo $PATH
     /home/animeshsarkar/bin:/home/animeshsarkar/bin:/
     ```

   - If yours not, here's what you can do. Check your shell by running ***"echo $SHELL"*** don't worry you can run these commands from any location since they are environment variable.

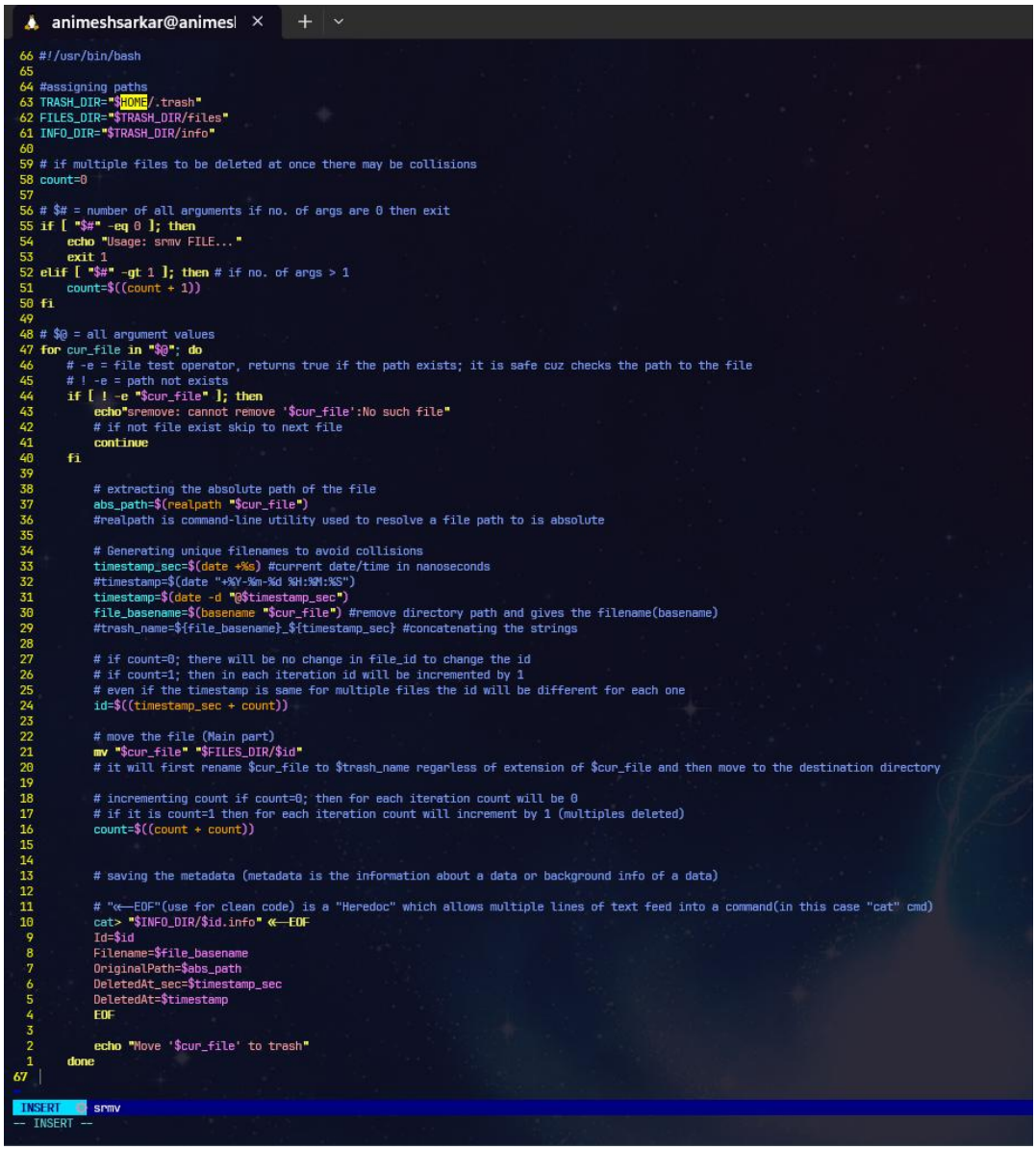
     ```
     animeshsarkar@animeshsarkar:~/.trash$ echo $SHELL
     /bin/bash
     animeshsarkar@animeshsarkar:~/.trash$
     ```

   - Mine is ***bash*** so I need to edit ***~/.bashrc*** file and add this at the bottom:
     export PATH="$HOME/bin: $PATH"
   - Apply changes: ***source ~/.bashrc***
   - If the ***bin*** folder is not created create one ***mkdir -p ~/bin***
   - Create a test cmd: ***echo -e '#!/bin/bash\necho "bin is working fine"' > /bin/testcmd***
   - Give executable permission: ***chmod +x ~/bin/testcmd***
   - Run cmd: ***testcmd.*** Enjoy.

3. Writing ***srmv*** cmd script
    - Creating the file

```
animeshsarkar@animeshsarkar:~/bin$ nvim srmv
```

    - The script



    - Giving executable permission: ***chmod a+x srmv*** (giving everyone executable permission)

```
animeshsarkar@animeshsarkar:~/bin$ ls -l srmv
-rw-r--r-- 1 animeshsarkar animeshsarkar 1539 Jan  2 00:57 srmv
animeshsarkar@animeshsarkar:~/bin$ chmod a+x srmv
animeshsarkar@animeshsarkar:~/bin$ ls -l srmv
-rwxr-xr-x 1 animeshsarkar animeshsarkar 1539 Jan  2 00:57 srmv
animeshsarkar@animeshsarkar:~/bin$
```

4. Clean-up script

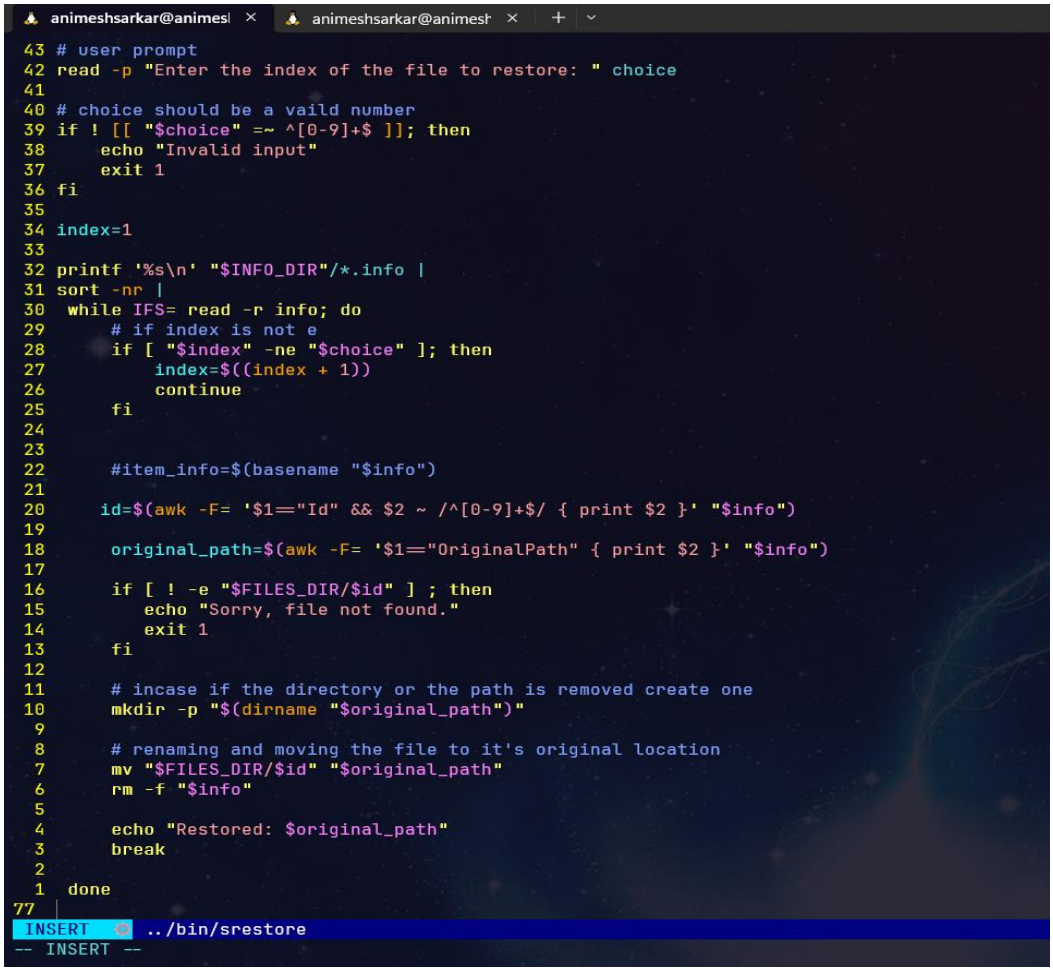
```
animeshsarkar@animesh  ×  animeshsarkar@animesh  ×  +  ~

FILES_DIR="$TRASH_DIR/files"
INFO_DIR="$TRASH_DIR/info"

# current time
cur_timestamp=$(date +%s)

# maximum age of a target (120 days) in seconds i.e. 10368000 sec
# max_age=$((120 * 24 * 60 * 60))
max_age=$(( 2 * 60 )) # for testing current it's 2 mins

# traversing over the info folder to determine if any target is aged for clean-up
# * → wild card; *.info → anytarget with ".info"
for target in "$FILES_DIR"/*; do
    # checking if any info target exist or not using target test operator(-e)
    # if path not exists [ -e "$info" ] will return false
    [ -e "$target" ] || continue # false ||(OR) continue → skip to next target

    # target = /home/animeshsarkar/.trash/files/1768058958
    target_id=$(basename "$target") # 1768058958 (basename)
    info="$INFO_DIR/$target_id.info" # /home/animeshsarkar/.trash/info/1768058958.info

    # else extract the deleted time in sec
    deletedAt_sec=$(awk -F= '$1=="DeletedAt_sec" && $2 ~ /^[0-9]+$/ { print $2 }' "$info")

    # using "awk" cmd for extracting;
    # -F→ field seperator; here "equals to(=)" serves as field separator
    # $1→ field 1; $2→field 2; ... $n→field n;
    # $1="DeletedAt_sec"; $2= is the timestamp(170001234)
    # 170001234→^(starting with); [0-9]+(one or more(plus sign "+") digits); $(ends with)
    # print $2 → print field 2
    # $info → the target

    if (( cur_timestamp-deletedAt_sec > max_age )); then
        # flag "-f" for force remove (no prompts)
        # flag "-r" for recursive for deleting folders

        # flag "-d" checks if the target is a directory
        if [ -d "$target" ]; then
            rm -rf "$target"
        else # the target is a file
            rm -f "$target"
        fi

        rm -f "$info"
    fi

done


 INSERT   ../bin/trash-cleanup
-- INSERT --
```

5. Create an interactive restore cmd (***srestore***)

- Show all the trash file info with **index**

```
animeshsarkar@animesh  ×  animeshsarkar@animesh  ×  +  ~

#!/usr/bin/bash

FILES_DIR="$HOME/.trash/files"
INFO_DIR="$HOME/.trash/info"

# Exit if no files (safe)
shopt -s nullglob

# Header
printf "%-5s %-20s %-22s %s\n" "Index" "ID" "DeletedAt" "OriginalPath"
printf "%-5s %-20s %-22s %s\n" "-----" "--------------------" "----------------------" "----------------------------------------------"

index=1

# Loop over .info files sorted numerically, newest first
printf '%s\n' "$INFO_DIR"/*.info |
sort -nr |
 while IFS= read -r info; do # here is info is the variable where the input value is stored
 #   item_info=$(basename "$info")

    id=$(awk -F= '$1=="Id" && $2 ~ /^[0-9]+$/ { print $2 }' "$info")

    delete_loc=$(awk -F= '$1=="OriginalPath" { print $2 }' "$info")

    delete_time=$(awk -F= '$1=="DeletedAt_sec" && $2 ~ /^[0-9]+$/ { print $2 }' "$info")
    # convert into humnan readable
    deleted_human=$(date -d "@$delete_time" "+%Y-%m-%d %H:%M:%S")

    printf "%-5d %-20s %-22s %s\n" "$index" "$id" "$deleted_human" "$delete_loc"

    index=$((index + 1))
done
```

- Taking user prompt to restore the file via index number



```
# user prompt
read -p "Enter the index of the file to restore: " choice

# choice should be a vaild number
if ! [[ "$choice" =~ ^[0-9]+$ ]]; then
    echo "Invalid input"
    exit 1
fi

index=1

printf '%s\n' "$INFO_DIR"/*.info |
sort -nr |
 while IFS= read -r info; do
     # if index is not e
     if [ "$index" -ne "$choice" ]; then
         index=$((index + 1))
         continue
     fi


     #item_info=$(basename "$info")

    id=$(awk -F= '$1=="Id" && $2 ~ /^[0-9]+$/ { print $2 }' "$info")

     original_path=$(awk -F= '$1=="OriginalPath" { print $2 }' "$info")

     if [ ! -e "$FILES_DIR/$id" ] ; then
        echo "Sorry, file not found."
        exit 1
     fi

     # incase if the directory or the path is removed create one
     mkdir -p "$(dirname "$original_path")"

     # renaming and moving the file to it's original location
     mv "$FILES_DIR/$id" "$original_path"
     rm -f "$info"

     echo "Restored: $original_path"
     break

 done
```

```
INSERT  ../bin/srestore
-- INSERT --
```

- Here is tip the moment your code works. Save it in some other folder as the quick backup. I overwrote the **srestore** cmd but luckily I had the backup file.

```
animeshsarkar@animeshsarkar:~/bin$ cp ../Backup/Scripts/srestore srestore
animeshsarkar@animeshsarkar:~/bin$ nvim srestore
```

6. Final Demonstration
   - Creating test files and directories

```
animeshsarkar@animeshsarkar:~/Test$ mkdir -p DeleteMe
animeshsarkar@animeshsarkar:~/Test$ nvim DeleteMe/test1.txt
animeshsarkar@animeshsarkar:~/Test$ nvim test2.c
animeshsarkar@animeshsarkar:~/Test$ ls
DeleteMe  qweoiqjot.txt  test2.c
animeshsarkar@animeshsarkar:~/Test$ ls DeleteMe/
test1.txt
animeshsarkar@animeshsarkar:~/Test$
```

   - Using our custom ***srmv*** cmd

```
animeshsarkar@animeshsarkar:~/Test$ srmv DeleteMe test2.c
Move 'DeleteMe' to trash
Move 'test2.c' to trash
animeshsarkar@animeshsarkar:~/Test$
```

- Verifying if the items moved to .**trash**

```
animeshsarkar@animeshsarkar:~/Test$ ls ../.trash/files/
1768152296  1768153430  1768162314  1768164764  1768164766
animeshsarkar@animeshsarkar:~/Test$ ls ../.trash/info
1768152296.info  1768153430.info  1768162314.info  1768164764.info  1768164766.info
animeshsarkar@animeshsarkar:~/Test$
```

- Restore file and directory. **Data is intact.**

```
animeshsarkar@animeshsarkar:~/Test$ srestore
Index ID                  DeletedAt             OriginalPath
----- ------------------- --------------------- ------------------------------------------------
1     1768165254          2026-01-12 02:30:52   /home/animeshsarkar/Test/test2.c
2     1768165253          2026-01-12 02:30:52   /home/animeshsarkar/Test/DeleteMe
3     1768165112          2026-01-12 02:28:32   /home/animeshsarkar/bin/trash-list
4                         2026-01-11 23:13:50   /home/animeshsarkar/Test/nfsioadsgfsg.txt
5                         2026-01-11 22:54:56   /home/animeshsarkar/Test/nfsioasg.txt
Enter the index of the file to restore: 1
Restored: /home/animeshsarkar/Test/test2.c
animeshsarkar@animeshsarkar:~/Test$ cat test2.c
int main() {
        return 0;
}
animeshsarkar@animeshsarkar:~/Test$ srestore
Index ID                  DeletedAt             OriginalPath
----- ------------------- --------------------- ------------------------------------------------
1     1768165253          2026-01-12 02:30:52   /home/animeshsarkar/Test/DeleteMe
2     1768165112          2026-01-12 02:28:32   /home/animeshsarkar/bin/trash-list
3                         2026-01-11 23:13:50   /home/animeshsarkar/Test/nfsioadsgfsg.txt
4                         2026-01-11 22:54:56   /home/animeshsarkar/Test/nfsioasg.txt
Enter the index of the file to restore: 1
Restored: /home/animeshsarkar/Test/DeleteMe
animeshsarkar@animeshsarkar:~/Test$ ls
DeleteMe  qweoiqjot.txt  test2.c
animeshsarkar@animeshsarkar:~/Test$ cd DeleteMe/
animeshsarkar@animeshsarkar:~/Test/DeleteMe$ ls
test1.txt
animeshsarkar@animeshsarkar:~/Test/DeleteMe$
```

- Creating and removing a new file whose max age is less than 2 min

```
animeshsarkar@animeshsarkar:~/Test$  echo "$(date '+%Y-%m-%d %H:%M:%S')" > testmeagain.txt
animeshsarkar@animeshsarkar:~/Test$
```

```
animeshsarkar@animeshsarkar:~/Test$ srestore
Index ID                  DeletedAt             OriginalPath
----- ------------------- --------------------- ------------------------------------------------
1     1768166694          2026-01-12 02:54:54   /home/animeshsarkar/Test/Deleteme
2     1768166683          2026-01-12 02:54:39   /home/animeshsarkar/Test/test3.txt
3     1768166681          2026-01-12 02:54:39   /home/animeshsarkar/Test/test2.txt
4     1768166680          2026-01-12 02:54:39   /home/animeshsarkar/Test/test1.txt
Enter the index of the file to restore:
```

- ***trash-cleanup*** the items which are older than assigned age i.e. 2 min for this case

```
animeshsarkar@animeshsarkar:~/Test$ srmv testmeagain.txt
Move 'testmeagain.txt' to trash
animeshsarkar@animeshsarkar:~/Test$ srestore
Index ID                  DeletedAt             OriginalPath
----- ------------------- --------------------- ------------------------------------------------
1     1768166932          2026-01-12 02:58:52   /home/animeshsarkar/Test/testmeagain.txt
2     1768166694          2026-01-12 02:54:54   /home/animeshsarkar/Test/Deleteme
3     1768166683          2026-01-12 02:54:39   /home/animeshsarkar/Test/test3.txt
4     1768166681          2026-01-12 02:54:39   /home/animeshsarkar/Test/test2.txt
5     1768166680          2026-01-12 02:54:39   /home/animeshsarkar/Test/test1.txt
Enter the index of the file to restore: ^C
animeshsarkar@animeshsarkar:~/Test$ trash-cleanup
animeshsarkar@animeshsarkar:~/Test$ srestore
Index ID                  DeletedAt             OriginalPath
----- ------------------- --------------------- ------------------------------------------------
1     1768166932          2026-01-12 02:58:52   /home/animeshsarkar/Test/testmeagain.txt
Enter the index of the file to restore:
```

7. **Alias rm**

Still can use **rm** via: ***\rm filename***

8. Schedule clean-up script (**use at your own risk**)

Will do later need some extra config to set up. If you want to setup just google it. And if you have a minute, just read the below explanation, it's awesome.

- Using **crontab -e (Cron jobs are executed by the cron daemon, which runs in the background independent of the shell and continues to run even when the user is logged out)**

  ┌─ minute (0–59)
  │ ┌─ hour (0–23)
  │ │ ┌─ day of month (1–31)
  │ │ │ ┌─ month (1–12)
  │ │ │ │ ┌─ day of week (0–7)
  │ │ │ │ │
  **** * * * command (absolute path)***

  How I remember this thing:

  - First of all, ***Wildcard character (*).*** It means **everything**.
  - Now, ***1 * * * *** means:

    1 = at 1 min
    * = every hour of the day
    * = every day of the month
    * = every month of the year
    * = every day of the week
    For example: the script will run, at 00:01, then 01:01, then 02:01 … 23:01. And then repeat the next day.

  - Common schedules

    ***0 * * * *** = Every hour (at the top of the hour). Like 17:00, 18:00, 00:00 etc.

    0 0 * * * = Every night at midnight. Like Monday at 00:00, Tuesday at 00:00 etc.
    0 = 0^{th} min
    0 = 0^{th} hour i.e. at midnight
    *** * * =** every day; every month; every day of the week

    ****/15 * * * **** = Every 15 minutes (the / means "increment").

23 9-17 * * 1-5 = Every 23rd min on the hour, but only between 9 AM and 5 PM, and only on weekdays (Monday-Friday).
Like on Monday at 09:23, 10:23 …17:23, on Tuesday at 09:23, 10:23 …17:23, … Friday at 09:23, 10:23 …17:23. But will not be executed on weekends. It's their holiday.