

How to use “safe-rm-restore”

1. Git clone script from github: [sate-rm-restore](https://github.com/Thin-Panha/safe-rm-restore)

git clone <https://github.com/Thin-Panha/safe-rm-restore.git>

2. Setup the scripts

cd safe-rm-restore

bash setup.sh

After setup, we can use command “**rm**” and “**restore**” to remove file/directory to /tmp/trash/ and restore it back to original place.

3. Create and delete file or directory for test script

```
panha@NEKE-2: ~/Project$ git clone https://github.com/Thin-Panha/safe-rm-restore.git
Cloning into 'safe-rm-restore'...
remote: Enumerating objects: 24, done.
remote: Counting objects: 100% (24/24), done.
remote: Compressing objects: 100% (17/17), done.
remote: Total 24 (delta 5), reused 21 (delta 5), pack-reused 0 (from 0)
Receiving objects: 100% (24/24), 6.19 KiB | 6.19 MiB/s, done.
Resolving deltas: 100% (5/5), done.
panha@NEKE-2: ~/Project$ cd safe-rm-restore/
panha@NEKE-2: ~/Project/safe-rm-restore$ bash setup.sh
✔ Setup complete. You can now use 'rm' and 'restore' safely!
🕒 Auto-deletion cron job added: Old trash items removed daily after 7 days.
panha@NEKE-2: ~/Project/safe-rm-restore$ cd ..
panha@NEKE-2: ~/Project$
panha@NEKE-2: ~/Project$ echo "Text 01" > file.txt
panha@NEKE-2: ~/Project$ rm file.txt
panha@NEKE-2: ~/Project$
panha@NEKE-2: ~/Project$ echo "Text 02" > file.txt
panha@NEKE-2: ~/Project$ rm file.txt
panha@NEKE-2: ~/Project$
```

```
panha@NEKE-2: ~/Project$
panha@NEKE-2: ~/Project$ mkdir Dir
panha@NEKE-2: ~/Project$ touch Dir/hi.txt
panha@NEKE-2: ~/Project$ tree
```

```
├── Dir
│   └── hi.txt
├── safe-rm-restore
│   ├── README.md
│   └── setup.sh
```

3 directories, 3 files

```
panha@NEKE-2: ~/Project$ rm -r Dir/
panha@NEKE-2: ~/Project$
panha@NEKE-2: ~/Project$ mkdir Dir
panha@NEKE-2: ~/Project$ touch Dir/hello.txt
panha@NEKE-2: ~/Project$ tree
```

```
├── Dir
│   └── hello.txt
├── safe-rm-restore
│   ├── README.md
│   └── setup.sh
```

3 directories, 3 files

```
panha@NEKE-2: ~/Project$ rm -r Dir/
panha@NEKE-2: ~/Project$
```

4. Show in /tmp/trash/ :

```
panha@NEKE-2: ~/Project$
panha@NEKE-2: ~/Project$ ls /tmp/trash/
Dir_2025-07-19_21:16:05  file_2025-07-19_21:15:47.txt  trash.log
Dir_2025-07-19_21:17:17  file_2025-07-19_21:16:05.txt  trash.log.json
panha@NEKE-2: ~/Project$
panha@NEKE-2: ~/Project$ cat /tmp/trash/trash.log
/home/panha/Project/file.txt -> /tmp/trash/file_2025-07-19_21:15:47.txt @ 2025-07-19_21:15:47
/home/panha/Project/file.txt -> /tmp/trash/file_2025-07-19_21:16:05.txt @ 2025-07-19_21:16:05
/home/panha/Project/Dir -> /tmp/trash/Dir_2025-07-19_21:16:55 @ 2025-07-19_21:16:55
/home/panha/Project/Dir -> /tmp/trash/Dir_2025-07-19_21:17:17 @ 2025-07-19_21:17:17
panha@NEKE-2: ~/Project$
```

5. Restore file

Ues: restore -> for file

retore -r -> for directory

```
panha@NEKE-2: ~/Project$ restore file.txt
Multiple matches found:
[1] /home/panha/Project/file.txt -> /tmp/trash/file_2025-07-19_21:15:47.txt @ 2025-07-19_21:15:47
[2] /home/panha/Project/file.txt -> /tmp/trash/file_2025-07-19_21:16:05.txt @ 2025-07-19_21:16:05
Select number to restore: 1
Restored to: /home/panha/Project/file.txt
panha@NEKE-2: ~/Project$ cat file.txt
Text 01
panha@NEKE-2: ~/Project$
panha@NEKE-2: ~/Project$ restore file.txt
▲ File already exists: /home/panha/Project/file.txt
Choose an option:
[1] Cancel restoring
[2] Rename restored file (append: _restored_TIMESTAMP)
[3] Overwrite existing file
Your choice: 2
Restored to: /home/panha/Project/file_restored_2025-07-19_21:25:02.txt
panha@NEKE-2: ~/Project$ ls
file_restored_2025-07-19_21:25:02.txt  file.txt  safe-rm-restore
panha@NEKE-2: ~/Project$ cat file_
cat: file_: No such file or directory
panha@NEKE-2: ~/Project$ cat file_restored_2025-07-19_21:25:02.txt
Text 02
panha@NEKE-2: ~/Project$
```

6. Restore Directory

```
panha@NEKE-2: ~/Project$ restore Dir
Multiple matches found:
[1] /home/panha/Project/Dir -> /tmp/trash/Dir_2025-07-19_21:16:55 @ 2025-07-19_21:16:55
[2] /home/panha/Project/Dir -> /tmp/trash/Dir_2025-07-19_21:17:17 @ 2025-07-19_21:17:17
Select number to restore: 1
Restored to: /home/panha/Project/Dir
panha@NEKE-2: ~/Project$ tree
├── Dir
│   ├── hi.txt
│   ├── file_restored_2025-07-19_21:25:02.txt
│   └── file.txt
├── safe-rm-restore
│   ├── README.md
│   ├── restore
│   ├── rm
│   └── setup.sh
└── 3 directories, 7 files
panha@NEKE-2: ~/Project$
panha@NEKE-2: ~/Project$ restore Dir
▲ File already exists: /home/panha/Project/Dir
Choose an option:
[1] Cancel restoring
[2] Rename restored file (append: _restored_TIMESTAMP)
[3] Overwrite existing file
Your choice: 3
Restored to: /home/panha/Project/Dir
panha@NEKE-2: ~/Project$ tree
├── Dir
│   ├── hello.txt
│   ├── file_restored_2025-07-19_21:25:02.txt
│   └── file.txt
├── safe-rm-restore
│   ├── README.md
│   ├── restore
│   ├── rm
│   └── setup.sh
```

7. Show /tmp/trash/trash.log.json to view action that we did of “rm” and “restore”:

```
panha@NEKE-2: ~/Project$ cat /tmp/trash/trash.log.json
{"action": "delete", "filename": "file.txt", "original_path": "/home/panha/Project/file.txt", "deleted_path": "/tmp/trash/file_2025-07-19_21:15:47.txt", "timestamp": "2025-07-19_21:15:47"}
{"action": "delete", "filename": "file.txt", "original_path": "/home/panha/Project/file.txt", "deleted_path": "/tmp/trash/file_2025-07-19_21:16:05.txt", "timestamp": "2025-07-19_21:16:05"}
{"action": "delete", "filename": "Dir", "original_path": "/home/panha/Project/Dir", "deleted_path": "/tmp/trash/Dir_2025-07-19_21:16:55", "timestamp": "2025-07-19_21:16:55"}
{"action": "delete", "filename": "file.txt", "original_path": "/home/panha/Project/file.txt", "deleted_path": "/tmp/trash/file_2025-07-19_21:17:17", "timestamp": "2025-07-19_21:17:17"}
{"action": "restore", "filename": "file.txt", "restored_path": "/home/panha/Project/file.txt", "from_trash": "/tmp/trash/file_2025-07-19_21:15:47.txt", "timestamp": "2025-07-19_21:24:32"}
{"action": "restore", "filename": "file_restored_2025-07-19_21:25:02.txt", "restored_path": "/home/panha/Project/file_restored_2025-07-19_21:25:02.txt", "from_trash": "/tmp/trash/file_2025-07-19_21:16:05.txt", "timestamp": "2025-07-19_21:24:45"}
{"action": "restore", "filename": "Dir", "restored_path": "/home/panha/Project/Dir", "from_trash": "/tmp/trash/Dir_2025-07-19_21:16:55", "timestamp": "2025-07-19_21:27:45"}
{"action": "delete", "filename": "Dir", "original_path": "/home/panha/Project/Dir", "deleted_path": "/tmp/trash/Dir_2025-07-19_21:28:08", "timestamp": "2025-07-19_21:28:08"}
{"action": "restore", "filename": "Dir", "restored_path": "/home/panha/Project/Dir", "from_trash": "/tmp/trash/Dir_2025-07-19_21:17:17", "timestamp": "2025-07-19_21:27:59"}
panha@NEKE-2: ~/Project$
```

8. Show “**crontab**” to view auto delete auto from **/tmp/trash** after 7 day:

```
panha@NEKE-2: /Project$ crontab -l
# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').
#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h dom mon dow   command
0 10 * * * /home/yourusername/foo.sh

0 0 * * * find /tmp/trash -mindepth 1 -mtime +7 -exec rm -rf {} +
panha@NEKE-2: /Project$
```