HW3 - Group 15

Deepak Gupta, Jatin Garg, Surbhi Bansal, Fnu Yashu

IOCTL's

1) **Undo ioctl** : **hw3/undo_ioctl.c**
   It decompresses/decrypts file as needed and moves it from trashbin folder back to its original location.

2) **Purge trashbin ioctl** : **hw3/purge_ioctl.c**
   It delete all the files from the trashbin folder belonging to particular user. If the user is root, then trashbin is purged completely.

3) **Set cipher Key** : **hw3/cipher_key_ioctl.c**
   If encryption is used, users will be able to set their cipher key via this ioctl.

Proc entries

1) **Trashbin Folder creation : hw3/proc_create_bin.c**
   Trashbin folder is created when this module is loaded using 'insmod proc_create_bin' command at root (/). Its contents are protected. A user can only ls, cat, stat files which belongs to him. Only root has permissions to view all files.

2) **Maximum number of files in trashbin folder** : **fs/proc/trashb_max_size.c**
   User can set the maximum number of files allowed in trashbin folder. By default it is set to 10. If trashbin has more files than this value, then background cleaning happens.
   Background cleaning thread : **hw3/trashb_clean.c**
   A kthread wakes up periodically(every 20 sec), iterates over all the files in trashbin folder and delete older files according to timestamp. This thread is started when trashbin folder is created.

3) **Maximum queue length** : **fs/proc/trashb_max_qlen.c**
   Max queue length can also be set by user. By default it is set to 10.
   a) If queue's length is reduced, existing entries in queue are still processed.
   b) If queue's length is reduced to 0, then all files are processed synchronously after existing entries are processed asynchronously.

4) **List queue's content** : **fs/proc/queue_info.c**
   User can list the queue's content to check currently queued items.

Clone Flags

1) **CLONE_PROT_MV (m)** : Flag used to move file to trashbin folder. When file is moved date and time is added to its name.
2) **CLONE_PROT_ZIP (z)** : Flag used to compress file before moving to trashbin folder. When file is compressed .cmp extension is added to its name.
3) **CLONE_PROT_ENC (e)** : Flag used to encrypt file before moving to trashbin folder. When a file is encrypted .enc extension is added to its name.

Combination of flags

1) No flags : File is permanently deleted, not moved to trashbin
2) Only m : File is moved to trashbin folder
3) Only z : File is first moved to trashbin folder and then compressed
4) Only e : File is first moved to trashbin folder and then encrypted
5) m and z : same as only z
6) m and e : same as only e
7) z and e : File is moved to trashbin folder, compressed, then encrypted.
8) m, z and e : same as z and e.

Design

1) Trashbin folder is created at reboot at root with name ".trashbin".
2) When a user deletes a file:
    a) If file size is less than 4KB, it is processed synchronously
    b) If file size is greater than 4KB, it is processed asynchronously

    Asynchronous processing of files

    a) File is added to work queue whose max length is 10 by default.
    b) If the work queue is full, then it is moved to wait queue.
    c) Item is moved from wait queue to work queue whenever an item is taken up by thread from work queue.

    Work and Wait queue's processing

    a) A scheduler thread is started when first file comes for deletion. It wakes up every 20 sec and check the length of both queues and schedules all other threads depending upon the length.
    b) We have used 3 checkpoints for starting kthreads dynamically according to length of work queue:
        i) Low_Watermark : It's value is 1. This means the number of files in work queue + wait queue is less than 3. Only 1 thread will be started in this case as work is less.
        ii) Medium_Watermark : It's value is 3. This means number of files in work queue + wait queue is less than 5. In this case, 2 threads will be started.
        iii) High_Watermark : It's value is 5. This means number of files in work queue + wait queue is greater than 5. In this case, 4 threads will be started.
    c) Threads are also killed dynamically according to length of both queues

i) If length if both queues is greater than 5, no thread is killed
ii) If length is less than equal to 5, then 2 threads are killed. Only 2 threads would be left running.
iii) If length is less than equal to 3, then 1 more thread is killed. Only 1 thread would be left running.

Files Modified

1) fs/namei.c
2) fs/readdir.c
3) fs/stat.c
4) fs/Makefile

Files Added

1) fs/proc/trashb_max_size.c
2) fs/proc/trashb_max_qlen.c
3) fs/proc/trashb_clean.c
5) fs/proc/workqueue_info.c
6) hw3/create_process.c
7) hw3/undo_ioctl.c
8) hw3/purge_ioctl.c
9) hw3/cipher_key_ioctl.c
10) hw3/proc_create_bin.c
11) hw3/Makefile
12) hw3/execve.c

Scripts (.sh) added

1) hw3/sync.sh
2) hw3/cp_sync.sh
3) hw3/sync_async.sh
4) hw3/cp_sync_async.sh
5) hw3/async_with_work_queue.sh
6) hw3/cp_async_with_work_queue.sh
7) hw3/async_with_work_wait_queue.sh
8) hw3/cp_async_with_work_wait_queue.sh
9) hw3/empty.sh
10) hw3/cp_empty.sh

Commands :

1) **Command to compile :**
   Run at hw3-cse506g15/ : make
   Run at hw3/ : make

2) Command to create the files to delete :

```
sh  hw3/cp*.sh
```

3) Command to remove the files :

```
./execve *.sh  OR ./create_process -emz file_to_be_deleted
```

4) Command to check the work queue content :

```
cat /proc/workqueue.c
```

5) Command to set the trashbin_size and trashbin_queue_len_max_size:

```
cat /proc/trashb_max_size
cat /proc/trashb_max_qlen
```

6) Command to fire the ioctls :

```
./undo_ioctl -u file_to_be_restored
./purge_ioctl
./cipher_key key_to_set
```