

Assignment 6

Srishti Shelke 111603056
Niramay Vaidya 111605075

1. A child process inherits real user id, real group id, effective user id and effective group id of the parent process, while process id and parent process id are not. Demonstrate.

The parent process executable's setgid bit was set so that it will have different rgid and egid. This will make the copy of egid of parent to egid of child more evident.

Similarly, the setuid bit was set and the owner of the file was changed such that it was different from the logged in user due to which the ruid and the euid will be different when the parent executes and copy of euid of parent to euid of child will also be more evident.

Output-

```
niranay@niranay: ~/Documents/sem7/AUP/assignment_6$ ll inherit_ids
-rwsrwxr-x 1 temp temp 9144 Sep 22 16:55 inherit_ids*
niranay@niranay: ~/Documents/sem7/AUP/assignment_6$ ./inherit_ids
Parent's real uid: 1000
Parent's effective uid: 1001
Parent's real gid: 1000
Parent's effective gid: 1000
Parent's process id: 6408
Parent's parent process id: 4659
Child's real uid: 1000
Child's effective uid: 1001
Child's real gid: 1000
Child's effective gid: 1000
Child's process id: 6409
Child's parent process id: 6408
```

2. Verify whether it is possible for a child process to handle a file opened by its parent Immediately after the fork() call?

Output-

```
niranay@niranay: ~/Documents/sem7/AUP/assignment_6$ cat parent_file
niranay@niranay: ~/Documents/sem7/AUP/assignment_6$ ./child_file handle
niranay@niranay: ~/Documents/sem7/AUP/assignment_6$ cat parent_file
coep:
niranay@niranay: ~/Documents/sem7/AUP/assignment_6$ ./child_file handle
niranay@niranay: ~/Documents/sem7/AUP/assignment_6$ cat parent_file
coepcoep:
```