



Artificial Intelligence and Data Science Department.

OS / Even Sem 2021-22 / Experiment 5.

YASH SARANG.

47 / D6AD.

EXPERIMENT - 5.

System Calls 2

Aim: Extend the above experiment to explore the following system calls: open, read, write, close, getpid, setpid, getuid, getgid, getegid, geteuid.

OS EXPERIMENT 5

Aim: To explore the following system calls
open, read, write, close, getpid,
setpid, getuid, getgid, getegid,
geteuid.

- Theory:**
- ① Read - to read from a file descriptor returns number of bytes read file position is advanced.
 - ② Open - to open a file/directory or an app
 - ③ write - writes upto count bytes of a file as referenced by the file descriptor.
 - ④ close - used to close the currently open file/directory
 - ⑤ getpid - gets the process ID (pid) of the calling process.
 - ⑥ setpid - used to set a unique process ID for a particular process.
 - ⑦ getuid - used to get the current user id of the calling process. Shall always be 0 successful and no return value is reserved to indicate the error.

Theory:

⑧ `setuid` - if the process has appropriate privileges, `setuid(1)` shall set the real user ID, effective user ID, and the saved set user ID of the calling process to UID.

⑨ `getgid` - returns the real group ID of the calling process. `getgid(1)` is always successful and no return value is reserved.

⑩ `setgid` - if the process has appropriate privileges, `setgid(1)` shall set the real group ID, effective group ID and saved set group ID of the calling process.

⑪ `getegid` - `getgid` subroutine returns the effective group ID of the calling process.

⑫ `setuid` - if uid is equal to the real user ID or the saved set user ID, or the process has appropriate privileges, `setuid(1)` shall set effective user ID of the calling process to uid.

Code:

```
Open  [icon] *lab5.cpp
~/Desktop/Example/Assignment Save [icon] [icon] [icon] [icon]

2 #include<sys/types.h>
3 #include<sys/stat.h>
4 #include<fcntl.h>
5 #include<unistd.h>
6 #include<stdlib.h>
7
8 int main(int args, char *argv[])
9 {
10     int fd;
11     char buf[14];
12     fd = open("trial.txt", O_CREAT | O_WRONLY, 0600);
13
14     if (fd == -1)
15     {
16         printf("Failed to create and open file\n");
17         exit(1);
18     }
19     write(fd, "Hello World\n", 13);
20     close(fd);
21     fd = open("trial.txt", O_RDONLY);
22
23     if (fd == -1)
24     {
25         printf("Failed to open and read\n");
26         exit(1);
27     }
28
29     read(fd, buf, 13);
30     buf[13] = '\0';
31     close(fd);
32     printf("buf: %s\n", buf);
33
34     printf("Real id is %d\n", getuid());
35     printf("Effective id is %d\n", geteuid());
36     printf("Real Group ID is %d\n", getgid());
37     printf("Effective Group ID is %d\n", getegid());
38     return 0;
39 }
```

C++ ▾ Tab Width: 8 ▾ Ln 37, Col 57 ▾ INS

Snippets:

```
(kali@kali)-[~/Desktop/Example/Assignment]
$ g++ lab5.cpp -o output5

(kali@kali)-[~/Desktop/Example/Assignment]
$ ./output5
buf: Hello World

Real id is 1000
Effective id is 1000
Real Group ID is 1000
Effective Group ID is 1000
```

Conclusion:-

Thus we can use the above commands and have learned in-depth about them.
