

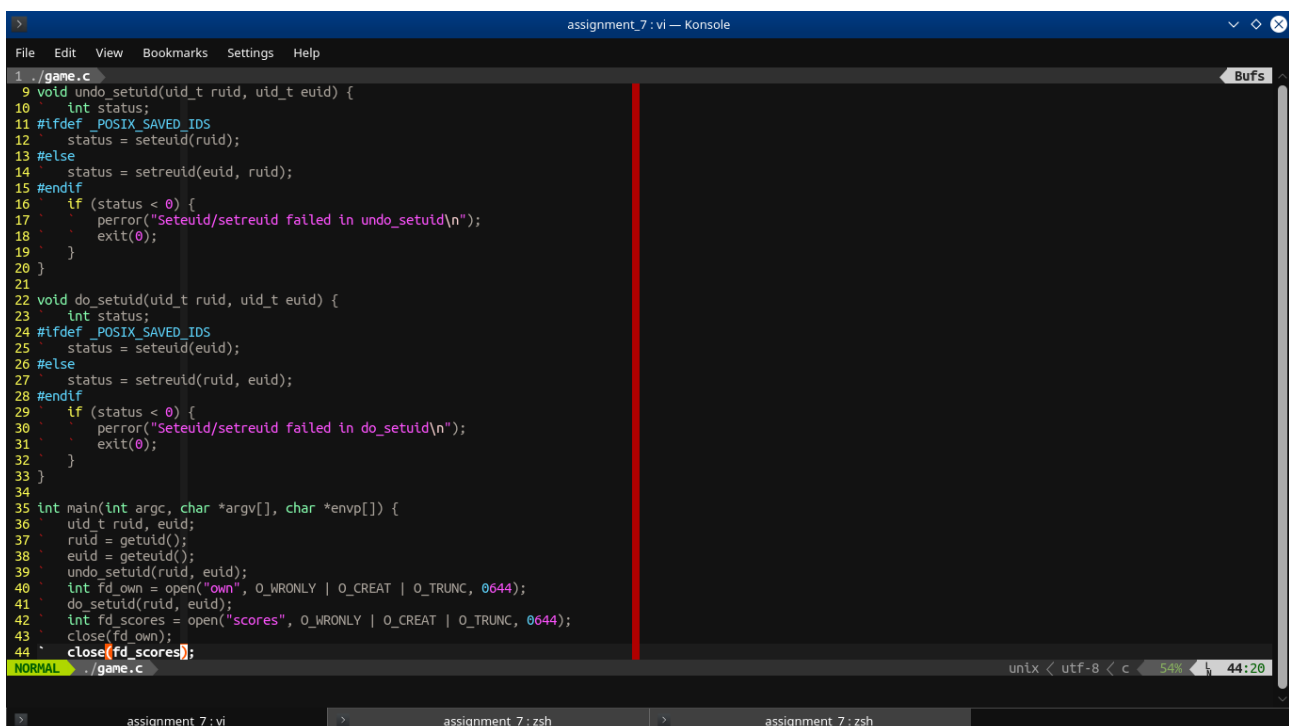
Assignment 7

Niramay Vaidya 111605075
Srishti Shelke 111603056

2. Create a game program that switches between the effective user ID and real user ID. The game player may write details (like game iteration number) to a file owned by the game player and manipulates a scores file that should be writable only by the game program owner. Both the game program and scores file are owned by the game program owner. Demonstrate that the game player can switch between the files in turns as own file, scores file, own file and scores file.

```
niramay@niramay: ~/Documents/sem7/AUP/assignment_7 $ ll game own scores
-rw-rw-r-x 1 temp  niramay 9136 Oct 14 19:37 game*
-rw-r--r-- 1 niramay niramay  4 Oct 14 19:37 own
-rw-r--r-- 1 temp  niramay  4 Oct 14 19:37 scores
niramay@niramay: ~/Documents/sem7/AUP/assignment_7 $
```

Code-



```
1 ./game.c
9 void undo_setuid(uid_t ruid, uid_t euid) {
10     int status;
11     #ifdef POSIX_SAVED_IDS
12     status = seteuid(ruid);
13     #else
14     status = seteuid(euid, ruid);
15     #endif
16     if (status < 0) {
17         perror("Seteuid/setreuid failed in undo_setuid\n");
18         exit(0);
19     }
20 }
21
22 void do_setuid(uid_t ruid, uid_t euid) {
23     int status;
24     #ifdef POSIX_SAVED_IDS
25     status = seteuid(euid);
26     #else
27     status = seteuid(ruid, euid);
28     #endif
29     if (status < 0) {
30         perror("Seteuid/setreuid failed in do_setuid\n");
31         exit(0);
32     }
33 }
34
35 int main(int argc, char *argv[], char *envp[]) {
36     uid_t ruid, euid;
37     ruid = getuid();
38     euid = geteuid();
39     undo_setuid(ruid, euid);
40     int fd_own = open("own", O_WRONLY | O_CREAT | O_TRUNC, 0644);
41     do_setuid(ruid, euid);
42     int fd_scores = open("scores", O_WRONLY | O_CREAT | O_TRUNC, 0644);
43     close(fd_own);
44     close(fd_scores);
45 }
```

```
assignment_7: vi — Konsole
File Edit View Bookmarks Settings Help
1 ./game.c
45 int num;
46 char newline = '\n';
47 char ch;
48 for (int i = 1; i <= 4; i++) {
49     if (i % 2 == 1) {
50         undo_setuid(ruid, euid);
51     }
52     else {
53         do_setuid(ruid, euid);
54     }
55     fd_own = open("own", O_WRONLY | O_APPEND);
56     fd_scores = open("scores", O_WRONLY | O_APPEND);
57     ch = (char)(i + 48);
58     if ((num = write(fd_own, &ch, sizeof(char))) != -1) {
59         if (num < sizeof(char)) {
60             printf("Less than a byte written\n");
61         }
62     }
63     if ((num = write(fd_own, &newline, sizeof(char))) != -1) {
64         if (num < sizeof(char)) {
65             printf("Less than a byte written\n");
66         }
67     }
68     if ((num = write(fd_scores, &ch, sizeof(char))) != -1) {
69         if (num < sizeof(char)) {
70             printf("Less than a byte written\n");
71         }
72     }
73     if ((num = write(fd_scores, &newline, sizeof(char))) != -1) {
74         if (num < sizeof(char)) {
75             printf("Less than a byte written\n");
76         }
77     }
78     close(fd_own);
79     close(fd_scores);
80 }
81 return 0;
82 }
NORMAL ./game.c
unix < utf-8 < c 77% 63:41
assignment_7: vi assignment_7: zsh assignment_7: zsh
```

```
assignment_7: vi — Konsole
File Edit View Bookmarks Settings Help
1 ./game.c
56 id = geteuid();
57 info = (id == euid) ? " euid:temp:1001" : " euid:niramay:1000";
58 len = strlen(info);
59 fd_own = open("own", O_WRONLY | O_APPEND);
60 fd_scores = open("scores", O_WRONLY | O_APPEND);
61 ch = (char)(i + 48);
62 if ((num = write(fd_own, &ch, sizeof(char))) != -1) {
63     if (num < sizeof(char)) {
64         printf("Less than a byte written\n");
65     }
66 }
67 if ((num = write(fd_own, info, len)) != -1) {
68     if (num < len) {
69         printf("Less than expected number of bytes written\n");
70     }
71 }
```

Output-

```
assignment_7: vi — Konsole
File Edit View Bookmarks Settings Help
1 ./own > 2 ./scores >
1 1 euid:niramay:1000
2 3 euid:niramay:1000
1 2 euid:temp:1001
2 4 euid:temp:1001
NORMAL ./own
unix < utf-8 50% 1:1
./scores
50% 1
assignment_7: vi assignment_7: vi
```

1. You have to create a process tree as shown below. Then you create a process group of (3, 4, 5) so that later (not to be implemented now) process 0 can send a signal to this group.

Code-

```
assignment_7: vi — Konsole
File Edit View Bookmarks Settings Help
1 ./change_process_group.c
11 #define FIFO_NAME "transfer"
12
13 int main(int argc, char *argv[], char *envp[]) {
14     pid_t pid = 0;
15     /* process 0 */
16     if ((pid = fork()) == -1) {
17         perror("Fork failed in process 0\n");
18         return errno;
19     }
20     else if (pid == 0) {
21         /* process 1 */
22         if ((pid = fork()) == -1) {
23             perror("Fork failed in process 1\n");
24             return errno;
25         }
26     }
27     else if (pid == 0) {
28         /* process 5 */
29         printf("process 5 pid: %ld ppid: %ld\n", (long)getpid(), (long)getppid());
30         int fd = open(FIFO_NAME, O_RDONLY);
31         if (fd == -1) {
32             perror("Open failed\n");
33             return errno;
34         }
35         int num;
36         long pgid 3;
37         if ((num = read(fd, &pgid, sizeof(long))) == -1) {
38             perror("Read failed\n");
39         }
40         else if (num < sizeof(long)) {
41             printf("Expected number of bytes not read\n");
42             exit(0);
43         }
44         printf("pgid 3 in process 5: %ld (to check FIFO transfer of pgid 3)\n", pgid);
45         printf("(before setpgid with pgid 3) process 5 pgid: %ld\n", (long)getpgid(0));
46         setpgid(0, (pid_t)pgid 3);
47         printf("(after setpgid with pgid 3) process 5 pgid: %ld\n", (long)getpgid(0));
48     }
49 }
NORMAL ./change_process_group.c
unix < c 33% 46:46
```

```
assignment_7: vi — Konsole
File Edit View Bookmarks Settings Help
1 ./change_process_group.c
47     sleep(10);
48     return 0;
49 }
50 else {
51     printf("process 1 pid: %ld ppid: %ld\n", (long)getpid(), (long)getppid());
52     sleep(10);
53     return 0;
54 }
55 }
56 else {
57     if ((pid = fork()) == -1) {
58         perror("Fork failed in process 0\n");
59         return errno;
60     }
61     else if (pid == 0) {
62         /* process 2 */
63         if ((pid = fork()) == -1) {
64             perror("Fork failed in process 2\n");
65             return errno;
66         }
67     }
68     else if (pid == 0) {
69         /* process 3 */
70         if ((pid = fork()) == -1) {
71             perror("Fork failed in process 3\n");
72             return errno;
73         }
74     }
75     else if (pid == 0) {
76         /* process 4 */
77         printf("process 4 pid: %ld ppid: %ld\n", (long)getpid(), (long)getppid());
78         /* if setpgid here gets called before the setpgid(0, 0) call
79          * in process 3, then pgid of process 4 will be set to the
80          * old pgid of process 3, not its new one
81          */
82         // setpgid(0, getpgid(getppid()));
83         /* even if just printf is called here, if it gets called
84          * before the setpgid(pid, getpid(0)) call in process 3,
85          */
86     }
87 }
NORMAL ./change_process_group.c
unix < c 59% 82:46
```

```
assignment_7: vi — Konsole
File Edit View Bookmarks Settings Help
1 ./change_process_group.c
83      * then the pgid of process 4 will be printed as the old
84      * pgid instead of the new one
85      */
86      // printf("process 4 pgid: %ld\n", (long)getpgid(0));
87      sleep(10);
88      return 0;
89  }
90  else {
91      printf("process 3 pid: %ld ppid: %ld\n", (long)getpid(), (long)getppid());
92      setpgid(0, 0);
93      printf("process 3 pgid: %ld\n", (long)getpgid(0));
94      setpgid(pid, getpgid(0));
95      printf("process 4 pgid: %ld\n", (long)getpgid(pid));
96      if (access(FIFO_NAME, F_OK) == -1) {
97          printf("In access\n");
98          if (mkfifo(FIFO_NAME, S_IFIFO | 0666) == -1) {
99              perror("Mkfifo failed\n");
100             return errno;
101         }
102     }
103     int fd = creat(FIFO_NAME, 0666);
104     if (fd == -1) {
105         perror("Creat failed\n");
106         return errno;
107     }
108     int num;
109     long pgid_3 = (long)getpgid(0);
110     printf("pgid_3 in process 3: %ld (to check FIFO transfer of pgid_3)\n", pgid_3);
111     if ((num = write(fd, &pgid_3, sizeof(long))) == -1) {
112         perror("Write failed\n");
113         return errno;
114     }
115     else if (num < sizeof(long)) {
116         printf("Expected number of bytes not written\n");
117         exit(0);
118     }
119 }
NORMAL ./change_process_group.c
unix < c 85% 118:21
assignment_6: vi assignment_7: vi assignment_7: zsh
```

```
assignment_7: vi — Konsole
File Edit View Bookmarks Settings Help
1 ./change_process_group.c
119     if (close(fd) == -1) {
120         perror("Close failed in process 3\n");
121         return errno;
122     }
123     sleep(10);
124     return 0;
125 }
126 else {
127     printf("process 2 pid: %ld ppid: %ld\n", (long)getpid(), (long)getppid());
128     sleep(10);
129     return 0;
130 }
131 }
132 }
133 else {
134     printf("process 0 pid: %ld\n", (long)getpid());
135     sleep(10);
136 }
137 }
138 return 0;
139 }
NORMAL ./change_process_group.c
unix < c 100% 139:1
assignment_6: vi assignment_7: vi assignment_7: zsh
```

(error handling has been done for setpgid() later on but is not visible in the screenshots of the code)

Output-

```
in@msys2:~/Documents/sem7/AUP/assignment_7$ ./change_process_group
process 0 pid: 13125
process 1 pid: 13126 ppid: 13125
process 5 pid: 13128 ppid: 13126
process 2 pid: 13127 ppid: 13125
process 3 pid: 13129 ppid: 13127
process 4 pid: 13130 ppid: 13129
process 3 pgid: 13129
process 4 pgid: 13129
pgid_3 in process 3: 13129 (to check FIFO transfer of pgid_3)
pgid_3 in process 5: 13129 (to check FIFO transfer of pgid_3)
(before setpgid with pgid_3) process 5 pgid: 13125
(after setpgid with pgid_3) process 5 pgid: 13129
in@msys2:~/Documents/sem7/AUP/assignment_7$
```