# Linux Operation System - Project 2

Team 19
108522078 陳頎皓
108522003 黃宇帆
108522011 楊承翰

一、

get_process_zero_session_group.c (system call source code)說明

1. for_each_process(struct tast_struct*) 掃描整個 process linked list
2. task_session(struct tast_struct*) 取得 sid
3. pid_nr(struct pid*) 取得 pid→numbers[0].nr

結果如下

```
james@james-VirtualBox:~/Desktop$ ./a.out
What follows are the PIDs of the porcesses that are in the same login session of
 process 0
[0] 2 [1] 3 [2] 5 [3] 6 [4] 7 [5] 8 [6] 9 [7] 10 [8] 11 [9] 12 [10] 13 [11] 15 [
12] 16 [13] 17 [14] 18 [15] 20 [16] 21 [17] 22 [18] 23 [19] 24 [20] 25 [21] 26 [
22] 27 [23] 28 [24] 30 [25] 31 [26] 32 [27] 33 [28] 34 [29] 35 [30] 36 [31] 37 [
32] 38 [33] 39 [34] 40 [35] 41 [36] 42 [37] 43 [38] 44 [39] 46 [40] 47 [41] 48 [
42] 49 [43] 50 [44] 51 [45] 62 [46] 63 [47] 64 [48] 65 [49] 87 [50] 88 [51] 89 [
52] 151 [53] 152 [54] 155 [55] 157 [56] 159 [57] 161 [58] 183 [59] 184 [60] 213
[61] 312 [62] 328 [63] 332 [64] 333 [65] 334 [66] 1750 [67] 1753 [68] 2042 [69]
2088 [70] 2134 [71] 2144 [72] 2145 [73] 2150
```

程式碼如下

```
asmlinkage int sys_get_process_zero_session_group(unsigned int *result, int size)
{
    struct task_struct *t;
    int exist = 0;
    for_each_process(t){
        if(exist>size)
            break;
        if(pid_nr(task_session(t)) == 0){
            result[exist++] = t->pid;
        }
    }
    return exist;
}
```

二、

get_process_session_group.c (system call source code)說明

1. 先找 current process 的 SID1
2. for_each_process(struct tast_struct*) 掃描整個 process linked list
3. 找整個 process linked list 的 SID，如果等於 SID1，把 pid 丟到 pid_vnr(pid)去撈 local pid，如果 local pid = 0，代表是 child 看不到 parent，就要用 task_active_pid_ns 去撈這個 task 的 namespace，然後再丟到 pid_nr_ns 去撈 local pid

結果如下

```
root@james-VirtualBox:~/Desktop# ./a.out
What follows are the PIDs of the porcesses that are in the same login session of
 this process
[0] 1611 [1] 1720 [2] 1721 [3] 1 [4] 23
```

程式碼如下

```
asmlinkage int sys_get_process_session_group(unsigned int *result, int size)
{
    struct task_struct *t1, *t2;
    int curr_pid,local_pid,  exist = 0;
    struct pid_namespace *ns;
    t1 = current;
    struct pid *sid, *pid;
    sid = task_session(t1);
    for_each_process(t2){
        if(exist>size)
            break;
        if(task_session(t2) == sid){
            pid = task_pid(t2);
            local_pid = pid_vnr(pid);
            if(local_pid == 0){
                ns = task_active_pid_ns(t2);
                local_pid = pid_nr_ns(pid, ns);
            }
            result[exist++] = local_pid;
        }
    }
    return exist;
}
```