운영체제 실습

[Assignment#5]

Class : 목34

Professor : 최상호 교수님

Student ID : 2021202003

Name : 강준우

Introduction

5-1에서는 processInfo 경로를 생성하고, 테스크 리스트를 기반으로 PID부터 상태까지 표기반으로 출력을 구현한다. 또한 pid를 1 또는 -1로 설정하는 기능을 통해서 필터링 기능을 추가하였다.
5-2에서는 fat 연결 리스트로 구현된 이미지 파일을 조작하여 사용자가 컨트롤할 수 있는 기능을 구현하였다, 내용 출력부터 파일 IO기능들을 수행하였다,

결과화면

Assignment 5-1

```
[ 5928.756298] e1000: ens33 NIC Link is Up 1000 Mbps Full Duplex, Flow Control: None
[ 5930.769346] e1000: ens33 NIC Link is Down
[ 5934.802395] e1000: ens33 NIC Link is Up 1000 Mbps Full Duplex, Flow Control: None
[ 6436.421568] proc_info: loading out-of-tree module taints kernel.
[ 6436.421856] proc_info: module verification failed: signature and/or required key missing - tainting kernel
[ 6436.439156] proc_info(2021202003): /proc/proc_2021202003/processInfo ready
[ 6533.850893] proc_info(2021202003): unloaded
[ 9919.013480] proc_info(2021202003): /proc/proc_2021202003/processInfo ready
```

로그에서 lsmod에 proc_info를 확인할 수 있다.

```
root@ubuntu:~/Assignment5/Assignment5-1# ls -l /proc/proc_2021202003/processInfo
-rw-rw-rw- 1 root root 0 Dec  2 06:37 /proc/proc_2021202003/processInfo
```

-rw-rw-rw-로 나와 모든 접근자에 대해 r과 w 권한을 부여함을 확인할 수 있다.

```
root@ubuntu:~/Assignment5/Assignment5-1# cat /proc/proc_2021202003/processInfo | head -n 30
Pid     PPid    Uid     Gid     utime       stime       State   Name
-----------------------------------------------------------------
1       0       0       0       406         5889        S       systemd
2       0       0       0       0           27          S       kthreadd
3       2       0       0       0           0           D       rcu_gp
4       2       0       0       0           0           D       rcu_par_gp
6       2       0       0       0           0           D       kworker/0:0H
8       2       0       0       0           0           D       mm_percpu_wq
9       2       0       0       0           8           S       ksoftirqd/0
10      2       0       0       0           147         D       rcu_sched
11      2       0       0       0           1           S       migration/0
12      2       0       0       0           0           S       idle_inject/0
14      2       0       0       0           0           S       cpuhp/0
15      2       0       0       0           0           S       cpuhp/1
16      2       0       0       0           0           S       idle_inject/1
17      2       0       0       0           7           S       migration/1
18      2       0       0       0           2           S       ksoftirqd/1
20      2       0       0       0           0           D       kworker/1:0H
21      2       0       0       0           0           S       cpuhp/2
22      2       0       0       0           0           S       idle_inject/2
23      2       0       0       0           5           S       migration/2
24      2       0       0       0           10          S       ksoftirqd/2
26      2       0       0       0           1           D       kworker/2:0H
27      2       0       0       0           1           S       cpuhp/3
28      2       0       0       0           0           S       idle_inject/3
29      2       0       0       0           7           S       migration/3
30      2       0       0       0           171         S       ksoftirqd/3
32      2       0       0       0           0           D       kworker/3:0H
33      2       0       0       0           2           S       kdevtmpfs
34      2       0       0       0           0           D       netns
```

systemd, kthreadd, rcu_* 등의 커널/ 유저 대표 프로세스를 확인할 수 있다.

```
root@ubuntu:~/Assignment5/Assignment5-1# echo 1 | sudo tee /proc/proc_2021202003/processInfo
1
root@ubuntu:~/Assignment5/Assignment5-1# cat /proc/proc_2021202003/processInfo
Pid     PPid    Uid     Gid     utime       stime       State   Name
-----------------------------------------------------------------
1       0       0       0       409         5897        S       systemd
```

출력이 1로 나옴에따라 w가 설정을 정상적으로 함을 확인할 수 있다.

이후 pid 1은 다음과 같다.

```
root@ubuntu:~/Assignment5/Assignment5-1# echo -1 | sudo tee /proc/proc_2021202003/processInfo
-1
root@ubuntu:~/Assignment5/Assignment5-1# cat /proc/proc_2021202003/processInfo | head -n 10
Pid     PPid    Uid     Gid     utime       stime       State   Name
-----------------------------------------------------------------
1       0       0       0       409         5897        S       systemd
2       0       0       0       0           27          S       kthreadd
3       2       0       0       0           0           D       rcu_gp
4       2       0       0       0           0           D       rcu_par_gp
6       2       0       0       0           0           D       kworker/0:0H
8       2       0       0       0           0           D       mm_percpu_wq
9       2       0       0       0           8           S       ksoftirqd/0
10      2       0       0       0           190         D       rcu_sched
```

-1로 필터 해재가 된 후에는 전체 목록에 대해서 변경점을 확인할 수 있다.

```
root@ubuntu:~/Assignment5/Assignment5-1# cat /proc/proc_2021202003/processInfo | grep -w systemd
1       0       0       0       409         5897        S       systemd
337     1       0       0       162         391         S       systemd-journal
381     1       0       0       169         126         S       systemd-udevd
745     1       101     103     85          173         S       systemd-resolve
746     1       102     104     28          26          S       systemd-timesyn
829     1       0       0       57          147         S       systemd-logind
1510    1       0       0       57          26          S       systemd
1686    1       1000    1000    123         174         S       systemd
root@ubuntu:~/Assignment5/Assignment5-1# cat /proc/proc_2021202003/processInfo | awk 'NR>2{c++}END{print c}'
304
```

데몬이 일관된 PPid=1을 가짐을 확인할 수 있다

Assignment 4-2

```
root@ubuntu:~/Assignment5/Assignment5-2# ./fat create a
root@ubuntu:~/Assignment5/Assignment5-2# ./fat write a "Hello, World!"
root@ubuntu:~/Assignment5/Assignment5-2# ./fat read a
Hello, World!
root@ubuntu:~/Assignment5/Assignment5-2# ./fat list
No  Name                Size   Head
0   a                   13     0
root@ubuntu:~/Assignment5/Assignment5-2# ./fat list
No  Name                Size   Head
0   a                   13     0
root@ubuntu:~/Assignment5/Assignment5-2# ./fat delete a
root@ubuntu:~/Assignment5/Assignment5-2# ./fat list
No  Name                Size   Head
root@ubuntu:~/Assignment5/Assignment5-2# ./fat create a
root@ubuntu:~/Assignment5/Assignment5-2# ./fat write a "Hello, World!"
root@ubuntu:~/Assignment5/Assignment5-2# ./fat read a > /tmp/out.txt
root@ubuntu:~/Assignment5/Assignment5-2# printf "Hello, World!" > /tmp/ref.txt
root@ubuntu:~/Assignment5/Assignment5-2# printf "Hello, World!" > /tmp/ref.txt
root@ubuntu:~/Assignment5/Assignment5-2# printf "Hello, World!" > /tmp/ref.txt
root@ubuntu:~/Assignment5/Assignment5-2# cmp -s /tmp/out.txt /tmp/ref.txt && echo "CONTENT OK" || echo "CONTENT MISMATCH"
CONTENT MISMATCH
root@ubuntu:~/Assignment5/Assignment5-2# ./fat create a
create: already exists
root@ubuntu:~/Assignment5/Assignment5-2# ./fat read no_such_file
read: no such file
root@ubuntu:~/Assignment5/Assignment5-2# ./fat delete no_such_file
delete: no such file
```

fat 바이너리 생성 확인 후 출력 Hello, World! 를 표준출력으로 확인하였다. 또한 리스트 기반으로 확인했을 때 메타 데이터에 대한 확인 또한 가능하다. 삭제후에는 리스트에서 또한 보이지 않으며 이외의 에러조건들도 정상적으로 보이는것을 확인할 수 있다.

고찰

.커널 5.4에서는 task gid를 얻기 위해 task->cred->gid 를 직접 참조하였는데, 해당 기능을 사용함에 따라 포인터 및 보안 관련된 에러를 확인할 수 있었다. 해당 내용을 추가로 알아본 결과 이런 OS관련 직접 참조는 데이터 누수 및 오작동을 유발할 수 있다는 사실을 알게 되었다.

Reference

실습자료 이용했습니다. 감사합니다.