

## OS Assignment1\_WIKI\_2022058740 양경혁

### Design :

이번 과제에서는 본인 학번과, process id, parents' process id를 출력하는 기능을 새로 만드는게 목표다. 첫번째로 getpid()라는 함수가 kernel directory의 sysproc.c 파일에 선언되어 있는데, 이 함수를 참고하여 getppid()라는 함수를 새로 sysproc.c 파일에 추가하여 앞서 예제로 해봤던 my\_userapp.c를 참고하여 시스템 콜을 가능하게 하려고 한다.

### Implement :

```
24 uint64
25 sys_getppid(void)
26 {
27     return myproc()->parent->pid;
28 }
```

위 사진의 코드는 sysproc.c에 getppid() 함수를 추가한 모습이다. 이 함수의 기능은 현재 프로세스의 부모 프로세스를 참조하여 그 process id를 호출한다.

```
94 extern uint64 sys_getppid(void);
121 [SYS_getppid] sys_getppid,
```

위 사진의 코드는 syscall.c에 sys\_getppid()를 선언한 모습이다. 그리고 바로 밑줄에 syscalls[]배열에 SYS\_getppid index를 추가함으로써 컴파일을 할 때 문제없이 가능하다.

```
24 #define SYS_getppid 23
```

위 사진의 코드는 getppid() 함수를 유저모드에서 호출시에 커널에 함수의 시스템 콜 번호를 전달하기 위해 syscall.h에 추가했다. syscall.h 파일에 이미 22번까지 함수들이 정의 되었어서 겹치지 않도록 23번으로 설정했다.

```
22 int getppid(void);
```

위 사진의 코드는 user.h에 추가한 코드인데, user.h는 xv6에서 사용자 영역에 사용할 수 있는 시스템 콜 함수들의 선언이 모여 있는 헤더파일이다. 그래서 유저 모드에서 함수를 호출할

```
36 entry("getppid");
```

수 있도록 함수의 선언을 알려주기 위해 추가했다.

위 사진의 코드는 usys.pl에 추가한 코드이다. usys.pl은 getpid() 함수를 시스템 콜 숫자인 23으로 변환하고, 그 숫자에 해당하는 함수를 호출하도록 도와주는 어셈블리 코드를 만든다. 그리고 그 결과값이 사용자 프로그램으로 돌아간다.

```
1 #include "kernel/types.h"
2 #include "user/user.h"
3 #include "kernel/stat.h"
4
5 int main() {
6     int pid = getpid();    // 현재 프로세스의 PID
7     int ppid = getppid();  // 부모 프로세스의 PID
8
9     printf("My student ID is 2022058740\n");
10    printf("My pid is %d\n", pid);
11    printf("My ppid is %d\n", ppid);
12
13    exit(0);
14 }
```

위 사진의 코드는 user 디렉토리에 ppid.c 파일의 코드이다. 우리가 최종적으로 xv6에서 원하는 결과를 얻기 위해 써야하는 명령어가 ppid이므로 ppid로 설정하였다.

144 \$U/\_ppid\

위 사진의 코드는 user/ppid.c를 컴파일해서 xv6에서 실행 가능한 프로그램 /ppid로 만들어주기 위해 Makefile에 추가했다.

Result :

```
yang-gyeonghyeog@kyeonghyeok-yangui-MacBookPro assignment-2022058740 % make qemu
qemu-system-riscv64 -machine virt -bios none -kernel kernel/kernel -m 128M -smp 1 -nographic
-global virtio-mmio.force-legacy=false -drive file=fs.img,if=none,format=raw,id=x0 -device v
irtio-blk-device,drive=x0,bus=virtio-mmio-bus.0

xv6 kernel is booting

init: starting sh
$ ppid
My student ID is 2022058740
My pid is 3
My ppid is 2
$
```

위 사진의 코드는 본인 컴퓨터에서 xv6에 ppid를 입력했을 때 결과이다. ppid를 xv6에 입력했을때, 시스템이 동작하는 흐름을 설명해보자면, make를 했을 당시에 usys.pl을 통하여 usys.S라는 어셈블리 파일을 만드는데, 이 어셈블리 파일은 getpid() 함수를 실제로 기능하도록 한다. Usys.S는 getpid()의 시스템 콜 번호를 가져오는데, 이는 우리가 kernel/syscall.h에 정의한 대로, 23을 가져온다. 또한, kernel/syscall.c의 syscall[] 배열에 sys\_getpid()로 연결되며 실행하도록 한다. 이때, sys\_getpid()함수는 sysproc.c에 선언되어 있어, sysproc.c에 선언한 대로 기능한다. 이전에 user/ppid.c를 실행하고, 그에 필요한 user/user.h에서 getpid()라는 함수의 선언유무를 확인하는데, 이때 컴파일을 할 때, 앞서 말한 usys.S에 연결되어 getpid() 함수의 결과값을 kernel에서 가져오는 것이다.

Troubleshooting : .