2018. 3. 26 월– 23 회차

과정 : TI, DSP, Xilinx Znq FPGA, MCU 기반의 프로그래밍 전문가 과정

Prof. 이상훈
gcccompil3r@gmail.com
Stu. 정상용
fstopdg@gmail.com

Linux 운영체제

1. exe
exe : 실행
ps 라는 main process 를 실행시키고, 인자로 ps, -e, -f 를 받는다.
0 은 끝남을 의미.

Ex>
```
#include <unistd.h>

int main(void)
{
   execlp("ps", "ps", "-e", "-f", 0);
   return 0;
}
```

2. exe 의 특징

```
#include <unistd.h>

int main(void)
{
   execlp("ps", "ps", "-e", "-f", 0);
   printf("after\n");
   return 0;
}
```
→ 위 프로그램을 동작시킬 경우, printf 함수가 동작하지 않음.

Memory layout 이 execlp(이 process 안에는 printf 가 포함되어있지 않음.)로 바뀌어버린다.
참고. execlp 는 둔갑술이라고 생각(fork 는 분신술이라 생각)

3. main process 의 함수를 출력하고 싶을 경우(w/ exe)
→ fork 를 이용

ex>
```
#include <stdio.h>
#include <unistd.h>

int main(void)
{
   int status;
   pid_t pid;
   if((pid = fork()) > 0)
   {
       wait(&status);
       printf("prompt >\n");
   }
   else if(pid == 0)
       execlp("ps", "ps", "-e", "-f", 0);
   return 0;
}
```

4. 동시다발적으로 프로그램을 돌리고 싶을 경우
→ fork & exe 을 이용(exe 는 외부의 process 를 가져옴)
Ex>
```
#include <unistd.h>
#include <stdio.h>

int main(void)
{
   int status;
   pid_t pid;
   if((pid = fork()) > 0)
   {
       wait(&status);
       printf("prompt>\n ");
   }
```

```
    else if(pid == 0)
        execl("./newpgm", "newpgm", "one", "two", (char *)0);
    return 0;
}
```

5. 환경변수 출력

→ 아래의 프로그램이 아닌, newpgm.c 에서 envp[](환경변수)를 인자로 추가

Ex>

```
#include <unistd.h>
#include <stdio.h>

int main(void)
{
    int status, i;
    pid_t pid;
    char *argv[] = {"./newpgm", "newpgm", "one", "two", 0};
    char *env[] = {"name = OS_Hacker", "age = 20", 0};



    if((pid = fork()) > 0)
    {
        wait(&status);
        printf("prompt > \n");
    }


    else if(pid == 0)
        execve("./newpgm", argv, env);

    return 0;
}
```

6. 시스템함수

```
#include <stdio.h>

int main(void)
{
    for(;;)
    {
```

```
        sleep(3);
        system("date");
    }

    printf("after\n");
    return 0;
}
```

→ date 자체를 출력(ls 로 바꾸면 ls 의 기능을 실행)

→ exe 와 달리 main process 의 다른 함수도 실행. 즉 system 은 내부에서 fork 를 자체적으로 진행

7. Daemon

→ 반드시 숙지하고 있어야하는 내용

Terminal 을 꺼도 Process 가 사라지지 않고 존재. 즉 서비스를 제공하는 소프트웨어(ex. 구글, 게임업체등)의 경우에 반드시 필요한 기능.

(확인하는 방법 : 파일을 실행시킨 후 ps -ef | grep a.out 에서 TTY 를 통하여 세션아이디를 확인 - 세션아이디가 ?로 나올경우 Daemon)

→ Daemon 을 없애고 싶은 경우

‘신이 등장 : kill -9 PID