

- ※ 다음 프로그램을 참고하여 조건에 맞는 프로그램을 작성 후 결과를 보이시오.
  - ※ 7장과 8장의 예제 소스 코드를 참조하시면 됩니다.
  - ※ 소스 파일명은 “날짜\_문제번호.c” 형식으로 git push 해 두세요.
- 예) 1번 문제 파일명 : 20230421\_1.c

```
$ cat ex08-03.c

#include <unistd.h>
#include <sys/types.h>

int main()
{
    pid_t pid1, pid2;
    int status;

    pid1 = pid2 = -1;

    pid1 = fork();
    if(pid1 > 0)
        pid2 = fork();

    if(pid1 > 0 && pid2 > 0)
    {
        waitpid(pid2, &status, 0);
        printf("parent: child2 - exit(%d)\n", status);
        waitpid(pid1, &status, 0);
        printf("parent: child1 - exit(%d)\n", status);
    }
    else if(pid1 == 0 && pid2 == -1)
    {
        sleep(1);
        exit(1);
    }
    else if(pid1 > 0 && pid2 == 0)
    {
        sleep(2);
        exit(2);
    }
    else
        printf("fail to fork\n");
}
```

< 문제 1 > ex08-03.c를 pid\_t형의 배열을 사용하여 여러 개의 자식 프로세스의 식별 번호를 저장하고, 자신이 부모 프로세스인지 자식 프로세스인지를 프로세스 식별 번호로 판단하도록 수정하시오.

```
// 작성 예시
$ cat 0803.c
....
int main(){
    pid_t pid[NUM];
    ....
    for( ; ; ){
        if( n == NUM )
            // 부모 프로세스
        else if ( n < NUM )
            // 자식 프로세스
    }
}
```

< 문제 2 > 다음과 같은 입력이 들어올 경우 fork와 exec 계열의 함수를 적절히 사용하여 각각의 파일에 대해 wc 명령어를 수행할 수 있도록 프로그램을 작성하시오.  
(wc는 특정 파일의 line, word, character 수를 출력하는 명령어이다.)

```
// 실행 결과 예시
$ ./0704wc a.txt b.txt c.txt
14  24  194  a.txt
10  25  192  b.txt
13  34  184  c.txt
```

< 문제 3 > execl을 사용하여 새로운 프로세스를 생성하되 그 전에 “MYDATA=5”라는 환경 변수를 추가하시오. 그리고 새로운 프로세스에서는 MYDATA를 비롯한 모든 환경변수 및 값을 화면에 출력하시오.(아래 소스 코드를 참고하세요.)

```
$cat ex08-12.c
#include <unistd.h>

main()
{
    char *envlist[] = {"APPLE=0", "BANANA=1", (char *)0 };
    execl("ex08-13", "ex08-13", (char *)0, envlist);
}

$cat ex08-13.c
#include <unistd.h>

extern char **environ;

main()
{
    while(*environ)
        printf("%s\n", *environ++);
}
```