

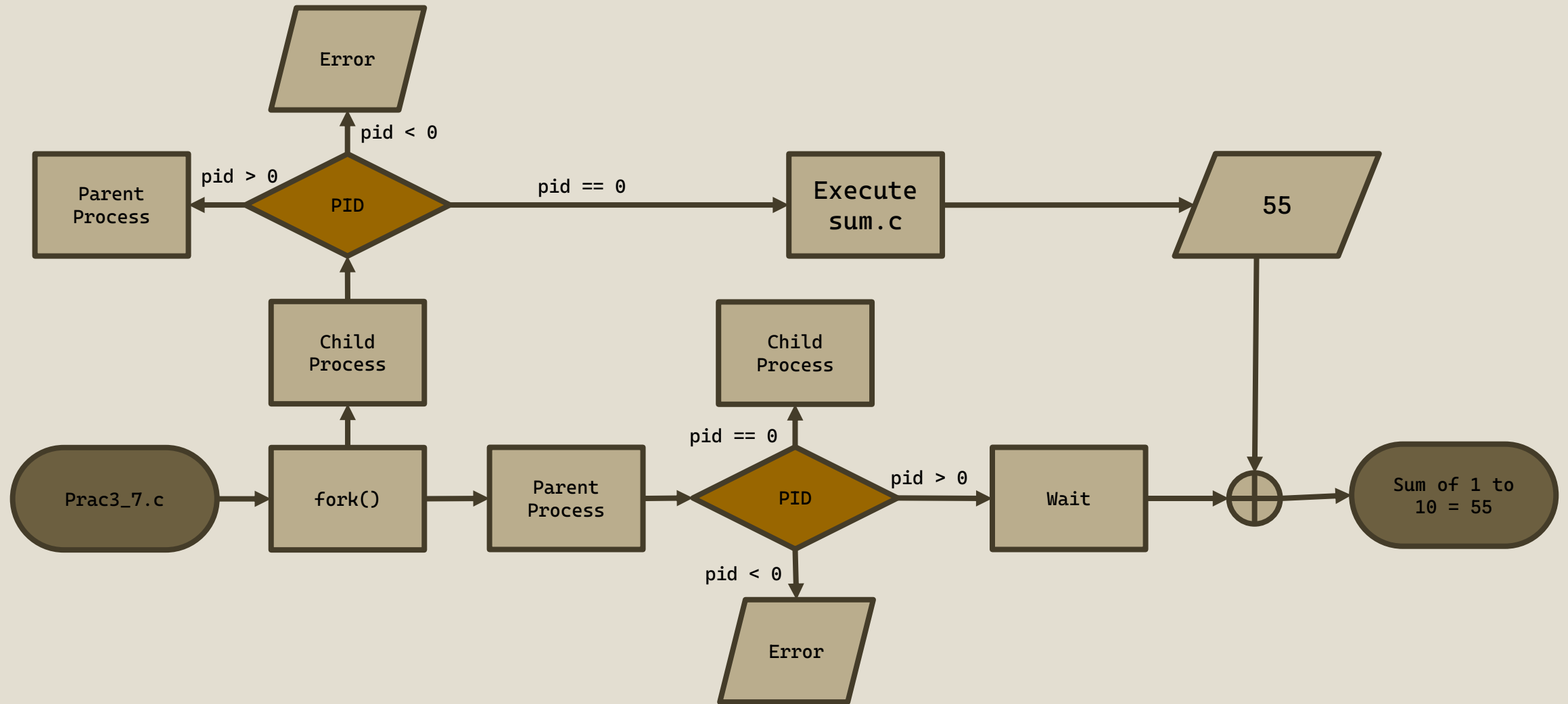


연습문제 #7, #8

2019305061 임주형

연습문제 #7

- 1부터 10까지의 합을 리턴하는 sum.c
- 자식 프로세스에서 sum.c 를 실행하여 종료코드로 리턴
- 부모 프로세스에서 리턴 받은 값을 출력



연습문제 #7 코드 및 결과

prac3_7.c

```
1 #include <stdio.h>
2 #include <sys/types.h>
3 #include <sys/wait.h>
4 #include <unistd.h>
5
6 int main() {
7     pid_t pid;
8     int status;
9
10    pid = fork(); // 자식 프로세스 생성
11
12    if (pid > 0) { // 부모 프로세스 코드
13        wait(&status); // 자식프로세스 종료 대기, status에 종료 코드 받음
14        printf("1에서 10까지 합한 결과는 %d\n", WEXITSTATUS(status));
15        return 0;
16    }
17    else if (pid == 0) { // 자식 프로세스 코드
18        execlp("./sum", "sum", NULL); // sum.c를 자식 프로세스로 실행
19    }
20    else { // fork() 오류
21        printf("fork 오류");
22        return 0;
23    }
24 }
```

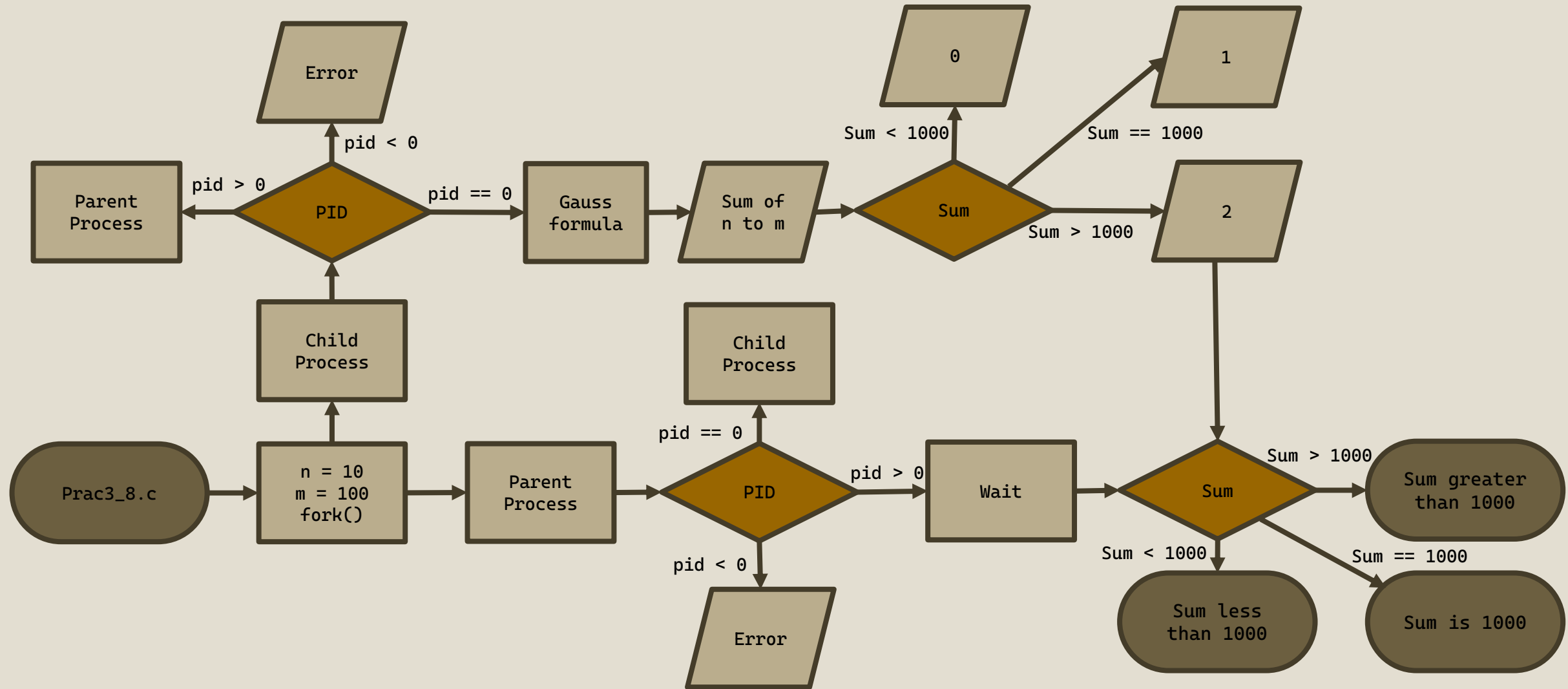
sum.c

```
1 #include <stdio.h>
2
3
4 int main(void) {
5     int i;
6     int sum = 0;
7
8     for (i = 1; i <= 10; i++) {
9         sum += i;
10    }
11
12    return sum;
13 }
```

```
~$ gcc -o sum sum.c
~$ gcc -o prac3_7 prac3_7.c
~$ ./prac3_7
1에서 10까지 합한 결과는 55
~$
```

연습문제 #8

- 자식 프로세스에게 $n = 10$, $m = 100$ 변수 전달
- 자식 프로세스는 n 부터 m 까지의 합을 구해 1000을 기준으로 종료코드를 달리 함
- 부모 프로세스는 자식 프로세스의 종료코드를 읽어 n 부터 m 까지의 합이 1000보다 큰 지 판별



연습문제 #8 코드 및 결과

```
#include <stdio.h>
#include <sys/types.h>
#include <sys/wait.h>
#include <unistd.h>
int main() {
    pid_t pid;
    int n = 10, m = 100;
    int sum = 0;
    int status;

    pid = fork(); // 자식 프로세스 생성

    if (pid > 0) { // 부모 프로세스 코드
        wait(&status); // 자식프로세스 종료 대기, status에 종료 코드 받음

        if (WEXITSTATUS(status) == 0)
            printf("부모 프로세스 : 1000보다 작다.\n");

        else if (WEXITSTATUS(status) == 1)
            printf("부모 프로세스 : 1000이다.\n");

        else if (WEXITSTATUS(status) == 2)
            printf("부모 프로세스 : 1000보다 크다.\n");

        return 0;
    }
```

```
    else if (pid == 0) { // 자식 프로세스 코드
        sum = (n + m) * (m - n + 1) / 2; //가우스 공식 사용
        printf("자식 프로세스 : %d에서 %d까지의 합은 %d\n", n, m, sum);

        if (sum < 1000) //1000보다 작으면 0리턴
            return 0;

        else if (sum == 1000) //1000이면 1리턴
            return 1;

        else //1000보다 크면 2리턴
            return 2;
    }
    else { // fork() 오류
        printf("fork 오류\n");
        return 0;
    }
}
```

```
~$ gcc -o prac3_8 prac3_8.c
~$ ./prac3_8
자식 프로세스 : 10에서 100까지의 합은 5005
부모 프로세스 : 1000보다 크다.
~$
```