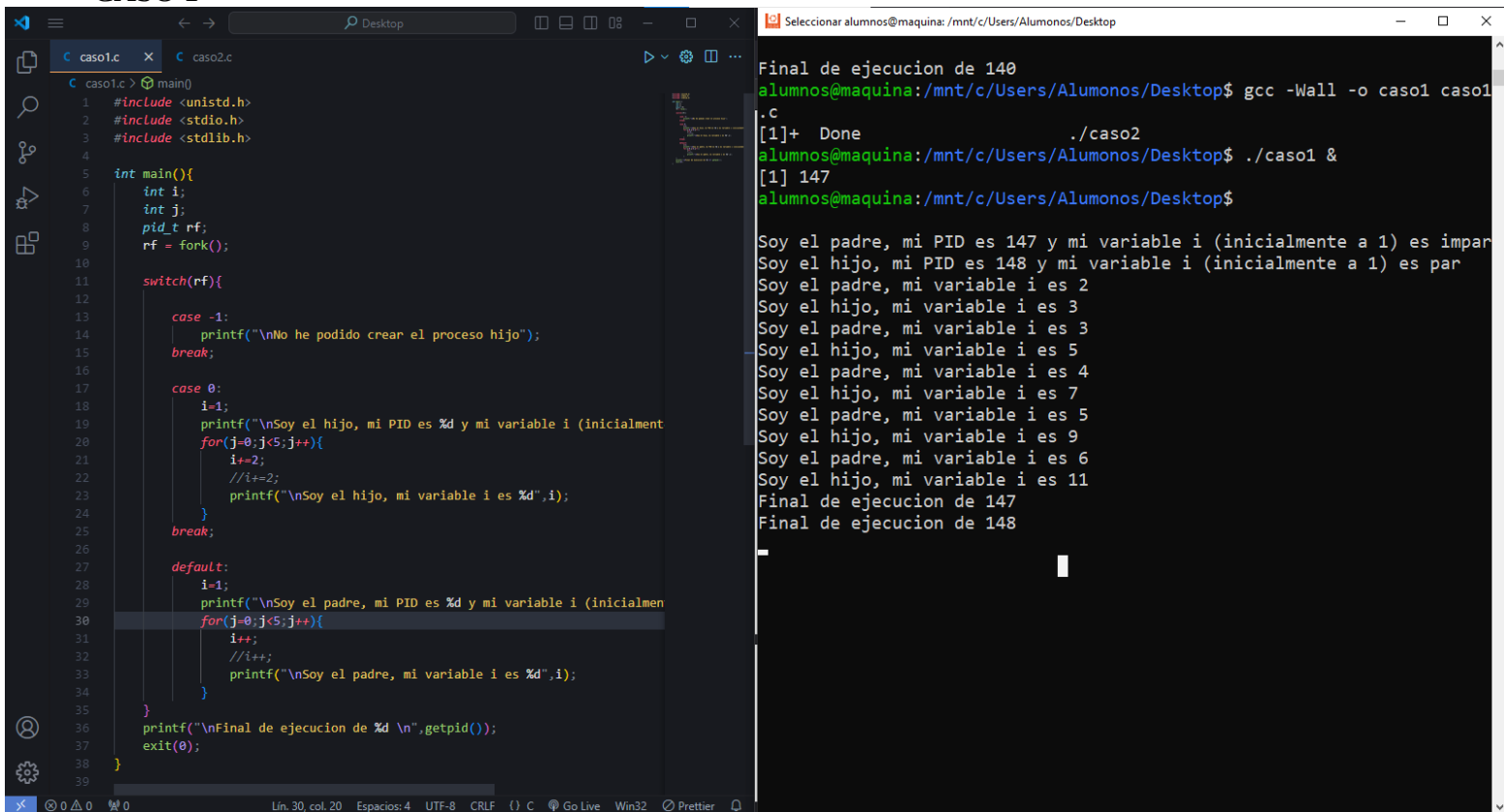


SJ_EXAMEN_SO_LOPEZ_RAMOS_ASAEL_EFRAIN

CASO 1

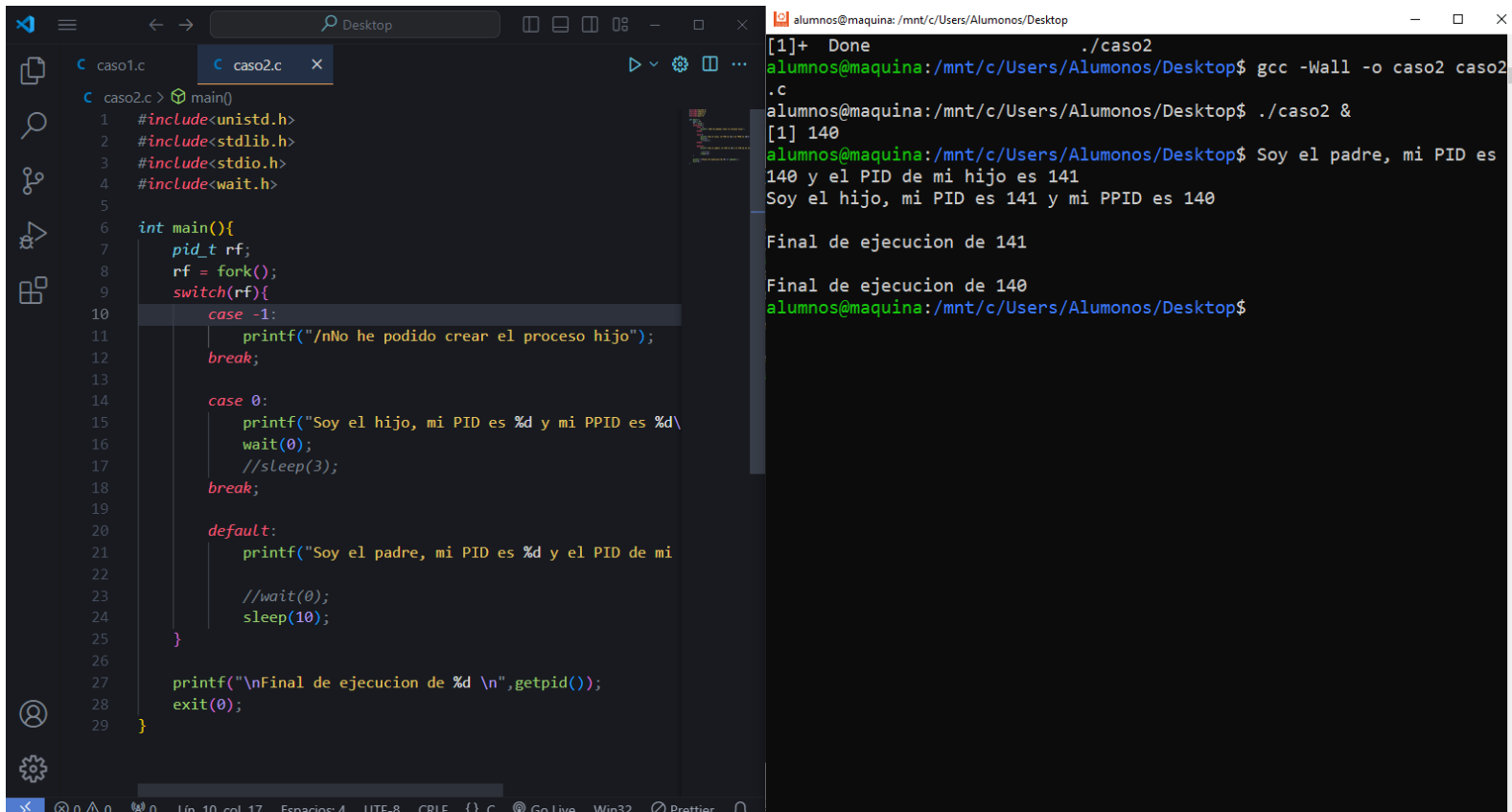


```
1 #include <unistd.h>
2 #include <stdio.h>
3 #include <stdlib.h>
4
5 int main(){
6     int i;
7     int j;
8     pid_t rf;
9     rf = fork();
10
11     switch(rf){
12
13         case -1:
14             printf("\nNo he podido crear el proceso hijo");
15             break;
16
17         case 0:
18             i=1;
19             printf("\nSoy el hijo, mi PID es %d y mi variable i (inicialment
20             for(j=0;j<5;j++){
21                 i+=2;
22                 //i+=2;
23                 printf("\nSoy el hijo, mi variable i es %d",i);
24             }
25             break;
26
27         default:
28             i=1;
29             printf("\nSoy el padre, mi PID es %d y mi variable i (inicialmen
30             for(j=0;j<5;j++){
31                 i++;
32                 //i++;
33                 printf("\nSoy el padre, mi variable i es %d",i);
34             }
35             printf("\nFinal de ejecucion de %d \n",getpid());
36             exit(0);
37         }
38     }
39 }
```

```
Final de ejecucion de 140
alumnos@maquina:/mnt/c/Users/Alumnos/Desktop$ gcc -Wall -o caso1 caso1
.c
[1]+  Done                  ./caso2
alumnos@maquina:/mnt/c/Users/Alumnos/Desktop$ ./caso1 &
[1] 147
alumnos@maquina:/mnt/c/Users/Alumnos/Desktop$

Soy el padre, mi PID es 147 y mi variable i (inicialmente a 1) es impar
Soy el hijo, mi PID es 148 y mi variable i (inicialmente a 1) es par
Soy el padre, mi variable i es 2
Soy el hijo, mi variable i es 3
Soy el padre, mi variable i es 3
Soy el hijo, mi variable i es 5
Soy el padre, mi variable i es 4
Soy el hijo, mi variable i es 7
Soy el padre, mi variable i es 5
Soy el hijo, mi variable i es 9
Soy el padre, mi variable i es 6
Soy el hijo, mi variable i es 11
Final de ejecucion de 147
Final de ejecucion de 148
```

CASO 2



```
1 #include<unistd.h>
2 #include<stdlib.h>
3 #include<stdio.h>
4 #include<wait.h>
5
6 int main(){
7     pid_t rf;
8     rf = fork();
9     switch(rf){
10         case -1:
11             printf("/nNo he podido crear el proceso hijo");
12             break;
13
14         case 0:
15             printf("Soy el hijo, mi PID es %d y mi PPID es %d\n",getpid(),getppid());
16             wait(0);
17             //sleep(3);
18             break;
19
20         default:
21             printf("Soy el padre, mi PID es %d y el PID de mi hijo es %d\n",getpid(),getppid());
22             //wait(0);
23             sleep(10);
24         }
25     }
26
27     printf("\nFinal de ejecucion de %d \n",getpid());
28     exit(0);
29 }
```

```
[1]+  Done                  ./caso2
alumnos@maquina:/mnt/c/Users/Alumnos/Desktop$ gcc -Wall -o caso2 caso2
.c
alumnos@maquina:/mnt/c/Users/Alumnos/Desktop$ ./caso2 &
[1] 140
alumnos@maquina:/mnt/c/Users/Alumnos/Desktop$ Soy el padre, mi PID es
140 y el PID de mi hijo es 141
Soy el hijo, mi PID es 141 y mi PPID es 140

Final de ejecución de 141

Final de ejecucion de 140
alumnos@maquina:/mnt/c/Users/Alumnos/Desktop$
```