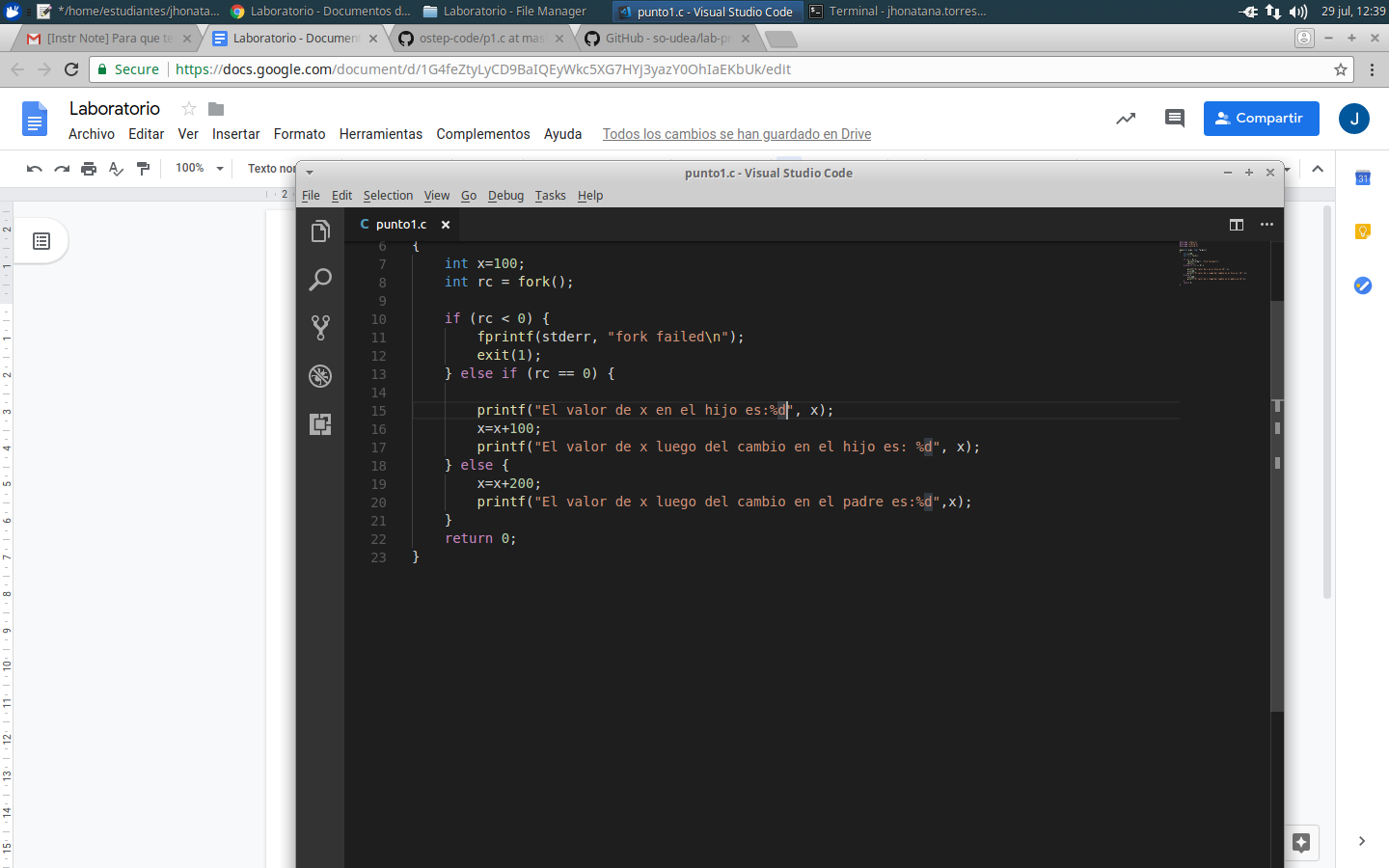
1)a)El valor de la variable del proceso hijo es 100 (valor inicial).

b)Cuando el valor de x es cambiado en los procesos hijo, padre, los procesos identifican qué valor es el que corresponde tanto al padre como al hijo. Ej:

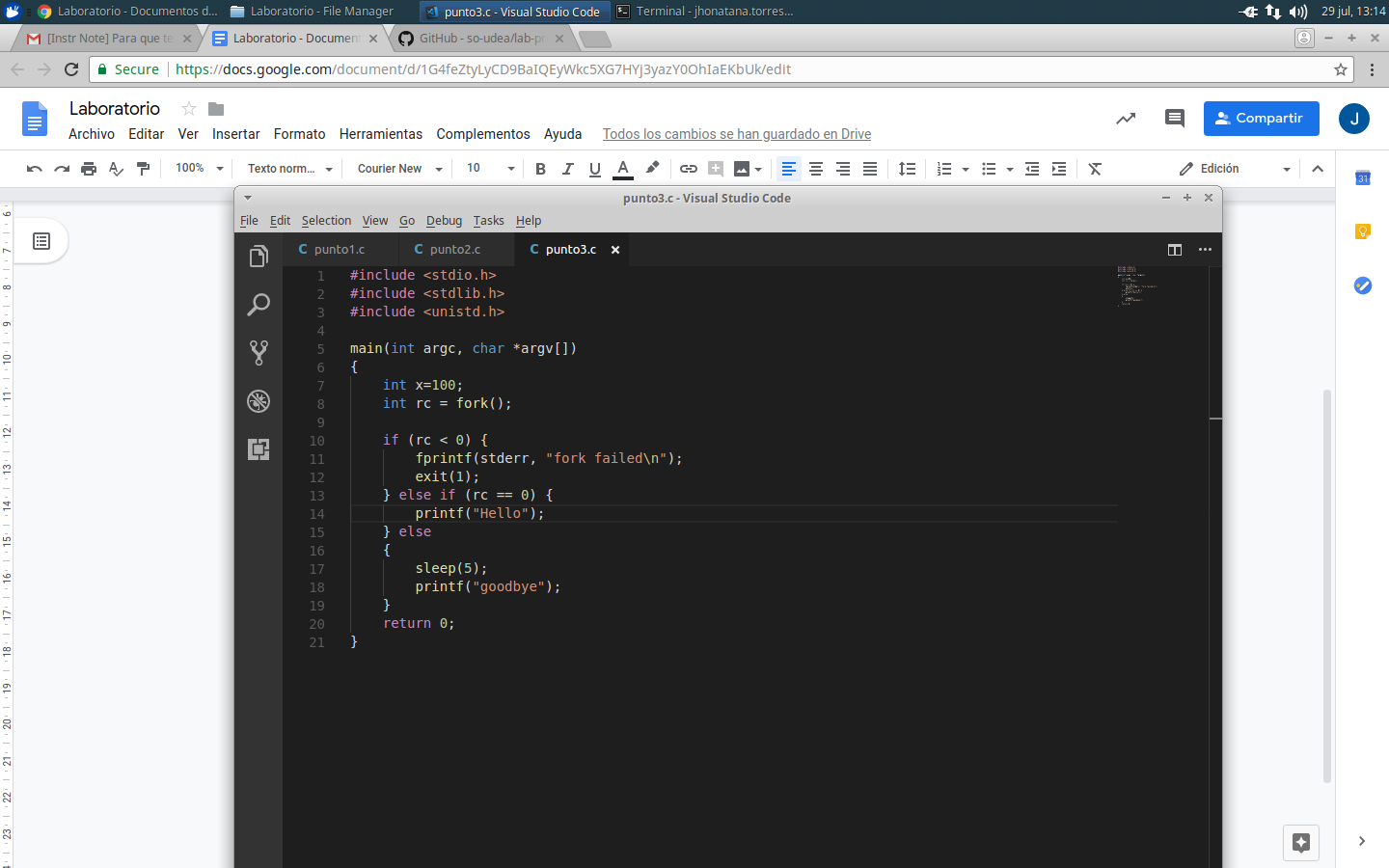


Al cambiar el valor del padre y sumarle 200 el valor de x será 300, mientras que eSn el hijo al sumarle 100 el valor de x será 200.

2)a)Tanto el padre como el hijo pueden acceder al archivo.

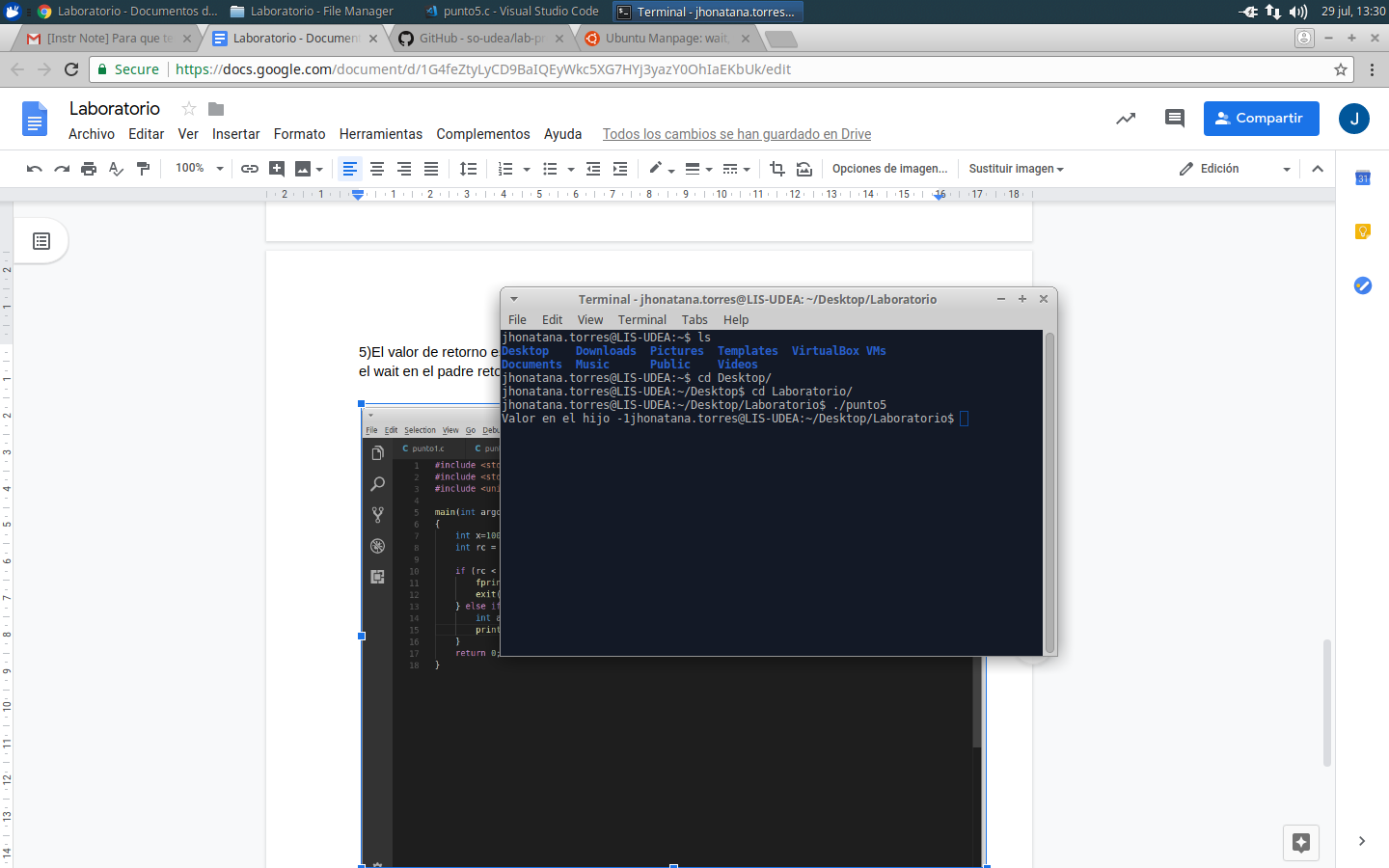
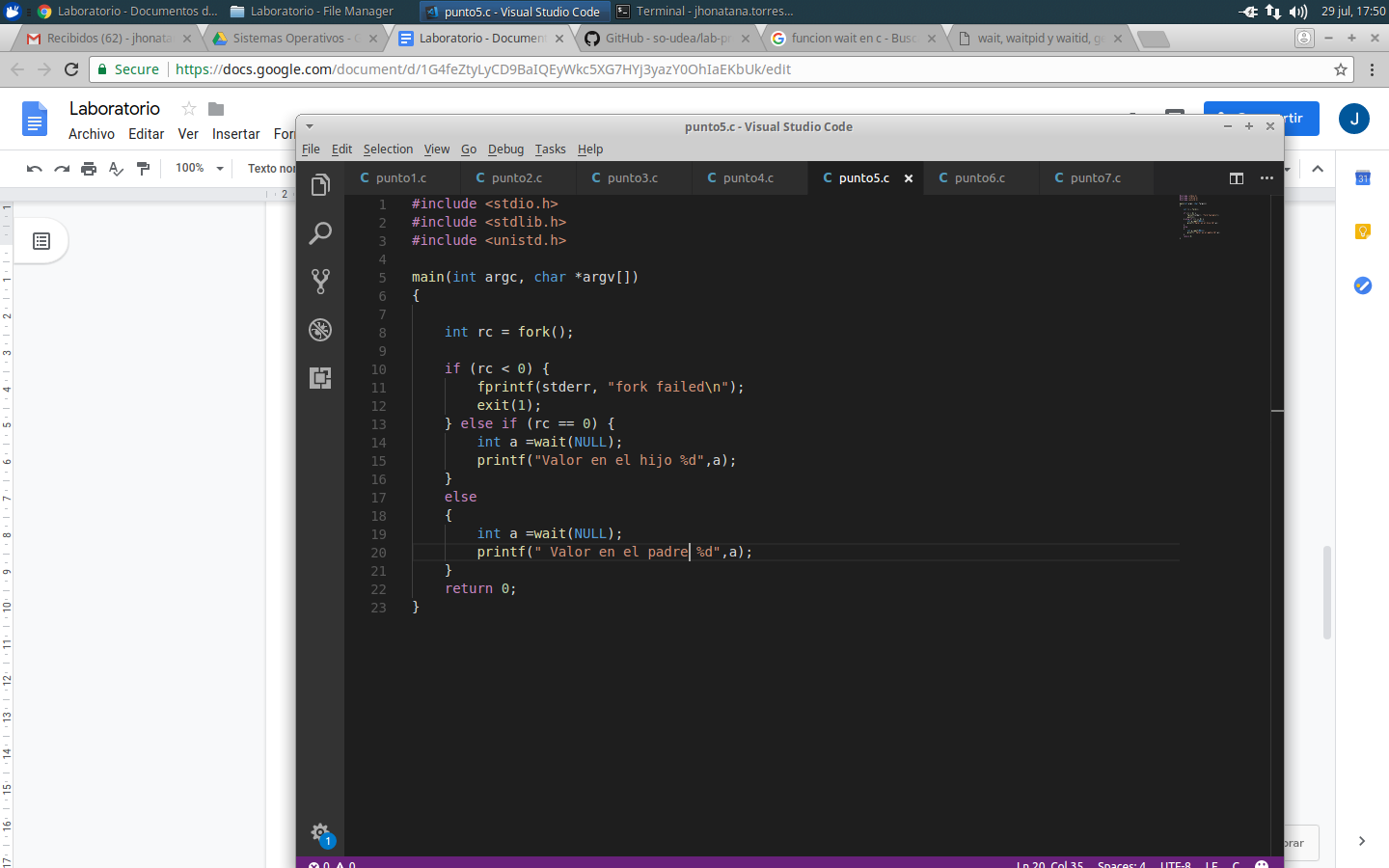
b)No pueden escribir al mismo tiempo, por lo cual el sistema operativo elige quien escribe primero.

3)Si se puede mediante un sleep, ya que el scheduler se encargará de que el proceso hijo se ejecute en el tiempo que el padre duerme. Ej:



4)Existen tantas variaciones de la función exec() para poder tener una variedad de herramientas que permitan resolver diferentes tipos de problemáticas, por ejemplo en los argumentos que se desean enviar o en los valores de retorno de las funciones.

5)El valor de retorno para el wait en el padre es el ID del proceso hijo que terminó su ejecución, en caso de que sea el wait en el hijo retorna -1, wait() espera hasta que uno cualquiera de los hijos del proceso que realiza la llamada a wait cambie de estado.



6)Puede ser útil cuando se quiere cubrir las limitaciones de la función wait, por ejemplo para dejar el proceso padre/hijo en un estado de bloqueo mientras que en el otro se realiza un cálculo para pasarlo como parámetro, si se ha especificado la constante WHOHANG (en el argumento de opción)la función no bloquea y devuelve cero si no hay ningún hijo que haya cambiado de estado.

7)Si el hijo llama a printf luego de cerrar el descriptor no mostrará en la terminar nada de lo que se ingresó en el printf y puede quedar como un proceso zombie.