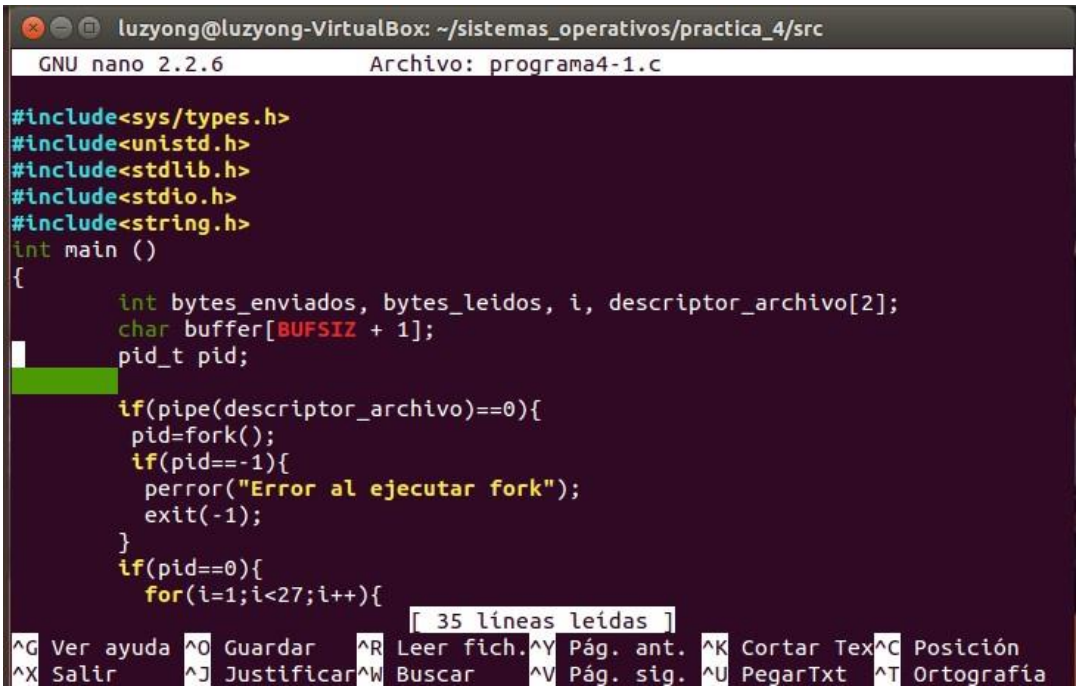
INSTITUTO POLITÉCNICO NACIONAL

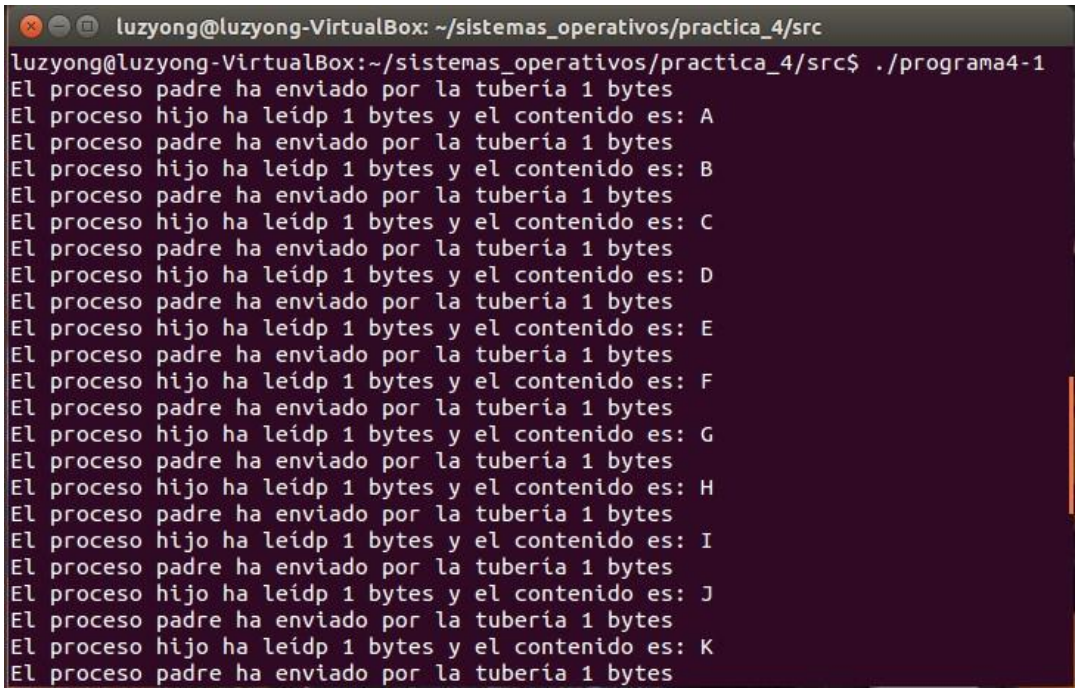
UNIDAD PROFESIONAL INTERDISCIPLINARIA DE INGENIERÍA Y CIENCIAS SOCIALES Y ADMINISTRATIVAS

Práctica 4: Tuberías.

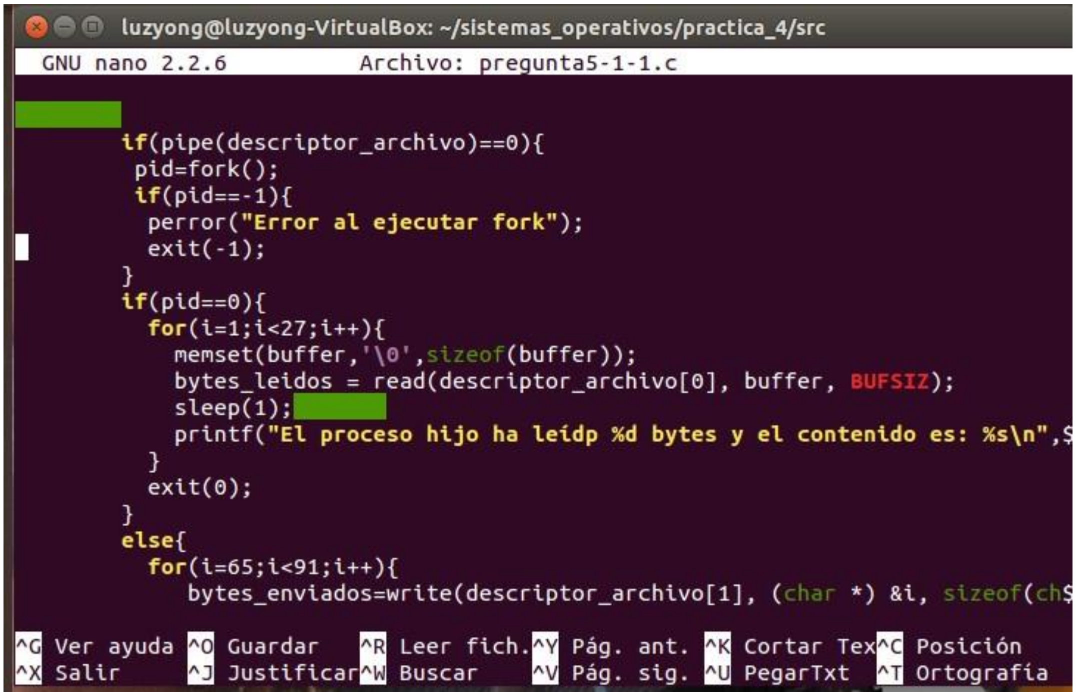
Yong Rodríguez Luz María

Programa 4.1: Crear una tubería entre padre e hijo que envía las letras del abecedario en mayúsculas. El hijo recibe y el padre envía. Los caracteres que se insertan están en su equivalente en código ASCII.

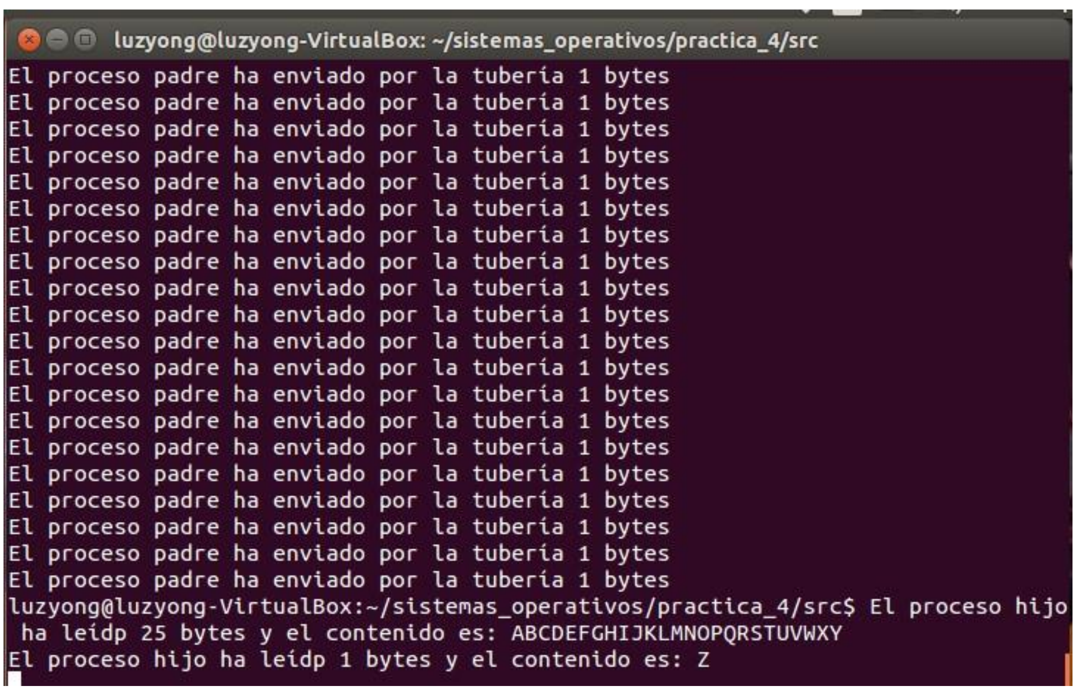




- <u>Pregunta 4-1:</u> ¿Qué cambios se darán en la ejecución si quitamos la función sleep(1), y la colocamos una línea después de la llamada al sistema read?
  Se enviarían primero todos los datos y después imprimiría en pantalla los datos recibidos.

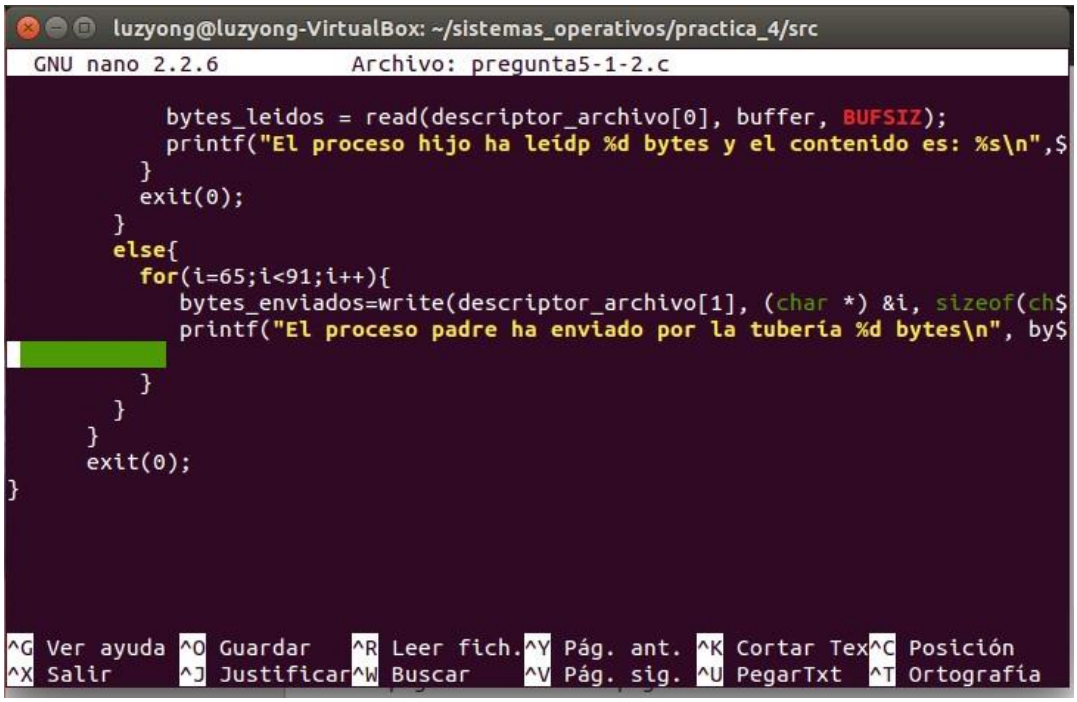


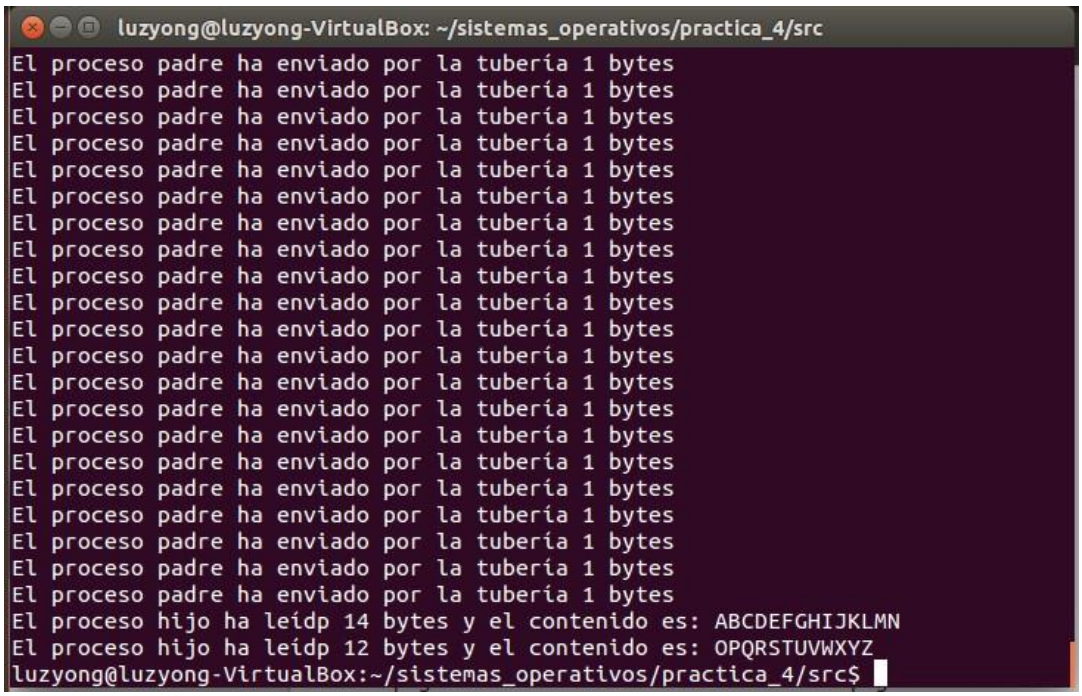
```
luzyong@luzyong-VirtualBox: ~/sistemas_operativos/practica_4/src
GNU nano 2.2.6            Archivo: pregunta5-1-1.c

        if(pipe(descriptor_archivo)==0){
         pid=fork();
         if(pid==-1){
          perror("Error al ejecutar fork");
          exit(-1);
        }
        if(pid==0){
          for(i=1;i<27;i++){
            memset(buffer,'\0',sizeof(buffer));
            bytes_leidos = read(descriptor_archivo[0], buffer, BUFSIZ);
            sleep(1);
            printf("El proceso hijo ha leídp %d bytes y el contenido es: %s\n",$
          }
          exit(0);
        }
        else{
          for(i=65;i<91;i++){
             bytes_enviados=write(descriptor_archivo[1], (char *) &i, sizeof(ch$

^G Ver ayuda ^O Guardar    ^R Leer fich.^Y Pág. ant. ^K Cortar Tex^C Posición
^X Salir     ^J Justificar^W Buscar     ^V Pág. sig. ^U PegarTxt  ^T Ortografía
```



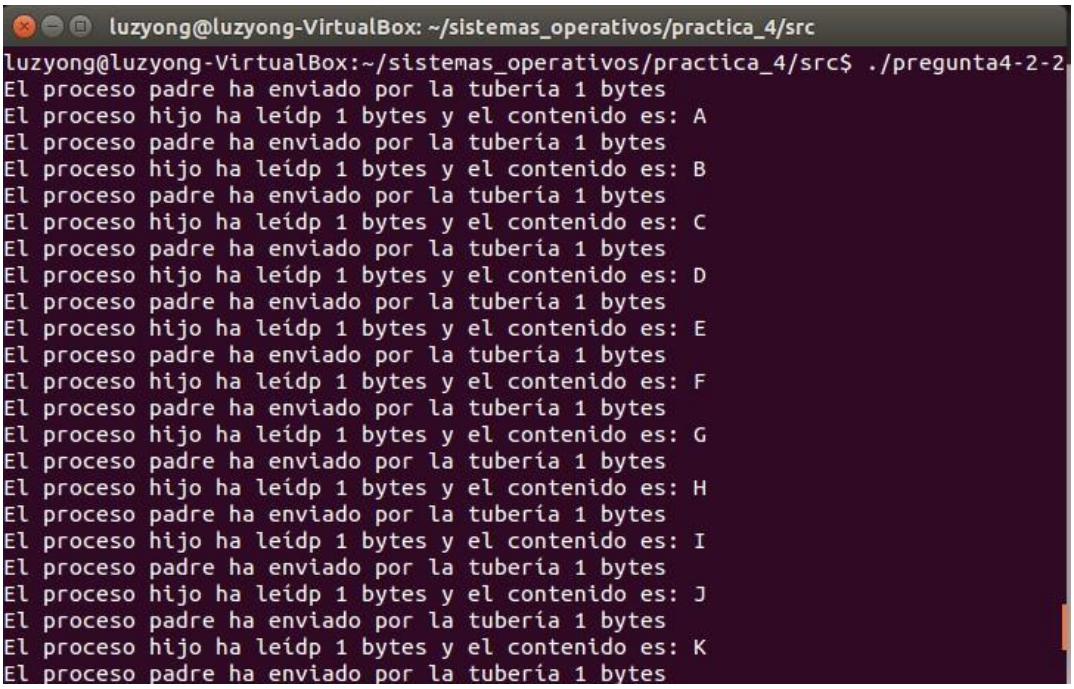
```
luzyong@luzyong-VirtualBox: ~/sistemas_operativos/practica_4/src
El proceso padre ha enviado por la tubería 1 bytes
El proceso padre ha enviado por la tubería 1 bytes
El proceso padre ha enviado por la tubería 1 bytes
El proceso padre ha enviado por la tubería 1 bytes
El proceso padre ha enviado por la tubería 1 bytes
El proceso padre ha enviado por la tubería 1 bytes
El proceso padre ha enviado por la tubería 1 bytes
El proceso padre ha enviado por la tubería 1 bytes
El proceso padre ha enviado por la tubería 1 bytes
El proceso padre ha enviado por la tubería 1 bytes
El proceso padre ha enviado por la tubería 1 bytes
El proceso padre ha enviado por la tubería 1 bytes
El proceso padre ha enviado por la tubería 1 bytes
El proceso padre ha enviado por la tubería 1 bytes
El proceso padre ha enviado por la tubería 1 bytes
El proceso padre ha enviado por la tubería 1 bytes
El proceso padre ha enviado por la tubería 1 bytes
El proceso padre ha enviado por la tubería 1 bytes
El proceso padre ha enviado por la tubería 1 bytes
El proceso padre ha enviado por la tubería 1 bytes
luzyong@luzyong-VirtualBox:~/sistemas_operativos/practica_4/src$ El proceso hijo
 ha leídp 25 bytes y el contenido es: ABCDEFGHIJKLMNOPQRSTUVWXY
El proceso hijo ha leídp 1 bytes y el contenido es: Z
```

¿Qué sucedería si no estuviese la función sleep(1)?
Pasaría lo mismo que en la pregunta anterior, con la diferencia de que los datos se dividen en dos al mostrarlos en pantalla.

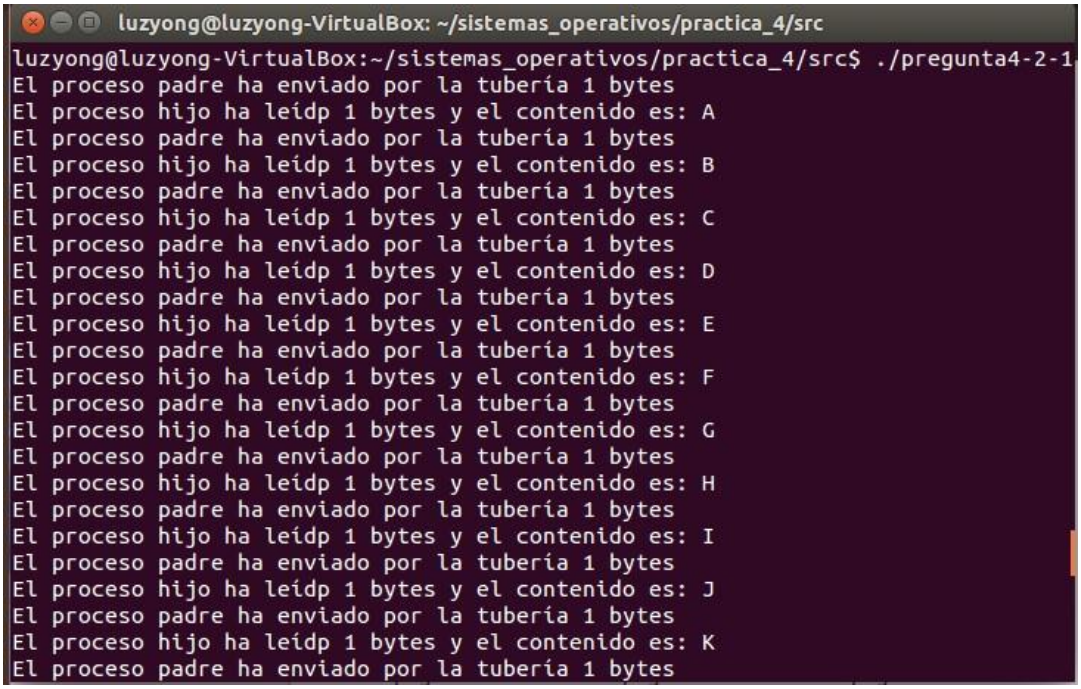




- Pregunta 4-2: ¿Qué sucedería si no existiese la función memset? ¿Y si solo existiese al inicio del código (antes de pipe)?
  La función memset sirve para asignar un carácter determinado en los primeros n espacios de memoria. En el código se le asigna el valor \0 que es como limpiarlo para no causar conflictos. Por eso es que si lo ponemos antes del pipe o lo quitamos del código no pasa nada y el código sigue funcionando.



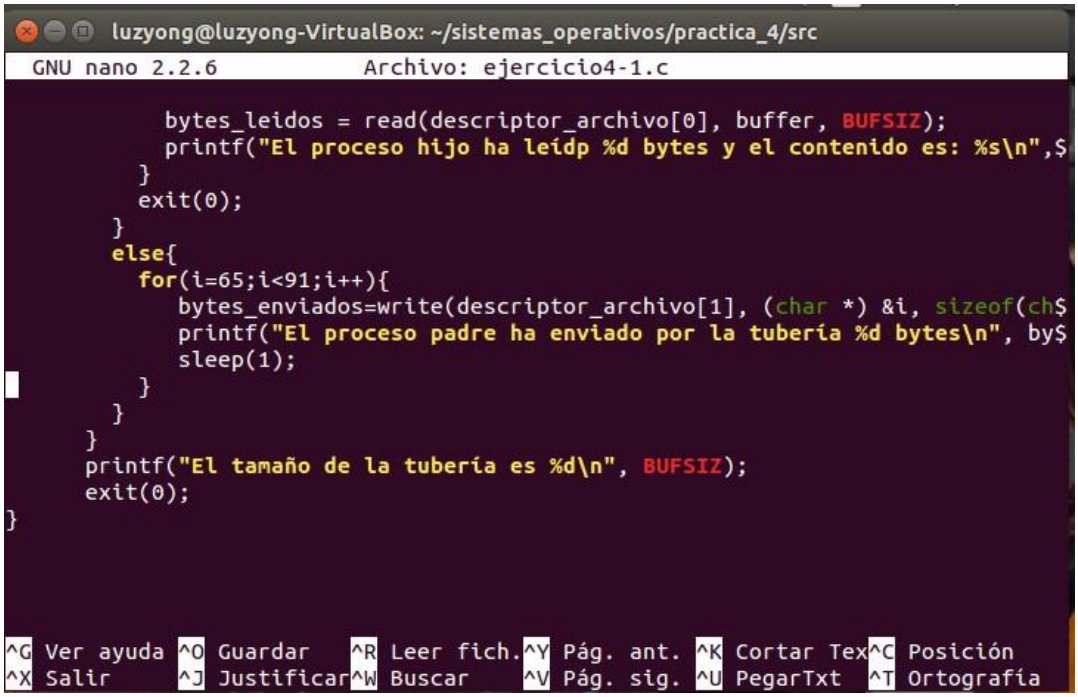
1.-Código con memset antes de pipe



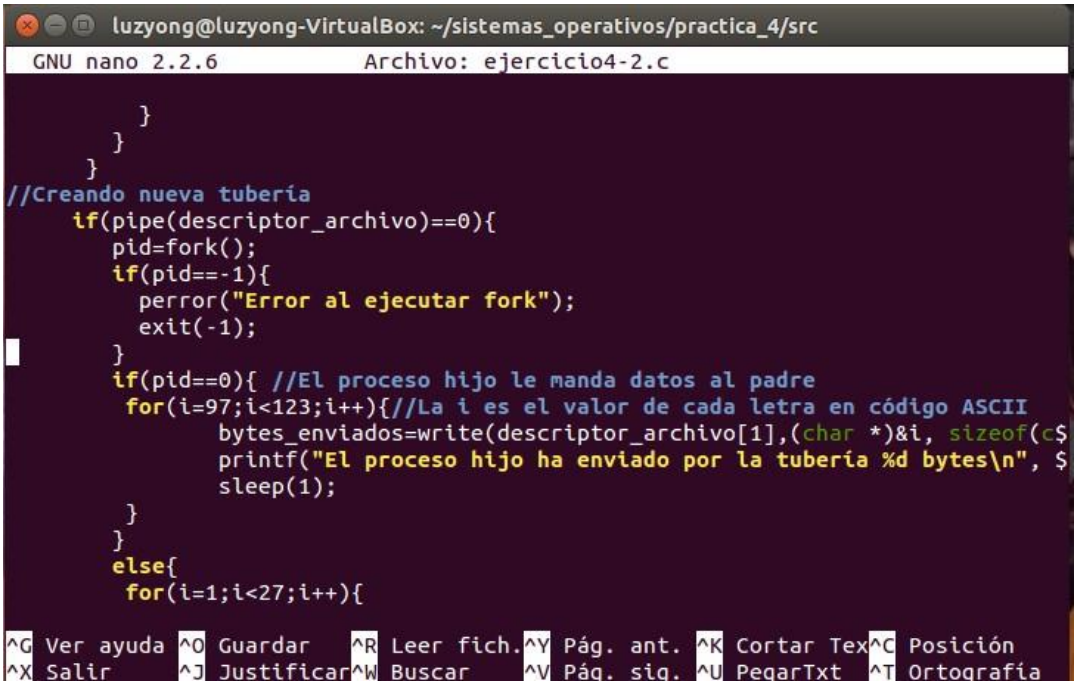
2.-Código sin memset

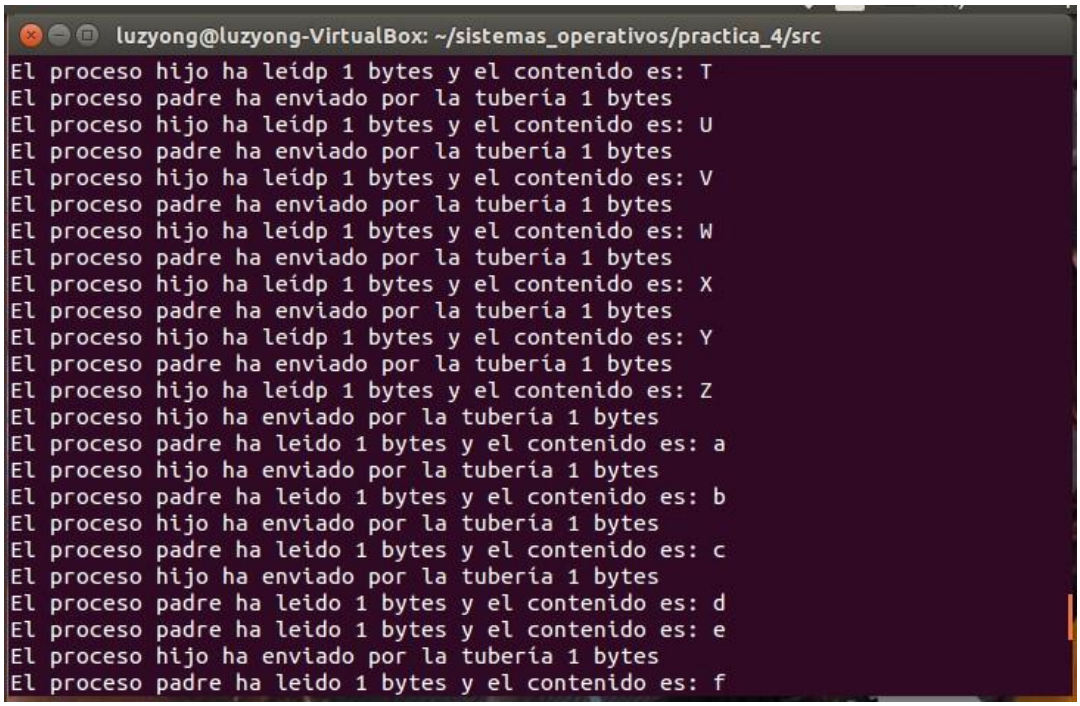
Ejercicio 4-1: Modifique el programa 4-1 para determinar cuál es el tamaño de la tubería en su sistema.

Al principio del código declaramos una variable buffer con tamaño BUSFIZ+1. En la práctica nos menciona que los datos de la tubería se almacenan en un buffer con un tamaño determinado que depende de cada sistema. Para saber el tamaño de nuestra tubería imprimí el valor de BUSFIZ.



Ejercicio 4-2: Modifique el programa 4-1 para que se añada una nueva tubería que pueda enviar datos del proceso hijo al proceso padre. El proceso hijo debe enviar las letras del abecedario, pero en minúsculas. En este código hice otra tubería después de la tubería con mayúsculas, en la que si el valor del pid era igual al proceso hijo, mandara el abecedario en minúsculas por medio del código ASCII.





Conclusiones.

En esta práctica aprendí la función de pipe, cómo se escriben y se leen datos desde archivos y en espacios de memoria y cómo podemos enviar datos entre procesos por medio de las tuberías. Comprendí mejor el concepto de tubería y su funcionamiento.