|  |  |
| --- | --- |
| Arquitectura de Sistemas Operativos | |
| Módulo I | **Aprender el objetivo y las funciones del sistema operativo, la planificación del procesador, la comunicación y sincronización entre procesos.** |
| Unidad 3 | Comunicación y sincronización entre procesos |
|  | Trabajo Práctico Requerido Nº 9 **El problema de los lectores y escritores** |

# Presentación

El objetivo de esta práctica es entender el problema de la comunicación y la necesidad de la sincronización cuando los procesos necesitan comunicarse, para ello pretendemos llegar a la implementación en el Sistema Operativo Linux del problema de estudio que se presenta en el libro de Silberschatz A. y Galvin P. en el capítulo 6 Sincronización de procesos en la página 173, de su 5ta edición de Sistemas Operativos.

La práctica se basa en dos aspectos, primero la identificación de los elementos estudiados en los temas de comunicación y sincronización entre proceso y segundo en la implementación real del problema, para poder verlo en funcionamiento.

En este trabajo el alumno podrá afianzar la teoría estudiada y aplicarla en un sistema operativo real, haciendo uso de herramientas indispensables para un profesional programador, estas herramientas son: los editores de texto, compiladores de lenguaje C, y el uso del intérprete de comandos.

Este trabajo intenta favorecerle el acceso a las siguientes metas de aprendizaje:

Entender los conceptos que se tienen que aplicar en la difícil problemática de la comunicación y sincronización entre procesos.

Comprender el nuevo y maravilloso mundo de la programación concurrente específicamente en la comunicación y sincronización entre procesos.

Comprender el problema de la comunicación y la sincronización y la sección crítica de los procesos.

Le presentamos a continuación, las consignas de trabajo:

# Consignas

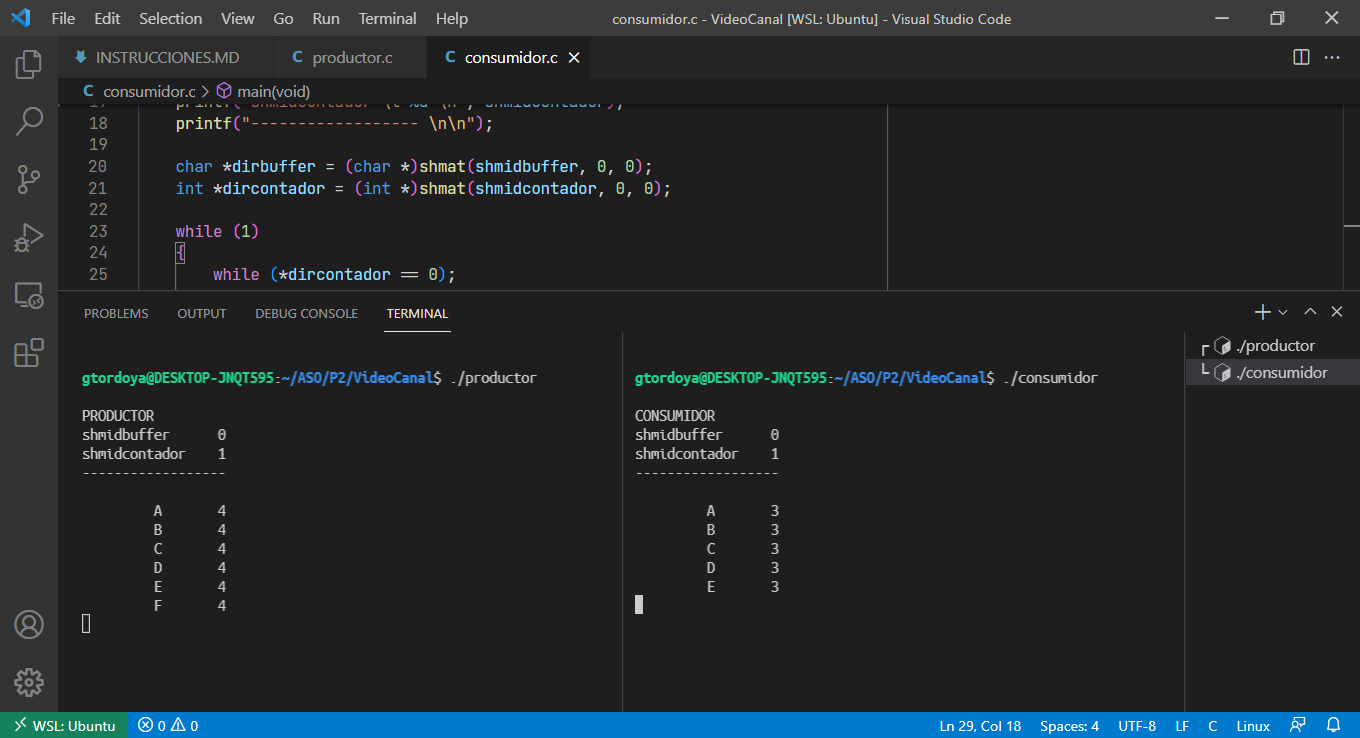
1. El libro de Silberschatz A. y Galvin P. en el capítulo 6 Sincronización de procesos en la página 173, de su 5ta edición de Sistemas Operativos, presenta el problema de los lectores y escritores como un problema clásico de sincronización.
   1. Identificar las secciones críticas.  
      La sección crítica es un segmento de instrucciones que ejecuta un proceso en el cual se accede a datos compartidos (que pueden ser: variables compartidas, archivos compartidos, etc.).  
      La característica importante del sistema es que, cuando un proceso está ejecutando su sección crítica, ningún otro proceso puede ejecutar su correspondiente sección crítica. Es decir, dos procesos no pueden ejecutar su sección crítica al mismo tiempo. El problema de la sección crítica consiste en diseñar un protocolo que los procesos puedan usar para cooperar de esta forma.  
      (En el código adjunto de los ejemplos están señaladas las líneas de código que son el área crítica).
   2. Identificar los mecanismos de comunicación.  
      El comando IPCS de Linux:  
      El comando nos muestra en consola:  
      -) los segmentos de memoria compartida: que es un recurso de comunicación entre procesos.
   3. Identificar los mecanismos de sincronización.  
      El comando IPCS de Linux:  
      El comando nos muestra en consola:  
      -) las matrices semáforo: que es un recurso de sincronización entre procesos.
   4. Asociar los mecanismos de comunicación con segmentos de memoria compartida.  
      El comando IPCS de Linux:  
      El comando nos muestra en consola:  
      -) las colas de mensajes: que es un recurso de comunicación y sincronización entre procesos.
   5. Asociar los mecanismos de sincronización con array de semáforos.  
      Entiendo que la consigna se refiere al problema de interbloqueo, que en el libro se ilustra con lo de la cena de los filósofos. En términos de código, así se haría la asociación de mecanismos de sincronización con array de semáforos:  
        
      do {  
       wait(palillofi]);  
       wait(palillo [ (i + 1)% 51);  
        
       // comer  
        
       signal (palillo [i] ) ;   
       signal(palillo [(i+1) % 5]);  
        
       // pensar  
        
      }while (TRUE);
   6. Definir los procesos lectores y escritores.  
      Imagine una base de datos que debe ser compartida por varios procesos concurrentes. Algunos de estos procesos pueden simplemente querer leer la base de datos, mientras que otros pueden desear actualizarla (es decir, leer y escribir). Diferenciamos entre estos dos tipos de procesos denominando a los primeros lectores y a los otros escritores. Obviamente, si dos lectores acceden simultáneamente a los datos compartidos, esto no dará lugar a ningún problema. Sin embargo, si un escritor y algún otro proceso (sea lector o escritor) acceden simultáneamente a la base de datos, el caos está asegurado. Para asegurar que estos problemas no afloren, requerimos que los procesos escritores tengan acceso exclusivo a la base de datos compartida. Este problema de sincronización se denomina problema de los lectores y escritores.
   7. Implementar el problema en un entorno operativo LINUX. Crear los segmentos de memoria compartida. Crear e inicializar los semáforos.

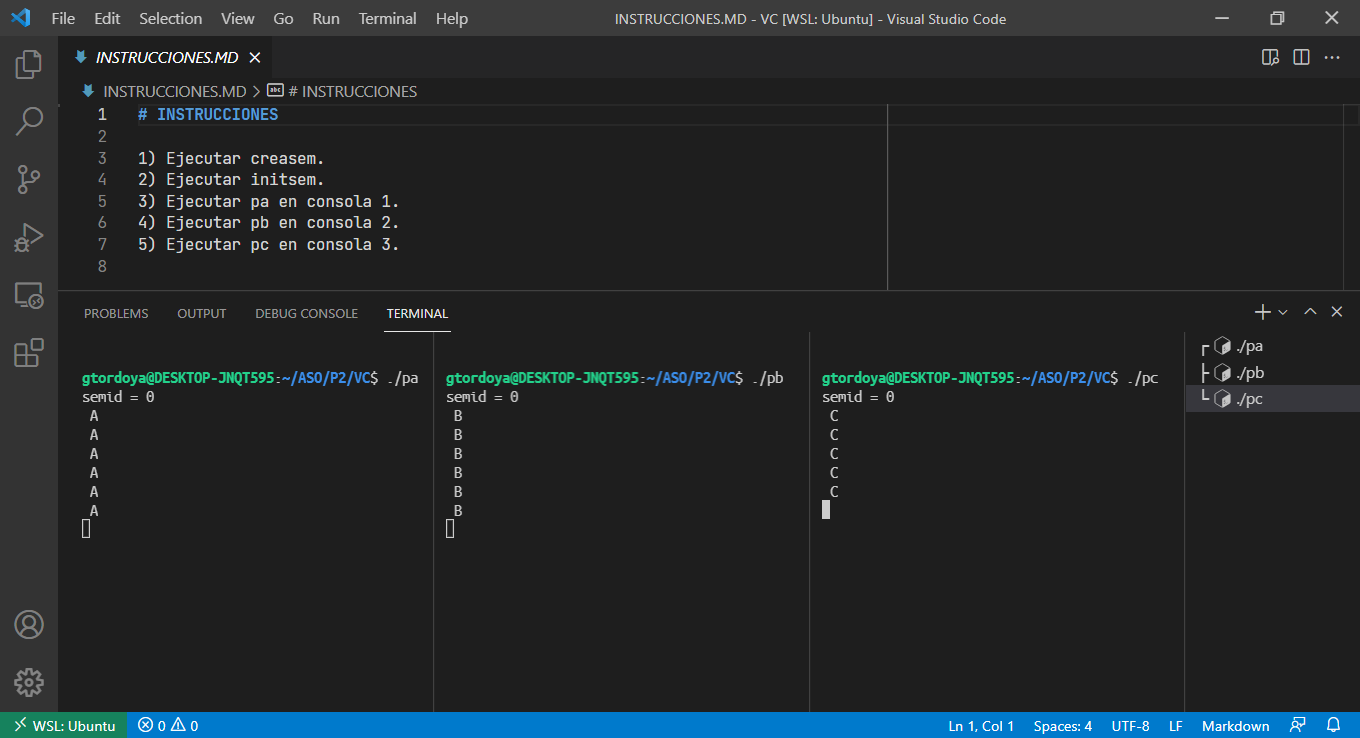
Codificar los procesos lectores y escritores. Ejecutar los procesos.

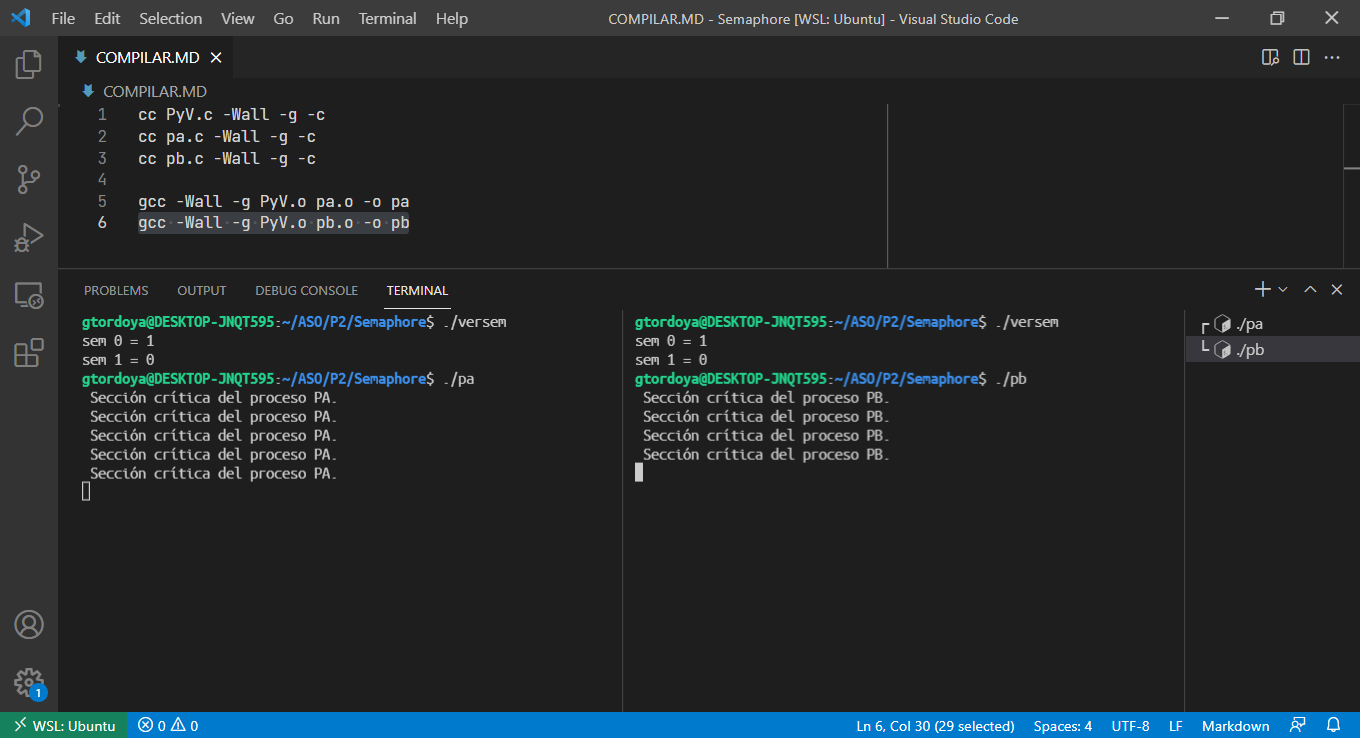
Realizar un informe de lo observado.

Todos los incisos: código adjunto (3 ejemplos).

Observado:







# Criterios de evaluación

En la corrección de este trabajo se evaluará la presentación del trabajo en tiempo y forma, las conclusiones obtenidas y el funcionamiento de los ejercicios en el entorno operativo LINUX.

**GERARDO TORDOYA**

**COMISIÓN 2J (Analista Programador a Distancia)**