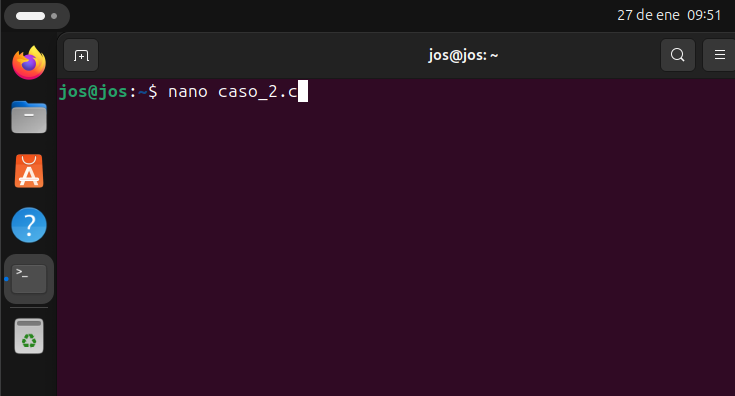
**Nombre:** Josué Merino

**NRC:** 2537

**Examen Práctico 2**

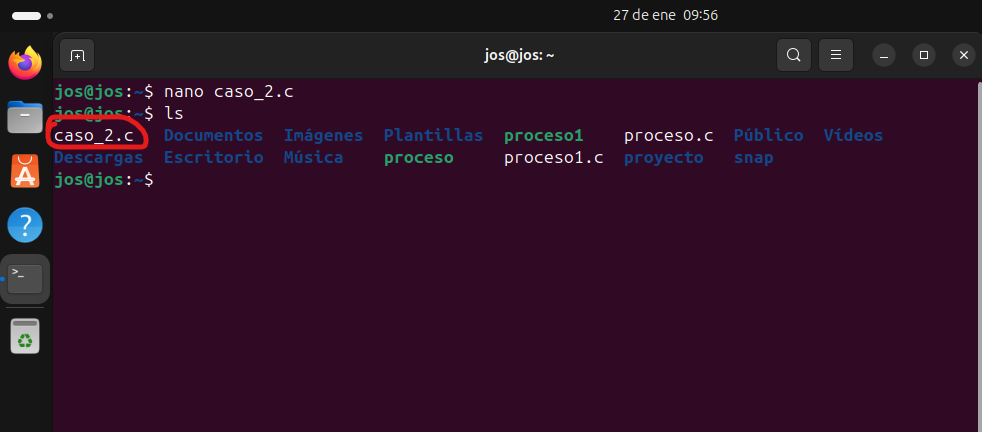
Se crea un archivo .c para introducir el código, mediante el siguiente comando:



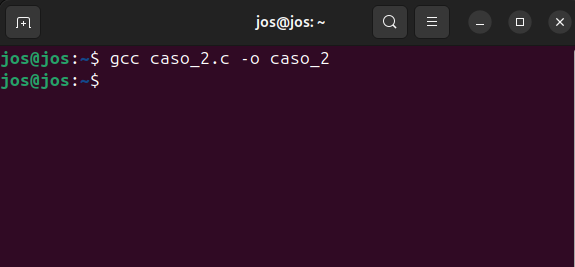
Se crea el archivo .c con el nombre **caso\_2.c**, se digita el siguiente código y se lo guarda:

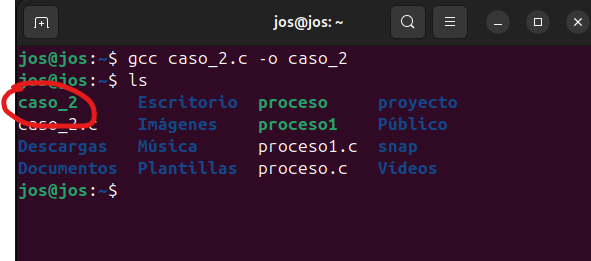


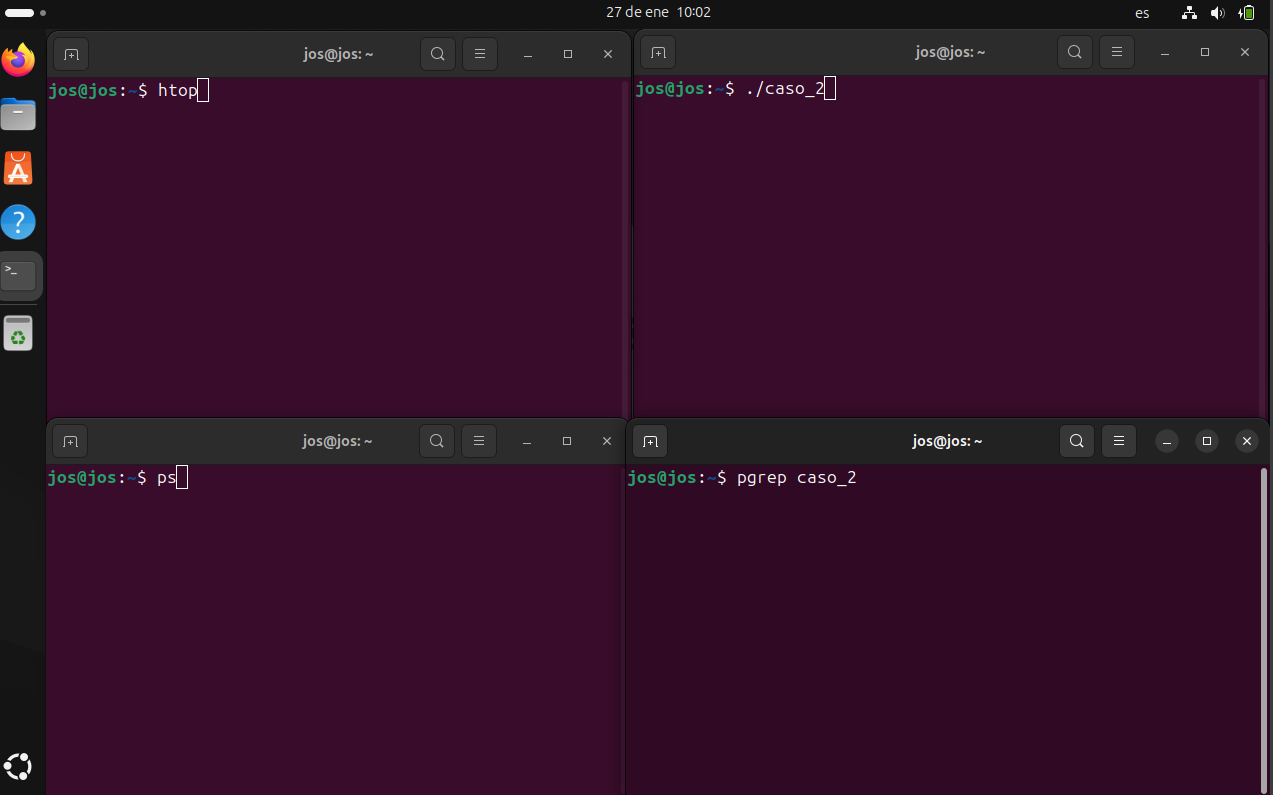
Se confirma que el archivo se creó:



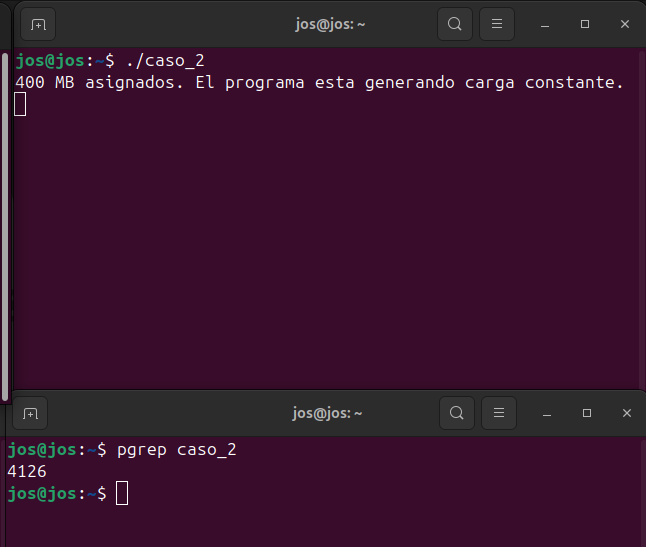
Se procede a la ejecución del archivo:



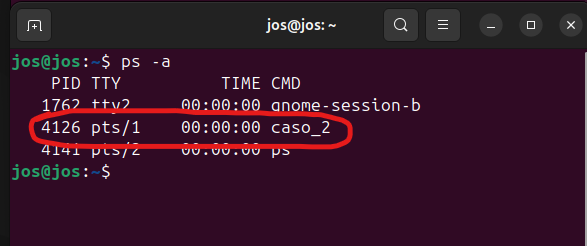


Para el siguiente paso el entorno de interfaces de líneas de comandos que se manejará es el siguiente:  


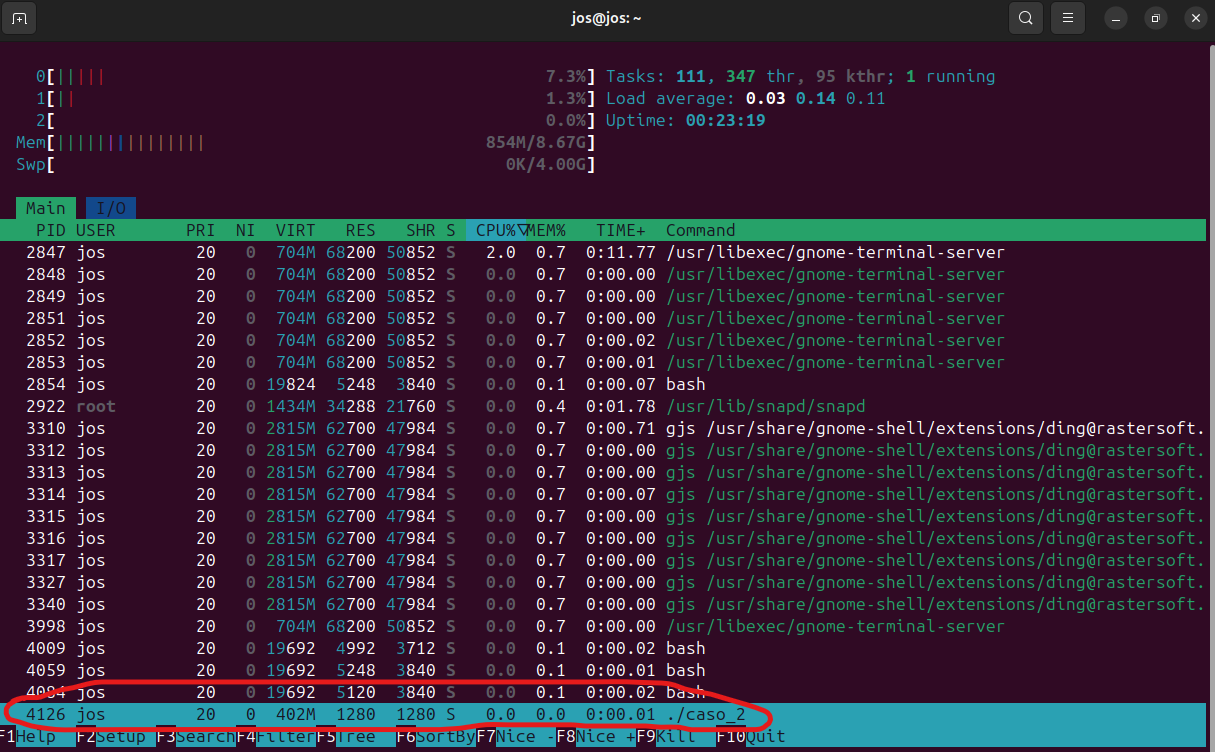
Se procede a ejecutar con: ./caso\_2:



Se verifica el proceso utilizando el comando ps -a:



Mediante el htop visualizamos los recursos que consume el proceso caso\_2:



**Explicación y Resumen**

El caso de estudio #2 sugiere una visualización de un proceso que asigna 400 MB de memoria y ejecuta un bucle infinito para simular una carga mediante el comando **htop** constante en el sistema.

Para empezar se realiza la creación del archivo en **lenguaje C** para posteriormente ejecutar y visualizar el proceso. La creación se la realizó con el comando: **nano caso\_2.c** . Dentro de este archivo se procede a digitar el código que utiliza librerías como: stdlib, unistd y stdio.

Una vez finalizada la codificación se ejecuta el archivo mediante el comando **gcc caso\_2.c –o caso\_2;** posterior a esto se diseña el entorno de interfaz de líneas de comandos que facilitará la visualización de la práctica. El entorno consiste en cuatro interfaces en las cuales cada una ejecutará un comando/instrucción diferente.

En la segunda interfaz se ejecuta el proceso, con: **./caso\_2**

Una vez realizado esto, en la tercera interfaz se ejecuta: **ps –a,** en la que se visualiza el PID del proceso caso\_2, en este caso: 4126.

En la cuarta interfaz se ejecuta el comando: **pgrep caso\_2,** donde aparece específicamente el PID del caso\_2 (caso contrario a ps –a donde se listaban todos los procesos en ejecución)

Finalmente en la primera interfaz se ejecuta el comando **htop** para monitorear visualmente el proceso, encontrándose este al final de la interfaz junto con sus características y consumos.

**Conclusión**

Se ha finalizado la práctica, obteniendo los resultados esperados correctamente. Se ha verificado que el proceso que asigna 400 MB de memoria y ejecuta un bucle infinito para simular una carga constante en el sistema dentro de la terminal visual (generada por el comando **htop**) ocupa 402MB de memoria, 2 MB más de la asignada dentro del programa realizado en C.

Esta práctica no solo familiariza al estudiante a retener comandos básicos de Linux como: nano, htop, ls, ps –a, pgrep, etc. Y por otro lado se visualiza el consumo de memoria que tiene un proceso y como se visualiza en la terminal **htop** en Linux. Entendiendo de esta manera la forma en la que funciona la gestión de memoria en un sistema operativo.