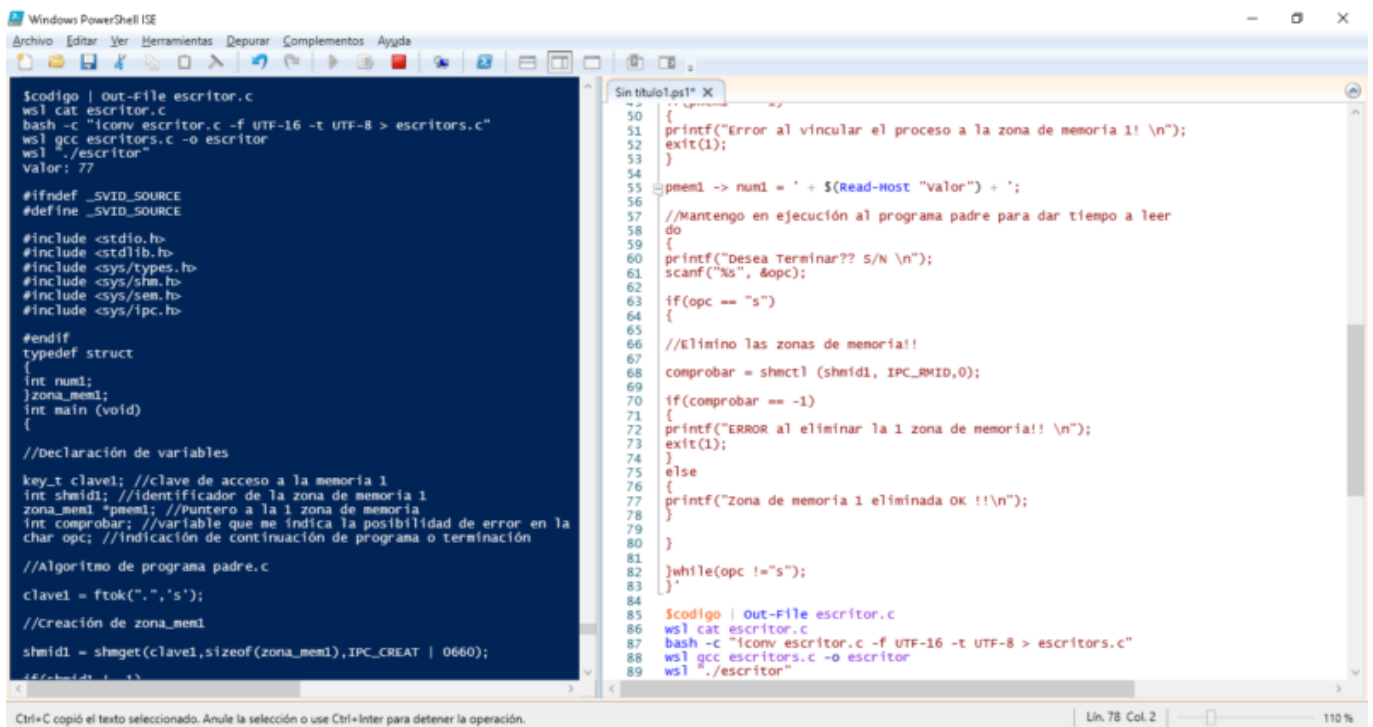


Utilizar zonas de memoria compartida en Linux mediante WSL desde PowerShell

Programa que crea la zona de memoria compartida y almacena un valor

[crayon-6a9d602e338c8630681697/]



```
$codigo | Out-File escritor.c
wsl cat escritor.c
bash -c "iconv escritor.c -f UTF-16 -t UTF-8 > escritores.c"
wsl gcc escritores.c -o escritor
wsl ./escritor
valor: 77

#ifndef _SVID_SOURCE
#define _SVID_SOURCE

#include <stdio.h>
#include <stdlib.h>
#include <sys/types.h>
#include <sys/shm.h>
#include <sys/sem.h>
#include <sys/ipc.h>

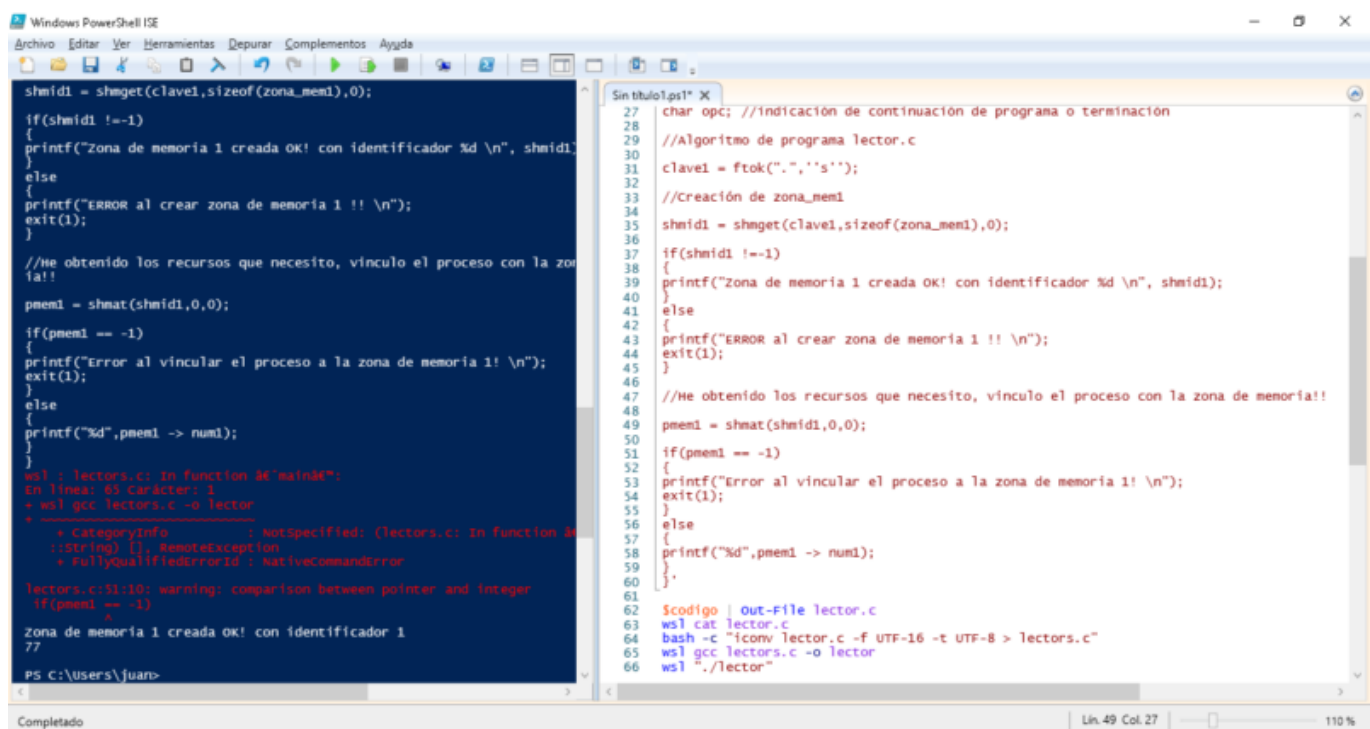
#endif
typedef struct
{
    int num1;
} zona_mem1;
int main (void)
{
    //Declaración de variables
    key_t clave1; //clave de acceso a la memoria 1
    int shm1d1; //identificador de la zona de memoria 1
    zona_mem1 *pmem1; //Puntero a la 1 zona de memoria
    int comprobar; //variable que me indica la posibilidad de error en la
    char opc; //Indicación de continuación de programa o terminación

    //Algoritmo de programa padre.c
    clave1 = ftok(".", 's');
    //Creación de zona_mem1
    shm1d1 = shmget(clave1, sizeof(zona_mem1), IPC_CREAT | 0660);
    if (shm1d1 < 0)
    {
        printf("Error al vincular el proceso a la zona de memoria 1! \n");
        exit(1);
    }
    pmem1 -> num1 = ' + $(Read-Host "valor") + ' ;
    //Mantengo en ejecución al programa padre para dar tiempo a leer
    do
    {
        printf("Desea Terminar?? S/N \n");
        scanf("%s", &opc);
        if (opc == "s")
        {
            //Elimino las zonas de memoria!!
            comprobar = shmctl (shm1d1, IPC_RMID, 0);
            if (comprobar == -1)
            {
                printf("ERROR al eliminar la 1 zona de memoria!! \n");
                exit(1);
            }
            else
            {
                printf("Zona de memoria 1 eliminada OK !!\n");
            }
        }
    } while (opc != "s");
}
```

Zona de memoria creada

```
ubuntu@lam54: ~  
ubuntu@lam54:~$ ipcs -m  
  
----- Shared Memory Segments -----  
key          shmid      owner        perms        bytes       nattch       status  
0x780c3f53  0             ubuntu       660          4           0  
  
ubuntu@lam54:~$
```

Programa que lee un valor en la zona de memoria compartida
[crayon-6a9d602e338d0629360010/]



```
Windows PowerShell ISE  
Archivo Editar Ver Herramientas Depurar Complementos Ayuda  
  
shmid1 = shmget(clave1, sizeof(zona_mem1), 0);  
if(shmid1 != -1)  
{  
    printf("Zona de memoria 1 creada OK! con identificador %d \n", shmid1);  
}  
else  
{  
    printf("ERROR al crear zona de memoria 1 !! \n");  
    exit(1);  
}  
  
//He obtenido los recursos que necesito, vinculo el proceso con la zona de memoria!!  
  
pmem1 = shmat(shmid1, 0, 0);  
if(pmem1 == -1)  
{  
    printf("Error al vincular el proceso a la zona de memoria 1! \n");  
    exit(1);  
}  
else  
{  
    printf("%d", pmem1 -> num1);  
}  
}  
  
wsl : lectores.c: in function 'main':  
En línea: 63 Carácter: 1  
+ wsl gcc lectores.c -o lector  
+ ~~~~~  
+ CategoryInfo          : NotSpecified: (lectores.c: in function 'main')  
+ FullyQualifiedErrorId : NativeCommandError  
  
lectores.c:51:10: warning: comparison between pointer and integer  
if(pmem1 == -1)  
    ^  
Zona de memoria 1 creada OK! con identificador 1  
77  
PS C:\Users\juar>  
  
Sin título1.ps1 X  
27 char opc; //Indicación de continuación de programa o terminación  
28  
29 //Algoritmo de programa lector.c  
30  
31 clave1 = ftok(".", 's');  
32  
33 //Creación de zona_mem1  
34  
35 shmid1 = shmget(clave1, sizeof(zona_mem1), 0);  
36  
37 if(shmid1 != -1)  
38 {  
39     printf("Zona de memoria 1 creada OK! con identificador %d \n", shmid1);  
40 }  
41 else  
42 {  
43     printf("ERROR al crear zona de memoria 1 !! \n");  
44     exit(1);  
45 }  
46  
47 //He obtenido los recursos que necesito, vinculo el proceso con la zona de memoria!!  
48  
49 pmem1 = shmat(shmid1, 0, 0);  
50  
51 if(pmem1 == -1)  
52 {  
53     printf("Error al vincular el proceso a la zona de memoria 1! \n");  
54     exit(1);  
55 }  
56 else  
57 {  
58     printf("%d", pmem1 -> num1);  
59 }  
60 }  
61  
62 $codigo | Out-File lector.c  
63 wsl cat lector.c  
64 bash -c "iconv lector.c -f UTF-16 -t UTF-8 > lectores.c"  
65 wsl gcc lectores.c -o lector  
66 wsl "./lector"
```