Uso de las funciones ftok(), shmget(), shmat() y shmctl()

Una zona de memoria puede ser compartida por más de un proceso. Esta es la forma de comunicación más rápida entre procesos. La memoria compartida se crea por un proceso mediante una llamada al sistema, la zona que se reserva en memoria no está en el espacio de direcciones del proceso, es una zona de memoria gestionada por el sistema operativo. Después, otros procesos a los que se les dé permiso para acceder a esa zona de memoria, podrán también leer o escribir de ella.

Para utilizar las memorias compartidas existen una serie de llamadas al sistema. Algunas de las funciones útiles para la práctica son:
[crayon-6a9d6060b799a847325498/]
La llamada a esta función genera una variable de tipo key_t, que se corresponde con una clave que tendrá siempre el mismo valor cuando se reciben los mismos argumentos. Esta clave será empleada después en la llamada a la función shmget.

Utilización:

- Primer argumento: pathname – nombre de un fichero existente y accesible.
- Segundo argumento: proj_id – variable entera que no debe ser 0.

[crayon-6a9d6060b79a2166847972/]
Se emplea en la creación o acceso a una zona de memoria compartida

Retorno:

- Identificador de la memoria compartida si no ha habido

error
- -1 si ha habido error

Utilización:

- Primer argumento: key – es una clave que genera el sistema y que será un elemento esencial para poder acceder a la zona de memoria que crearemos. La clave se obtiene previamente utilizando la función ftok(); que produce siempre una clave fija con los mismos argumentos: Para usarla:
  key = ftok(«.», 'S');
  La clave debe ser la misma para todos los procesos que quieran usar esta zona de memoria.
- Segundo argumento: size – indicará el tamaño de la memoria compartida. Se suele utilizar la opción de sizeof(tipo_variable) para que se tome el tamaño del tipo de dato que habrá en la memoria compartida.
- Tercer argumento: indicará los derechos para acceder a la memoria, si se crea si no existe y caso de que exista, se da o no un error.
  Para crear la zona de memoria usaremos IPC_CREAT | 0660 (ese número indica ciertos permisos de lectura y escritura).
  Cuando se esté utilizando esta llamada al sistema para acceder a una memoria que ya se ha creado previamente, este argumento tomará el valor 0.

[crayon-6a9d6060b79a5968460330/]
Para que un proceso pueda acceder a esa memoria compartida previamente creada, será necesario que alguna variable del proceso «apunte» a esa zona de memoria que no pertenece a su espacio de direccionamiento. Para ello, se utiliza esta llamada al sistema que permite vincular esa zona de memoria al direccionamiento lógico del proceso.

Retorno:

- Puntero a la zona de memoria compartida
- -1 si ha habido error

Utilización:

- Primer argumento: shmid – identificador de la memoria compartida. Lo habremos obtenido con la llamada shmget.
- Segundo argumento shmaddr: normalmente, valdrá 0 o NULL que indicará al sistema operativo que busque esa zona de memoria en una zona de memoria libre, no en una dirección absoluta
- Tercer argumento shmflg: se suele poner también 0 en este campo, que corresponde a los flags.

[crayon-6a9d6060b79a7461965356/]
Llamada al sistema para el control y gestión de la zona de memoria compartida (en particular, se va a usar para liberar la memoria compartida).

Retorno:

- 0 si ha sido un éxito
- -1 si ha habido error

Utilización:

- Primer argumento: shmid – identificador de la memoria compartida. Lo habremos obtenido con la llamada shmget.
- Segundo argumento: existen varias opciones que indicarán distintos comandos para realizar. Nosotros vamos a utilizar la opción IPC_RMID que marca esa zona de memoria para ser liberada cuando ya no haya ningún proceso vinculado a ella.
- Tercer argumento: buf: en el caso en que se va a utilizar esta llamada al sistema, valdrá 0.

Para poder utilizar estas funciones es necesario incluir:
[crayon-6a9d6060b79aa127373317/]