Crear un proceso hijo mediante la función fork de Linux (en Bash y en PowerShell mediante WSL)

Código para crear un proceso hijo mediante la función fork

```
#include <stdio.h>
int main() {
        int pid;
        pid = fork();

        // En cuanto llamamos a fork se crea un nuevo proceso. En el proceso
        // padre pid contendrá el pid del proceso hijo. En el proceso hijo
        // pid valdrá 0. Eso es lo que usamos para distinguir si el código
         // que se está ejecutando pertenece al padre o al hijo.

        if (pid) // Este es el proceso padre
        {
                printf("PADRE: Mi padre tiene el pid: %d\n", getppid());
                printf("PADRE: Soy el proceso padre y mi pid es: %d\n", getpid());
                 printf("PADRE: Mi hijo tiene el pid: %d\n", pid);
        }
        else // Proceso hijo
        {
                printf("HIJO: Soy el proceso hijo y mi pid es: %d\n", getpid());
```

```
                printf("HIJO: Mi padre tiene el pid: %d\n",
getppid());
        }
}
```

Código para PowerShell

```
$codigo = '
#include <stdio.h>
int main() {
        int pid;
        pid = fork();

        // En cuanto llamamos a fork se crea un nuevo proceso.
En el proceso
        // padre pid contendrá el pid del proceso hijo. En el
proceso hijo
        // pid valdrá 0. Eso es lo que usamos para distinguir
si el código
         // que se está ejecutando pertenece al padre o al
hijo.

        if (pid) // Este es el proceso padre
        {
                printf("PADRE: Mi padre tiene el pid: %d\n",
getppid());
                printf("PADRE: Soy el proceso padre y mi pid
es: %d\n", getpid());
                 printf("PADRE: Mi hijo tiene el pid: %d\n",
pid);
        }
        else // Proceso hijo
        {
               printf("HIJO: Soy el proceso hijo y mi pid es:
%d\n", getpid());
                 printf("HIJO: Mi padre tiene el pid: %d\n",
getppid());
        }
}
'
```

```
$codigo | Out-File fork.c
bash -c "iconv fork.c -f UTF-16 -t UTF-8 > forks.c"
wsl gcc forks.c
bash -c "./a.out"
```

Código ejecutado en PowerShell mediante WSL (el resultado es un poco extraño a la hora de obtener los identificadores de proceso tanto del proceso padre como del proceso hijo)

Código ejecutado en Bash (los identificadores de proceso del padre y del hijo son normales)