

Punto 15 (Prerrequisito)

15. El siguiente programa recibe como parámetro de entrada un número entero y muestra como resultado su factorial.

```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>

long long int factorial (int n) {

    long long int resultado= 1;
    int num;

    for (num= 2; num<= n; num++) {
        resultado= resultado* num;
        printf ("Factorial de %d, resultado parcial %lld\n", n, resultado);
        sleep (random()\%3);
    }
    return resultado;
}

int main (int argc, char *argv[]) {

    if (argc== 2)
        printf ("El factorial de %s es %lld\n",
                argv[1], factorial (atoi (argv[1])));
    return 0;
}
```

- a) Escríbelo, compílalo y ejecútalo para comprobar su funcionamiento.
 - b) Escribe un nuevo programa que reciba dos números enteros como parámetros de entrada y cree dos procesos de manera que cada uno calcule el factorial de uno de los números, de forma concurrente, y utilizando el fichero ejecutable obtenido en el apartado anterior.
 - c) Haz que el proceso padre sea el último en terminar, es decir, que espere a la terminación de sus procesos hijos.
16. Generaliza la solución del problema 15 de manera que no esté limitado a 2 el número de factoriales a calcular. Procede de la siguiente manera:
- a) Crea un proceso por cada factorial a calcular, y que todos los procesos se ejecuten de forma concurrente.
 - b) El proceso padre deberá esperar a todos los procesos hijos y mostrar un mensaje a medida que vayan terminando indicando el PID del proceso finalizado.
 - c) Modifícalo para que no se imprima mensaje cuando el primer proceso hijo finalice, pero si para los demás.

```

bairon@bairon123:~$ ./punto16 5 4 6
Factorial de 5, resultado parcial 2
Factorial de 6, resultado parcial 2
Factorial de 4, resultado parcial 2
Factorial de 6, resultado parcial 6
Factorial de 5, resultado parcial 6
Factorial de 4, resultado parcial 6
Factorial de 5, resultado parcial 24
Factorial de 6, resultado parcial 24
Factorial de 6, resultado parcial 120
Factorial de 4, resultado parcial 24
El factorial de 4 es 24
Factorial de 5, resultado parcial 120
Factorial de 6, resultado parcial 720
El factorial de 5 es 120
Proceso hijo con PID 6304 finalizado
El factorial de 6 es 720
Proceso hijo con PID 6306 finalizado

```

a) Cada vez que el bucle **for** itera, se crea un nuevo proceso hijo mediante **fork()**. Este proceso hijo ejecuta inmediatamente el programa factorial a través de **execl** pasándole un número para calcular su factorial. Cada hijo se ejecuta de manera concurrente porque el proceso padre no espera a que uno termine antes de iniciar el siguiente; todos se lanzan en rápida sucesión.

b) Después de lanzar todos los procesos hijo, el proceso padre entra en un bucle que utiliza **wait()** para esperar a que cada proceso hijo termine. Cuando **wait()** retorna, devuelve el PID del proceso hijo que terminó, permitiendo al padre saber cuál de sus hijos ha finalizado. En el código se imprime un mensaje cada vez que un proceso hijo termina, indicando el PID del proceso finalizado. Ya con esto cumple con el requisito de que el padre debe informar sobre la finalización de los procesos hijos.

c)

- La variable **count** se inicializa en **0** antes de entrar en el bucle **while**.
- Dentro del bucle, cada vez que **wait()** retorna indicando que un proceso hijo ha terminado, incrementamos **count**.
- Solo imprimimos el mensaje si **count** es mayor que **1**, lo que significa que este no es el primer hijo que ha terminado. Esto cumple con el requisito de omitir un mensaje para el primer hijo finalizado.