

## PRACTICA 4

### Objetivo General

Poner en práctica lo aprendido con respecto a generación de árboles de procesos mediante la llamada al sistema `fork()`.

### Objetivos Particulares:

- Desarrollar un par de programas en C que permita desarrollar un árbol de procesos binario de cualquier nivel de profundidad, factor que será solicitado al usuario.
- Los dos programas que se solicitan en esta práctica corresponden a una implementación iterativa o una implementación recursiva.
- Esta práctica a primera vista pudiera parecer sencilla, pero se requiere que el alumno despierte técnicas de optimización de código, ya que se puede resolver con a lo más 20 líneas de código lógicas por cada parte que se solicita.

### Documentos para entregar

- La práctica debe realizarse de forma individual
- Video de ejecución de cada parte que se solicita en la práctica, en el cual se muestre la compilación usando el comando: `gcc -Werror -Wall nombre_fuente.c -o nombre_fuente`. Después de la compilación se tiene que ejecutar el código. Finalmente, se debe explicar la lógica de cada parte, OJO, la lógica no es solo leer el código, si no decir porque se realizó esta implementación, de esto depende su calificación principal (100 %).

### Plazo de entrega

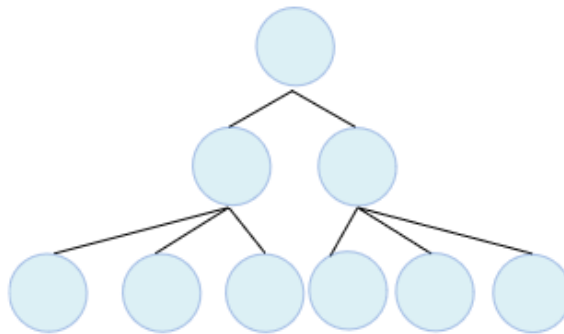
La hora y fecha límite para entregarla será el martes 18 de marzo del 2025 antes de la sesión del laboratorio, ya que la próxima semana se reanudan los exámenes en vivo durante esta sesión.

**Nota. No se recibirá ninguna práctica fuera de ese horario, sin ninguna excepción.**

**Especificaciones del programa para entregar:****Practica4\_A.c**

Realizar un árbol de procesos donde en cada nivel se incremente el total de hijos como se muestra en la Figura 1:

El aspecto visual de esta parte de la práctica se muestra en la Figura 1:



**Figura 1.** Árbol creciente.

En la Figura 2 podemos ver la ejecución con numero de niveles = 4, contando desde 0,1,2,3,4.



**Figura 2.** Salida deseada del primer problema

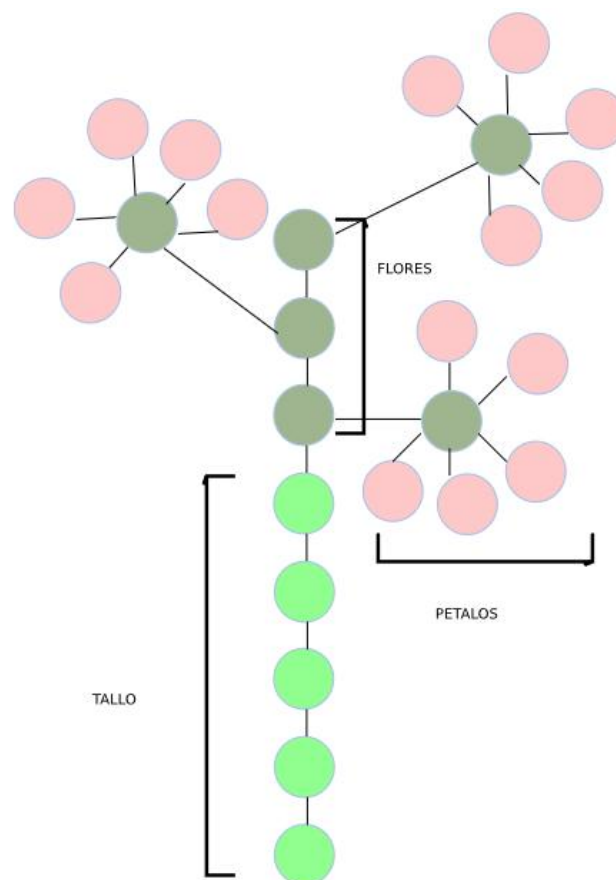
Donde el nivel 4 corresponde a los nodos padres de 5 nodos, este es el nivel máximo que puede contar correctamente el programa por una restricción que ustedes le pueden asignar, para ello deben investigar la instrucción `WEXITSTATUS(v)`; para saber cuándo los hijos de un padre van terminando, usando el siguiente fragmento de código:

```
int p = getpid();  
...  
int v_aux = 1;  
...  
if(getpid() == p){  
    printf("Proceso padre\n");  
}else{  
    exit(v_aux);}
```

**Código 1. Forma en como un proceso tiene que terminar**

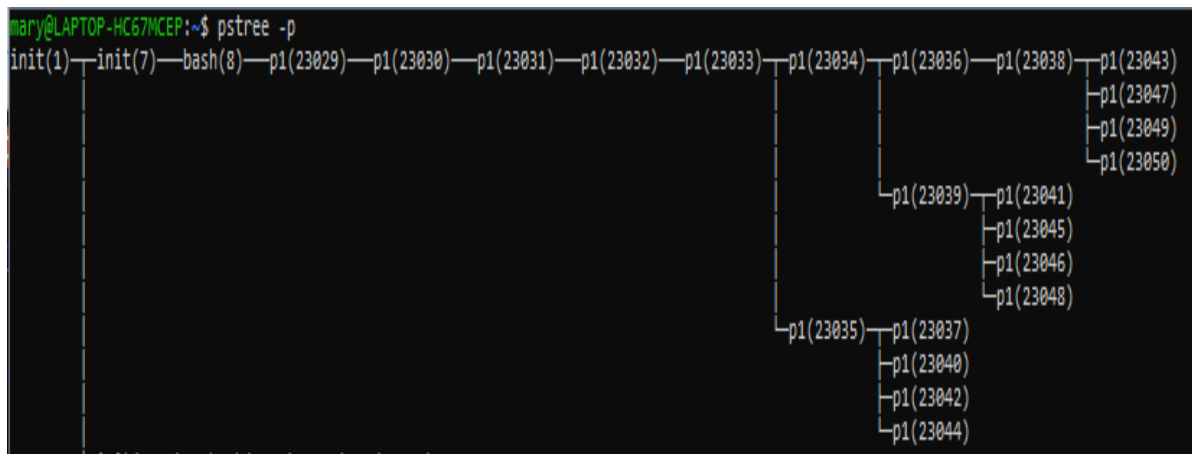
**Practica4\_B.c**

Realizar un árbol de procesos que represente una flor como se muestra en la Figura 3.



**Figura 3. Flor de Procesos**

En la Figura 4 podemos ver una flor constituida por 4 nodos para la raíz, 3 flores y 4 pétalos por flor.



**Figura 4.** Ejemplo de ejecución de la segunda parte de la práctica.