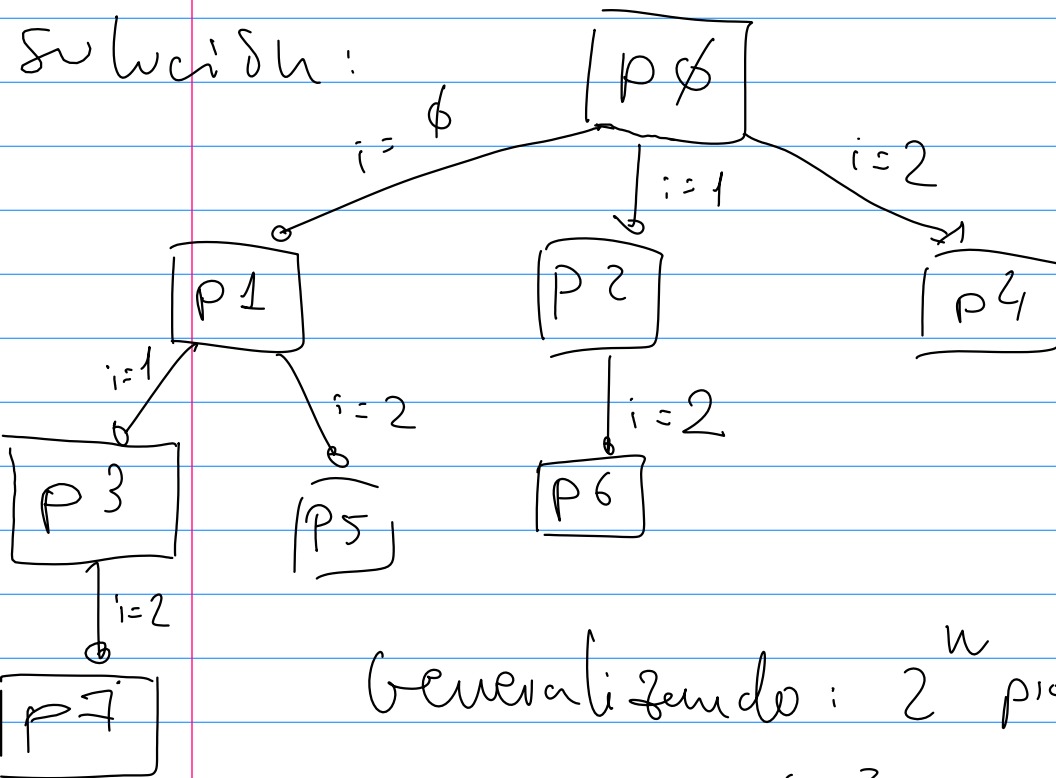


```

(1) void main(void) {
    int i;
    for(i=0; i<3; i++) fork();
}

```

Solution:



Generalizando: 2^n procesos
 $(2^3 = 8)$

②

→ ¿Qué se mostrará por pantalla al ejecutar el código?

Solución:

→ Soy el proceso original. Mi PID es 100;

→ Soy el proceso con PID 101. Mi padre es 100.
GLOBAL: 15 LOCAL: 2 ← Mijo

→ Soy el proceso con PID 100. Mi padre es 1.
GLOBAL: 20 LOCAL: 13 ← PADRE

- porque es otra ^b región de memoria independiente
- No hay forma sencilla con un fork de compartir regiones de memoria !!

→ ¿Es posible que los valores de las variables varíen según el planificador?

→ No, es tan solo aseverando con el main y con regiones de memoria independientes

→ ¿Podrían salir los mensajes en otro orden?

No! por el wait, nos aseguramos el orden

③ Si hay un hijo, se reemplaza todo su código por otro, en este caso, "comunto".

sólo habrá 4 hijos entonces

Como mínimo coexistirán 2 procesos

Sólo el padre llega a "imprimir", en este caso a vale 7 y $b - 2 \rightarrow$

(el $b++$ de $p = 5$ no interfiere por que es independiente).