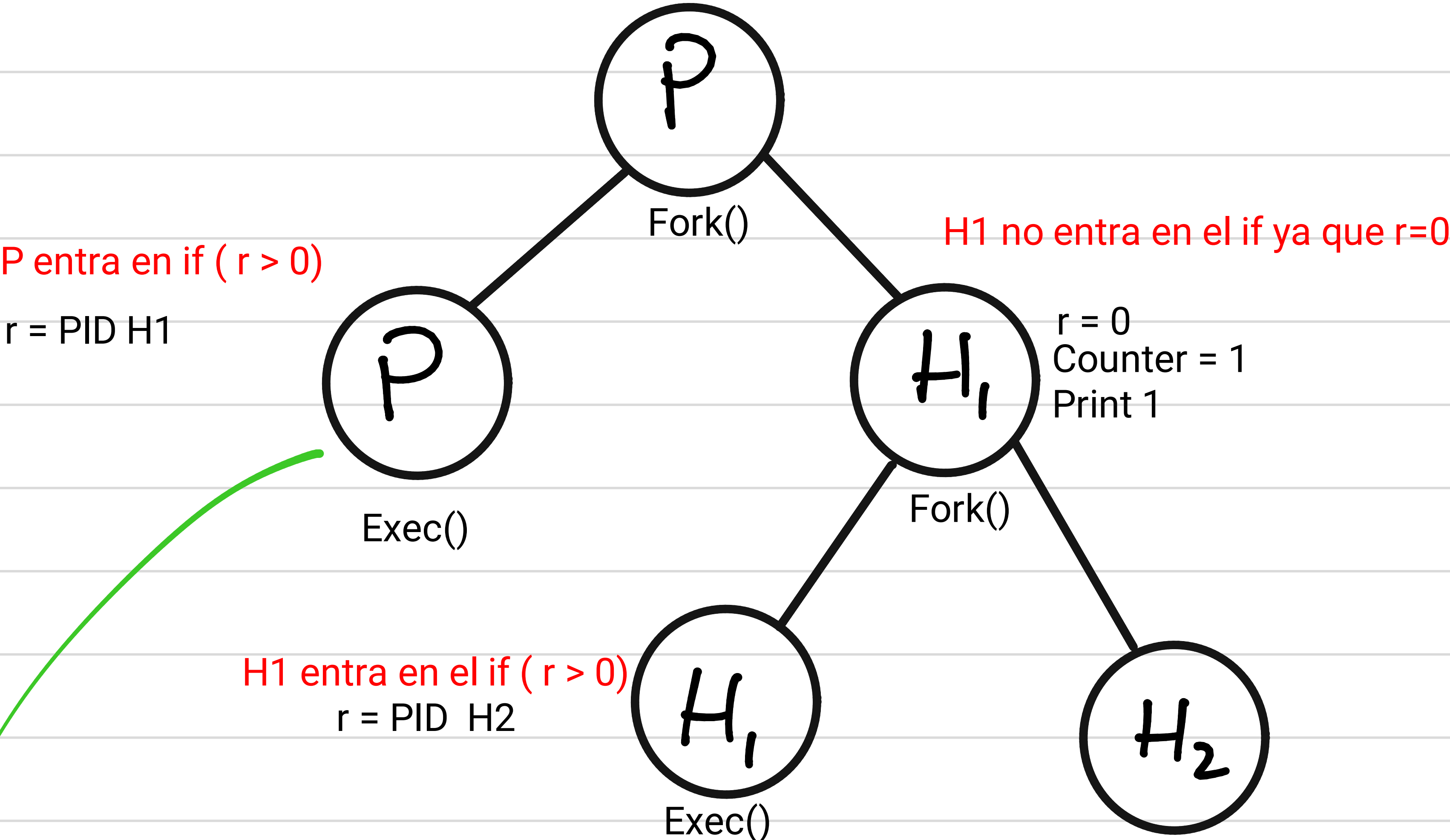
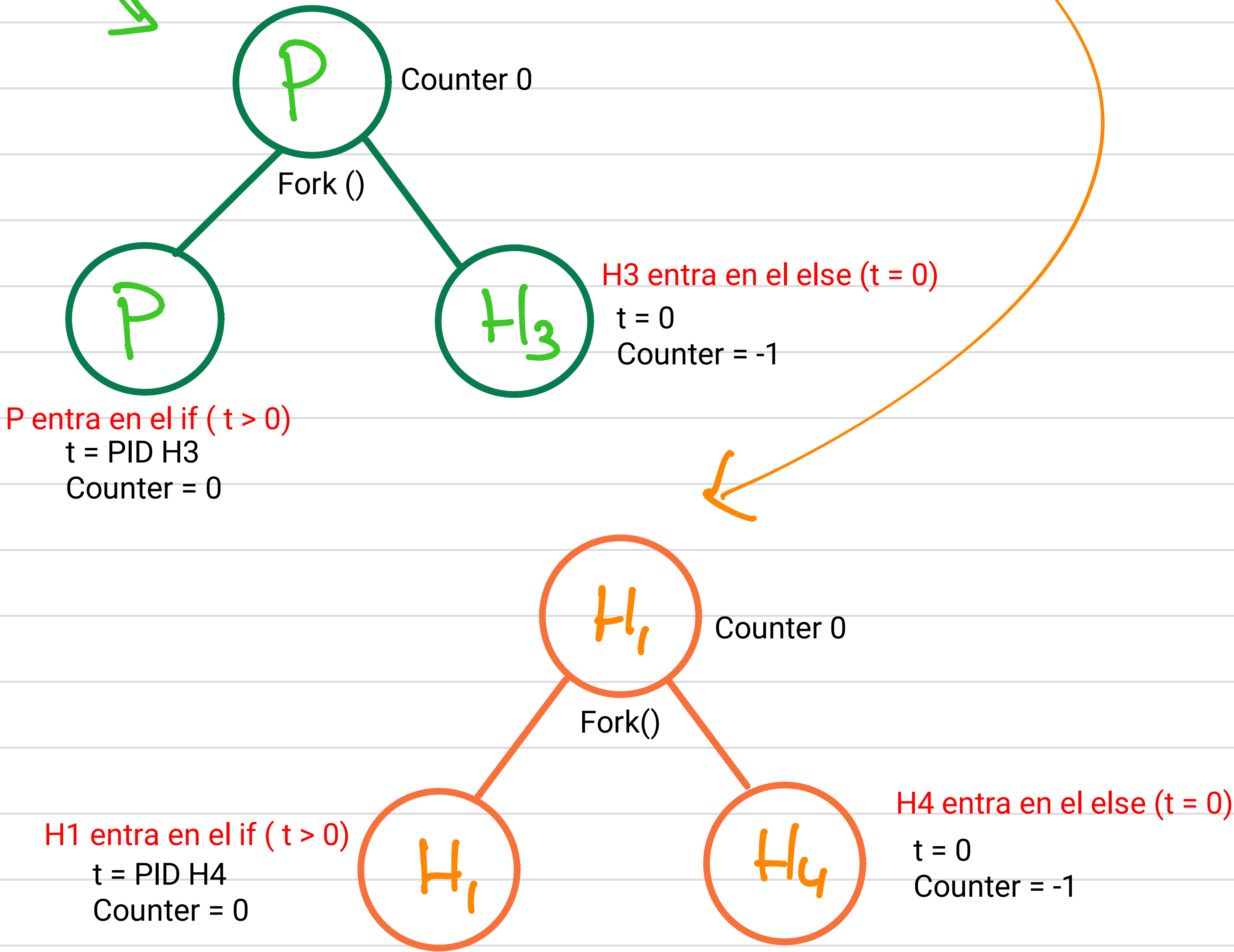


1



Al hacer exec el pid de P es el mismo siempre, ya que no cambian al ejecutar un exec, pero todo el código anterior a este se olvida.

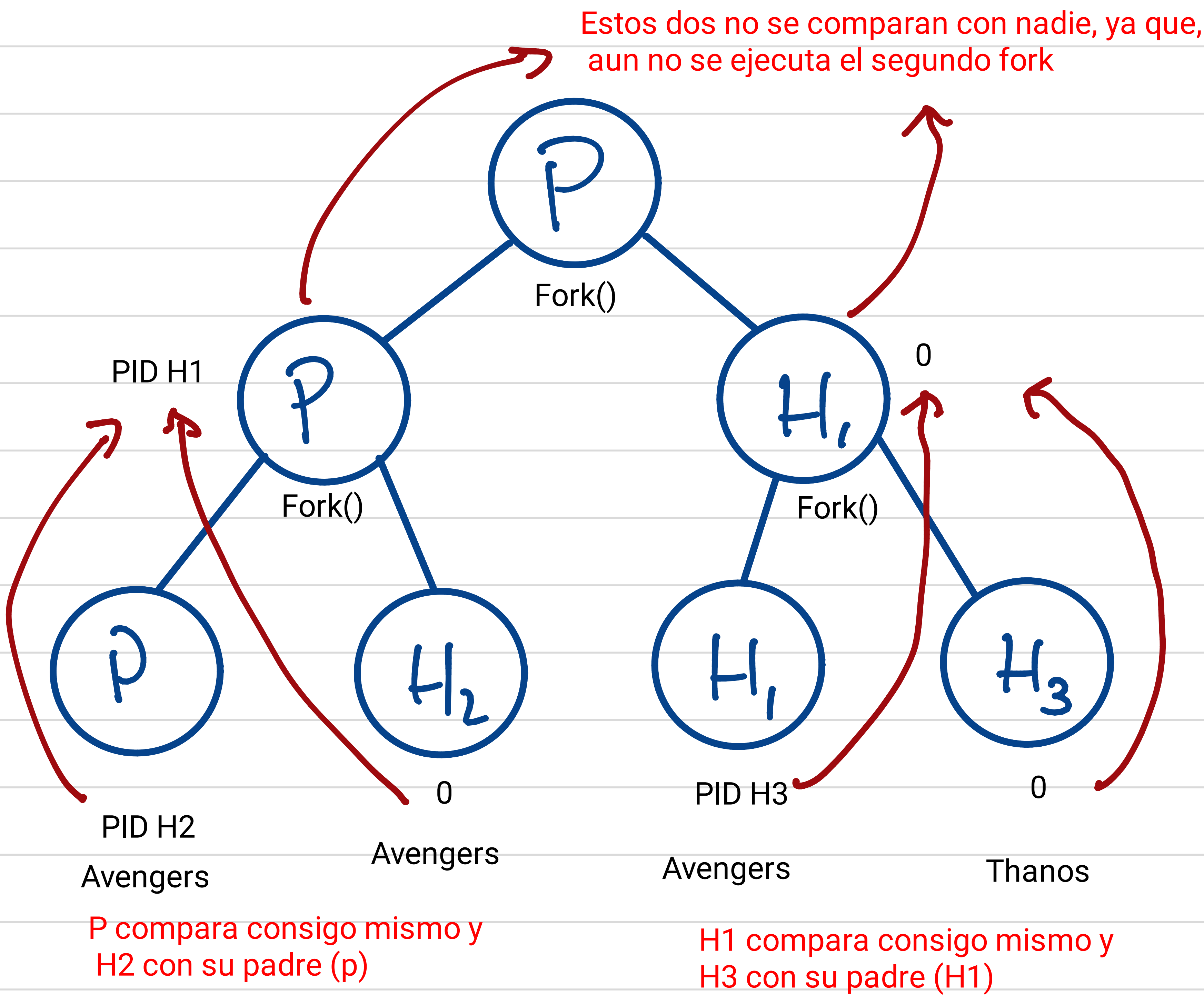


A) ¿Cuántos procesos se crean?
Se crean 5 procesos, donde se crean 3 procesos en el código 1, juntos a 2 procesos creados en el código 2 al momento de hacer exec.

B) Explique la razón detrás de la existencia de múltiples salidas a partir de la ejecución.
ejemplifique

Una posible salida es 2,1,0,-1,0,-1, pero existen múltiples salidas debido a la falta de sincronización entre los procesos, eso nos indica que puede variar la salida por cada ejecución del código.

2



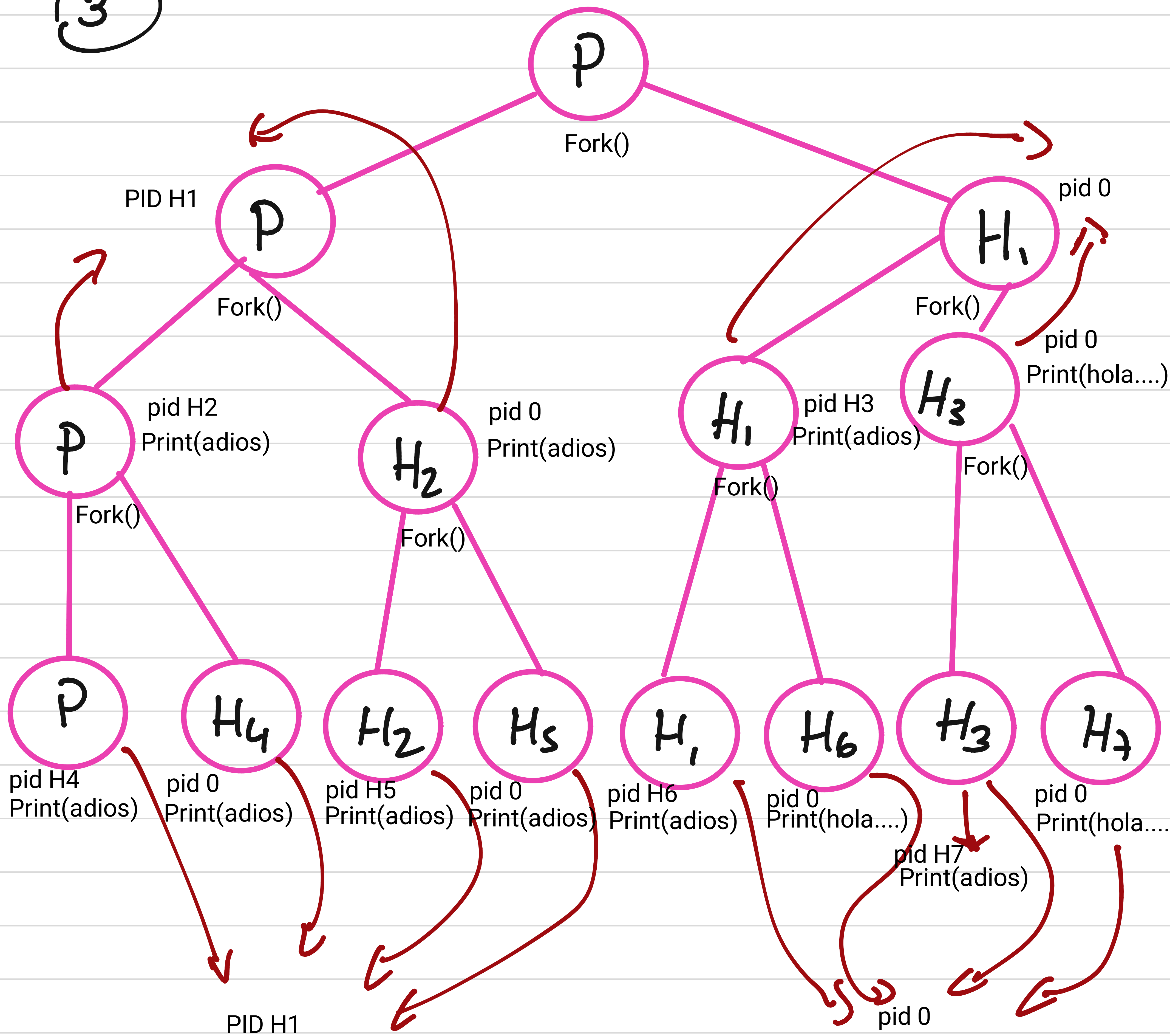
A) ¿Cuántos procesos se crean?

Se crean 4 procesos en el código.

B)¿Cual puede ser una salida del código?

Una salida puede ser Avengers, Avengers,Thanos, Avengers, pero ta como se explico anteriormente al no tener sincronizacion puede haber muchos resultados.

3

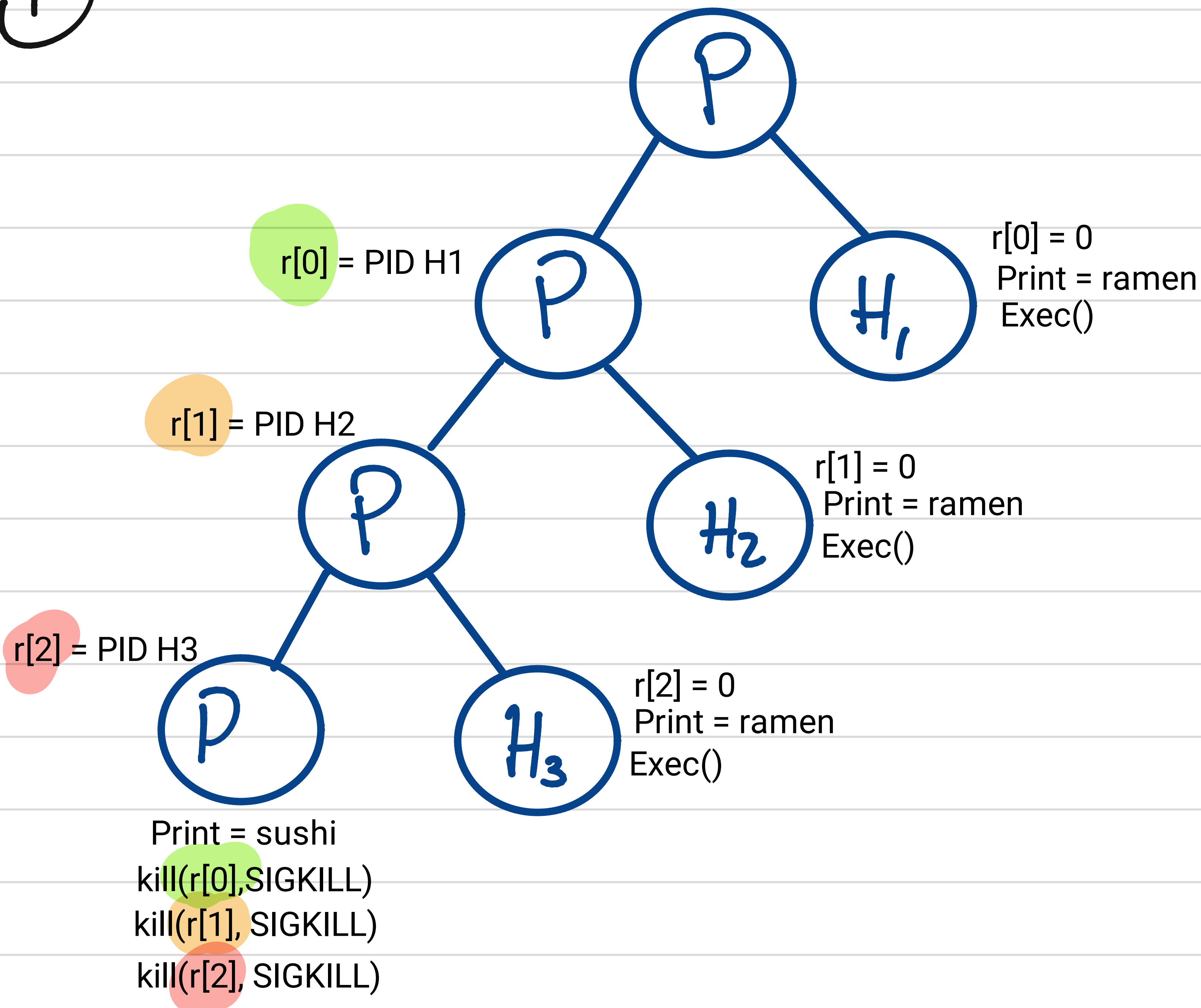


Tal como el ejemplo anterior por el lado izquierdo todos se compara con el PID de H1, al ninguno tener su PID todos muestran Adios, en el caso de la derecha todos los pid que llevan 0 muestran Hola....., y el resto que no entra en el if muestra adios

A) se crean 8 procesos

B) una salida puede ser Adios,Adios,Adios,Hola...,Adios,Adios,Hola...,Adios,Adios,Hola...,Adios, Adios por las razones ya explicadas.

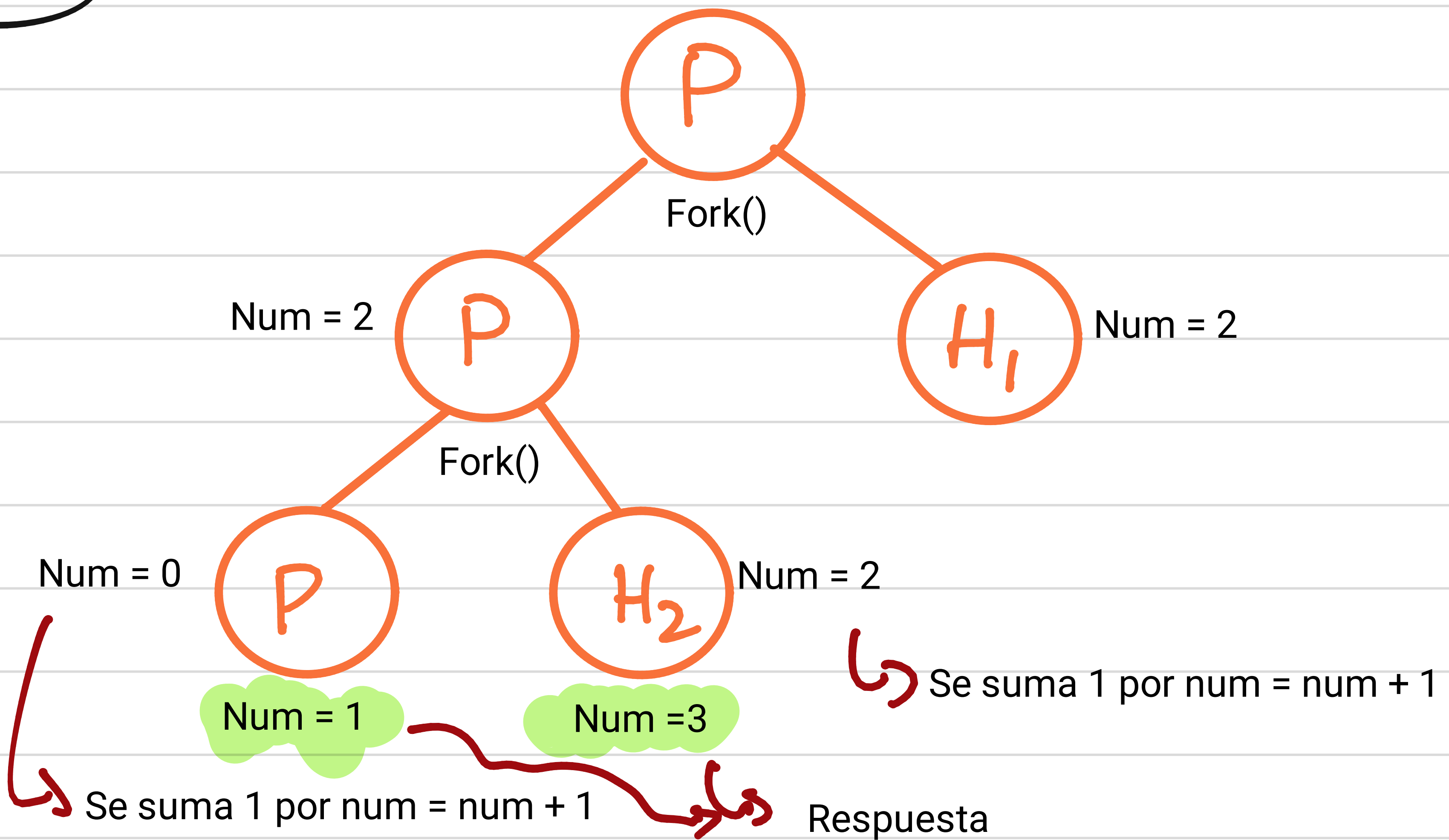
4



Sigkill: fuerza la eliminacion de un proceso

A) se crean 4 procesos, ya que al momento de hacer exec hace un sleep() de 666 segundos y no alcanza a hacer otro fork porque el proceso padre solo tiene que esperar 10 segundos y entra a un for donde elimina todos los procesos creados ya que previamente guardo sus PID, lo que hace que pueda eliminarlos facilmente una posible salida es ramen,ramen,ramen,shushi

S



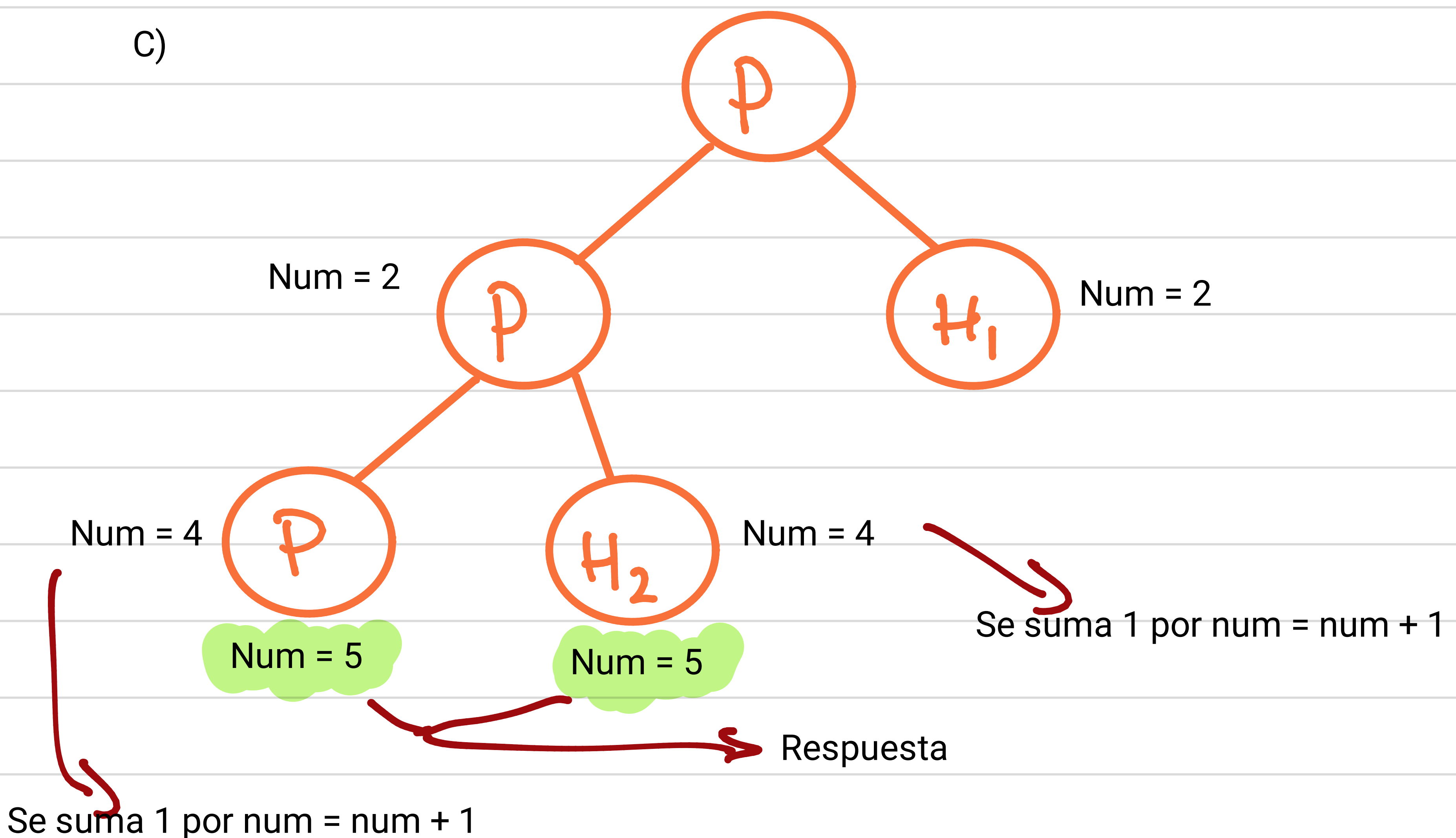
A) ¿Cuantoas procesos se crean en total?

Se crean 3 procesos

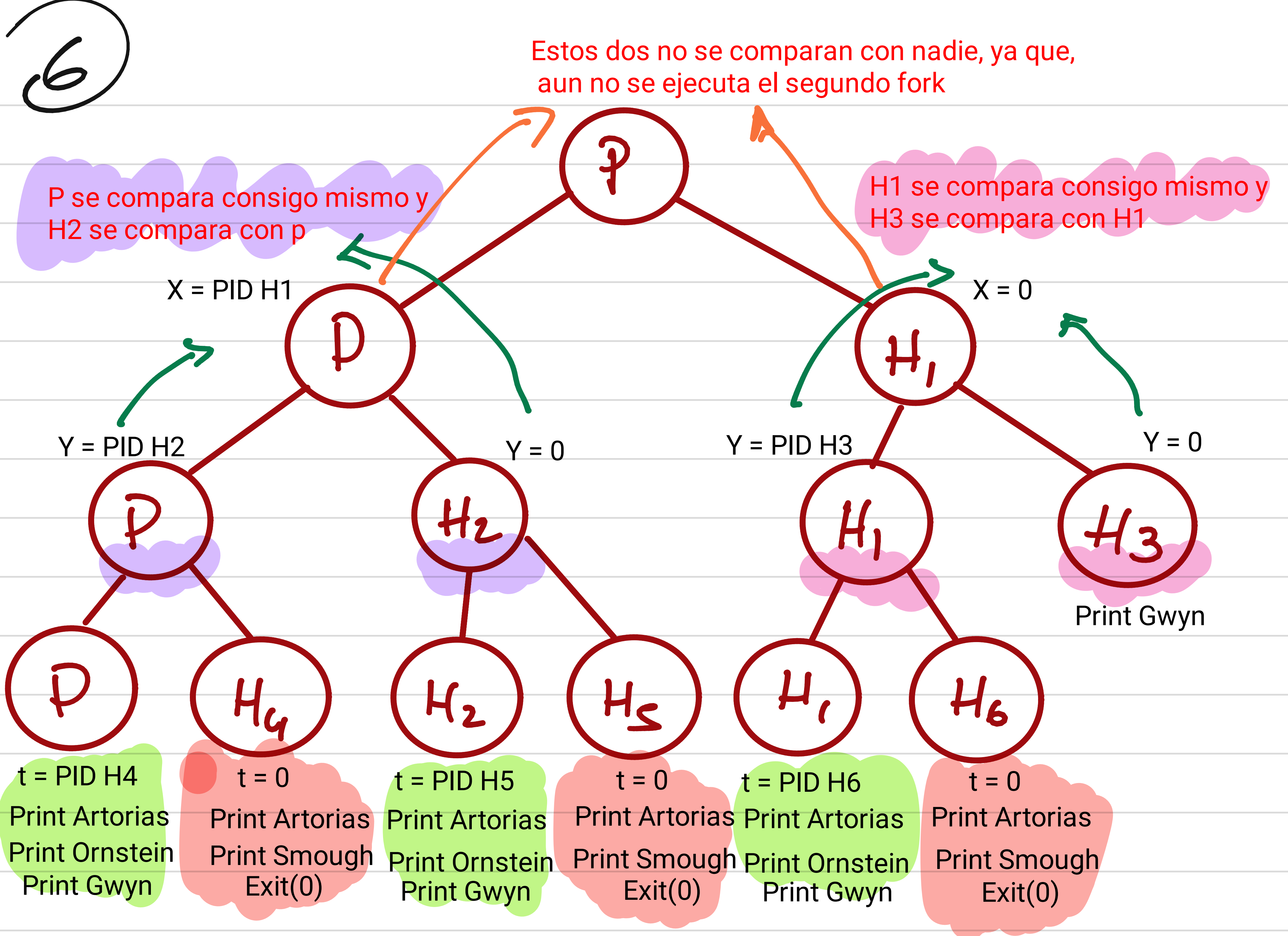
B) Indique dos posibles salidas

1,2,3
2,1,3
3,2,1.... etc

C)



una posible salida es 5,5,2 ya que al fallar el fork() devuelve -1, por ende entra en el else if (tt<0).



→ Compilacion donde $t = 0$

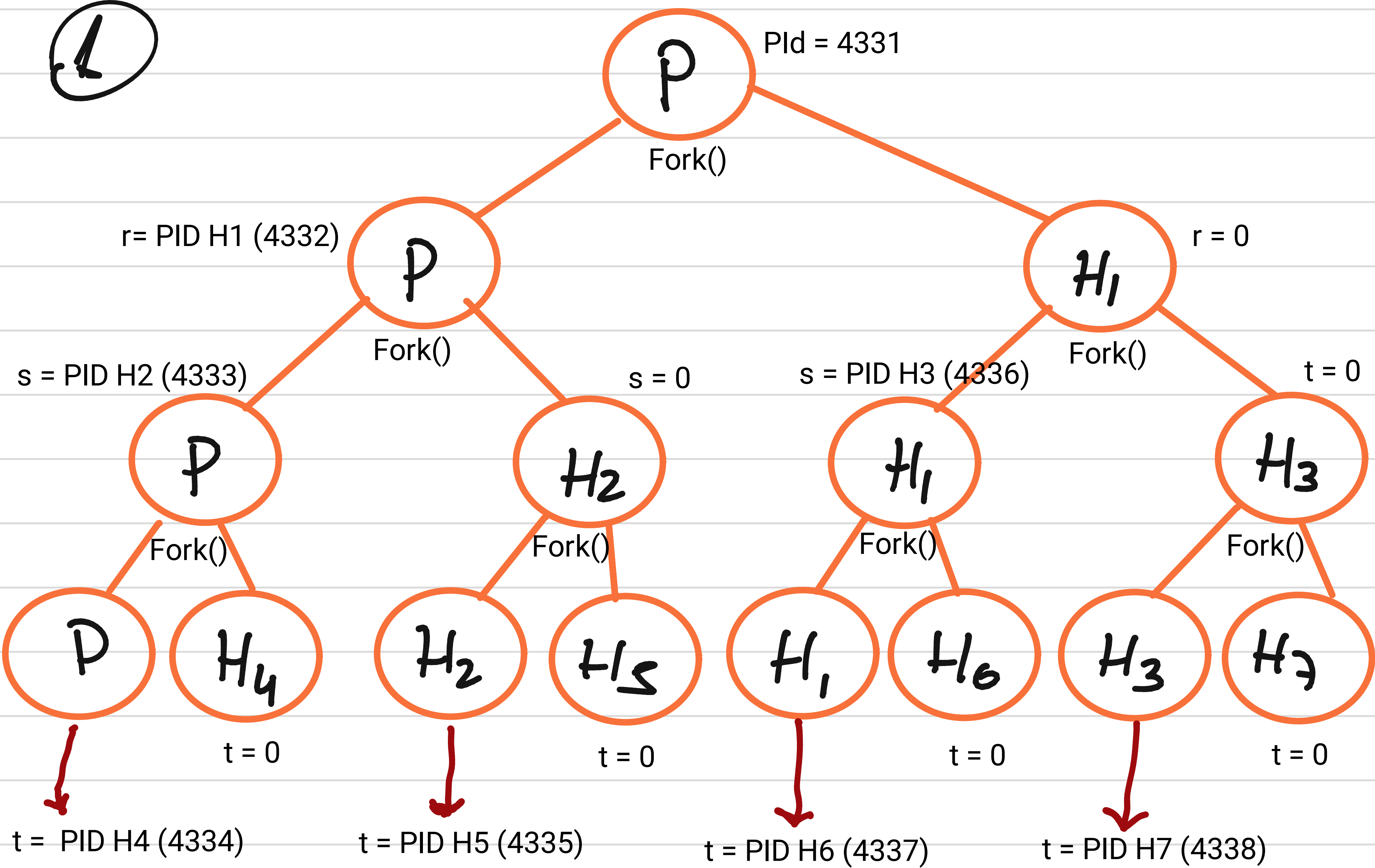
→ Compilacion donde $t > 0$

A) Se crean en total 7 procesos, como muestra el arbol generado

B) Se imprime 6 Artorias, 3 Ornstein, 3 Smough y 4 Gwyn

Control

1



A)

	r	s	t
r = PID H1 (4332)	s = PID H2 (4333)	t = PID H4 (4334)	
		t = 0	
		t = PID H5 (4335)	
r = 0	s = 0	t = 0	
		t = PID H6 (4337)	
	s = PID H3 (4336)	t = 0	
		t = PID H7 (4338)	
	s = 0	t = 0	

B) 4 Procesos pasan al codigo Y gracias a ser impares, pero al pasar al codigo Y siempre caeran en $m \% 2 == 1$, esto hace que compilen `Exit(0)` y terminen los procesos, los procesos que terminan son P, H2, H5, H6.

C) al hacer el cambio de $m\%2 == 1$ a $m\%2 == 0$, los procesos que ingresen al if se volveran infinitos, ya que, el codigo X siempre creara un proceso impar , esto hace que pose al codigo Y, y asi creandose un codigo infinito

2

Codigo Padre

```
int main(){
pid_t padre;
padre = fork();
if(padre > 0){
fork();
}
if(padre > 0){
tomar_desayuno();
ducharse();
vestirse();
preparar_auto();
wait(&status);
}else{
exec("./Gemelos","",NULL);
}
return 0;
}
```

Codigo Gemelos

```
int main(){
tomar_biberon();
bañarse();
vestirse();
return 0;
}
```