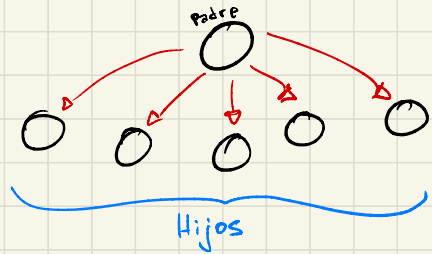


Pauta Ayudantía 1

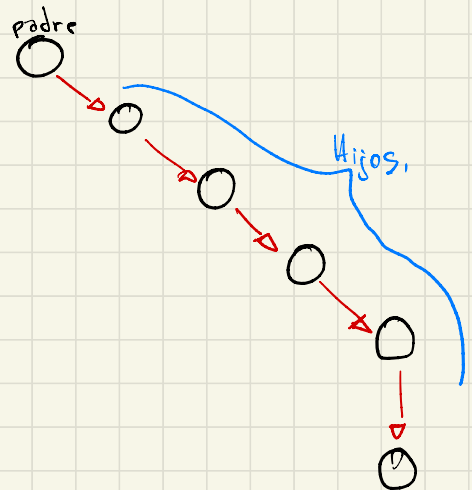
1)

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <unistd.h>
4
5  int main(){
6      pid_t pid;
7
8      for(int i = 0 ; i < 5 ; i++){
9          pid = fork();
10         if(pid == 0){
11             break;
12         }else if(pid < 0){
13             break;
14         }
15     }
16 }
```



2)

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <unistd.h>
4
5  int main(){
6      pid_t pid;
7
8      for(int i = 0 ; i < 5 ; i++){
9          pid = fork();
10         if(pid > 0){
11             break;
12         }else if(pid < 0){
13             break;
14         }
15     }
16 }
```



3)

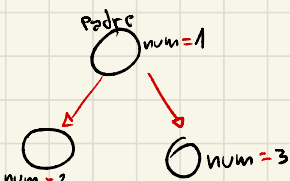
```

1 int num = 1; //variable global
2 int main(){
3     pid_t t = fork();
4     if (t!=0){ //Solo padre
5         num = num+2;
6         pid_t tt = fork();
7         if (tt>0){ //Padre
8             num = num-2;
9         }
10        else if (tt<0){
11            fork(); //error
12            num = num+2;
13        }
14    }
15    num = num+1;
16    sleep(1);
17    printf("%d\n", num);
18 }

```

siempre suponer
que los fork(s)
funcionan a menos
que se diga lo
contrario

- ¿Cuántos procesos se crean en total? Explique.
- Indique 2 posibles salidas. Explique.
- Suponga ahora que por una razón misteriosa la primitiva fork() de la línea 6 falla para todos los posibles procesos que ejecutan dicha instrucción. ¿Cuál(es) sería(n) la(s) posible(s) salida(s)? Explique.

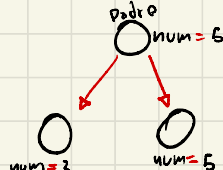


• 3 procesos

• 1 2 3
2 1 2
3 3 1

posibles salidas
no se sabe qué proceso
terminará primero.

• En caso que falle fork()
en línea 6.



• Posibles salidas:

5 5 2
5 2 5
2 5 5

4)

Dado el siguiente código en C:

```

1 int main() {
2     pid_t t = fork();
3     int i = 2;
4     int status = -1;
5     if (t > 0){
6         status = 2;
7         i = 6;
8         wait(&status);
9     }
10    else{
11        if (i > 5){
12            exit(2);
13        }
14        else{
15            exit(3);
16        }
17    } //continua al lado

```

Padre

Hijo/Error

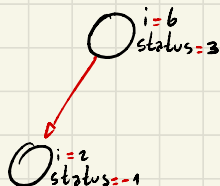
```

1 if (WEXITSTATUS(status) == -1){
2     printf("Los\n");
3 }
4 if (WEXITSTATUS(status) == 2){
5     printf("Pollos\n");
6 }
7 if (WEXITSTATUS(status) == 3){
8     printf("Hermanos\n");
9 }
10 return 0;
11 }

```

Solo compara
status

Determine la(s) posible(s) salida(s). Explique.



Posible salida: Hermanos