

PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ
FACULTAD DE CIENCIAS E INGENIERÍA

INF239 SISTEMAS OPERATIVOS

Semestre 2023-1

Laboratorio 2

1) (8 puntos – nombre del programa: *chainOne.c*) Se le solicita escribir un programa en C, donde el proceso principal cree una cadena de n procesos. El número n debe ser ingresado como argumento por la línea de ordenes. Después que se ha creado los n procesos, el proceso principal deberá ejecutar la orden `pstree` para mostrar la cadena de procesos que se ha generado. El proceso principal debe ir eliminando cada proceso creado, desde el último hasta el primero. Cada vez que elimina un proceso, deberá ejecutar `pstree` para mostrar que el proceso correspondiente ha sido eliminado.

No puede hacer uso de la función de librería *sleep*.

La salida deberá ser análoga a la siguiente.

```
alejandro@abdebian:2023-1$ ./chainOneSol 5
chainOneSol(794465)──chainOneSol(794466)──chainOneSol(794467)──chainOneSol(794468)──chainOneSol(794469)──chainOneSol(794470)
chainOneSol(794465)──sh(794471)──pstree(794472)
chainOneSol(794465)──chainOneSol(794466)──chainOneSol(794467)──chainOneSol(794468)──chainOneSol(794469)
chainOneSol(794465)──sh(794473)──pstree(794474)
chainOneSol(794465)──chainOneSol(794466)──chainOneSol(794467)──chainOneSol(794468)
chainOneSol(794465)──sh(794475)──pstree(794476)
chainOneSol(794465)──chainOneSol(794466)──chainOneSol(794467)
chainOneSol(794465)──sh(794477)──pstree(794478)
chainOneSol(794465)──chainOneSol(794466)
chainOneSol(794465)──sh(794479)──pstree(794480)
chainOneSol(794465)──sh(794481)──pstree(794482)
```

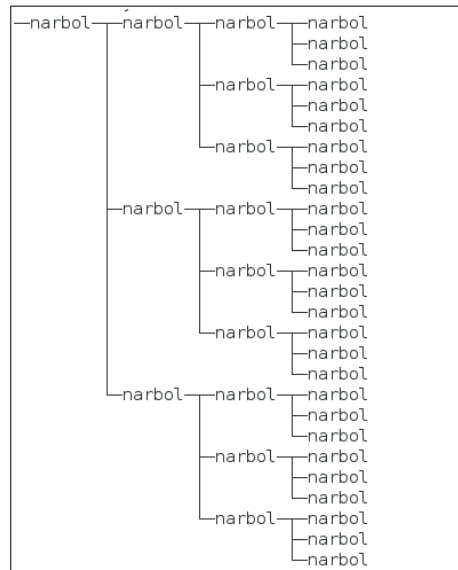
2) (8 puntos – nombre del programa: *swapTwo.c*) Se le solicita escribir un programa en C, donde el proceso principal cree dos procesos hijos. Cada proceso hijo genera n números aleatorios, donde n es ingresado como argumento del programa. Cada proceso hijo debe analizar su *pid*, si su *pid* es par deberá imprimir los números pares tanto los suyos como los de su hermano. De formas análoga para el proceso cuyo *pid* es impar, deberá imprimir los números impares tanto los suyos como los de su hermano. Incluya mensajes apropiados para identificar los procesos y los momentos en que se imprimieron.

Por ejemplo, una salida sería:

```
alejandro@abdebian:2023-1$ ./swapTwoSol 5
Antes: My pid=795253 my number=28
Antes: My pid=795253 my number=40
Antes: My pid=795253 my number=16
Antes: My pid=795253 my number=11
Antes: My pid=795253 my number=27
Después: My pid=795253 my number=11
Después: My pid=795253 my number=27
Antes: My pid=795254 my number=28
Antes: My pid=795254 my number=31
Antes: My pid=795254 my number=3
Antes: My pid=795254 my number=13
Antes: My pid=795254 my number=24
Después: My pid=795254 my number=28
Después: My pid=795254 my number=40
Después: My pid=795253 my number=31
Después: My pid=795254 my number=16
Después: My pid=795253 my number=3
Después: My pid=795254 my number=28
Después: My pid=795253 my number=13
Después: My pid=795254 my number=24
```

3) (4 puntos – nombre del programa: *narbol.c*) Se le solicita escribir un programa en C de forma que lea de la línea de comandos un número entero positivo mayor que 1. El proceso principal deberá crear n hijos, y cada uno de sus hijos hará a su vez lo mismo. Esto se debe repetir n niveles. El padre se encuentra en el nivel 0. El programa mismo deberá mostrar la forma del árbol formado, usando la función `pstree`.

Por ejemplo, para $n = 3$



Porf. Alejandro T. Bello Ruiz