



INSTITUTO POLITÉCNICO NACIONAL



Escuela Superior de Cómputo

Unidad de Aprendizaje: Sistemas operativos

Práctica 1

Profesor: Josué Rangel Gonzalez

Integrantes del equipo:

Martín Eduardo Barriga Vargas

José Manuel Ramirez Vives

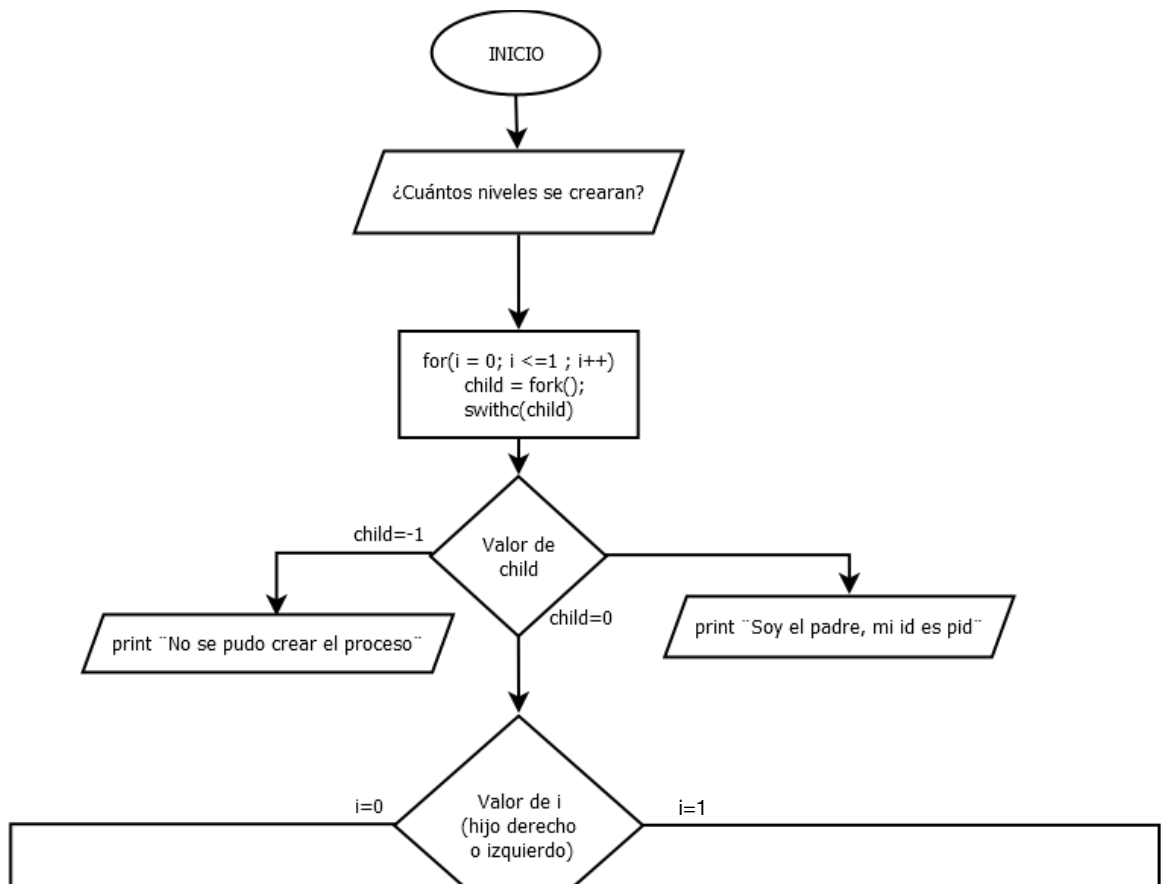
Fecha de entrega: Viernes 23 de marzo del 2018

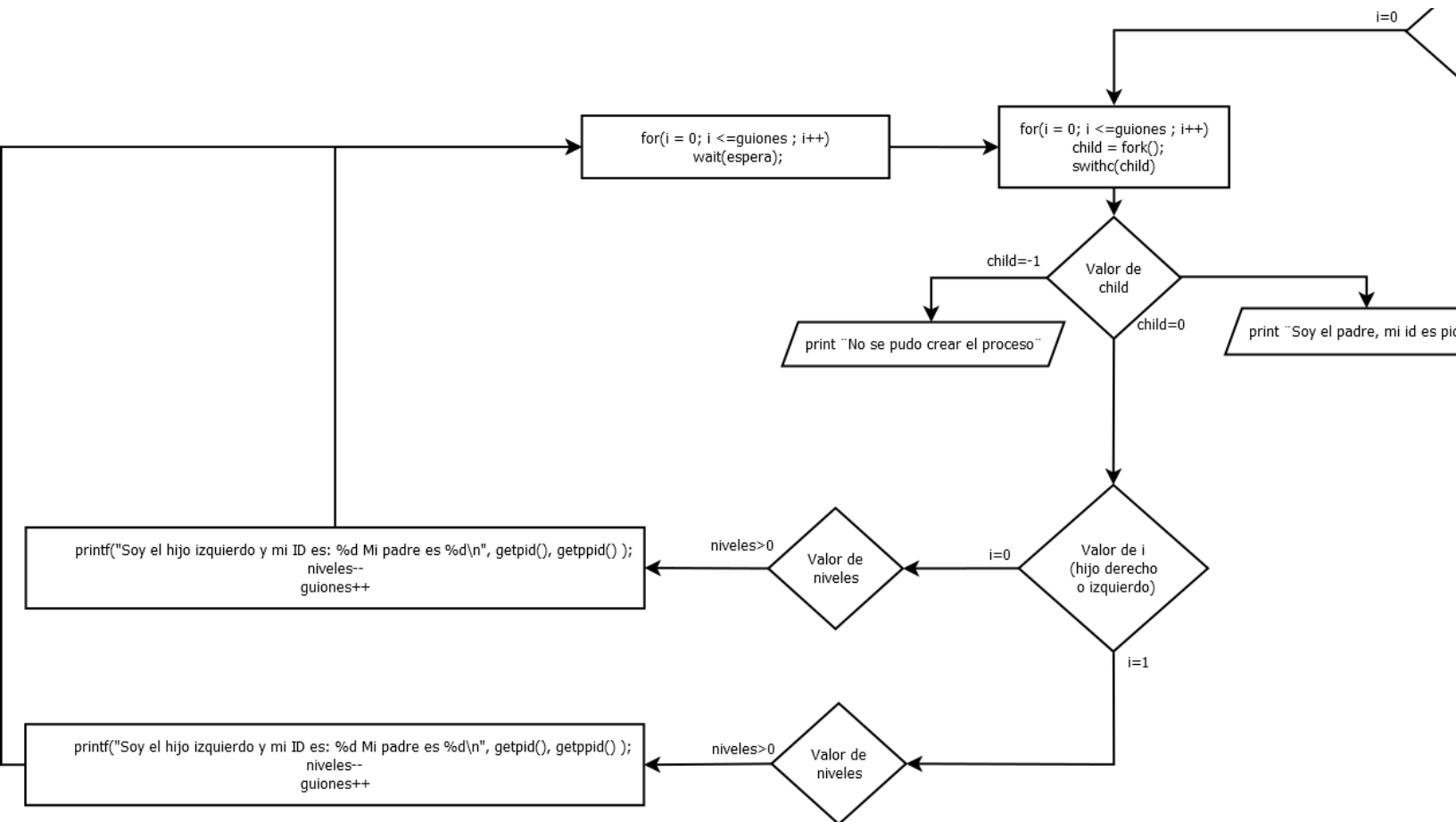
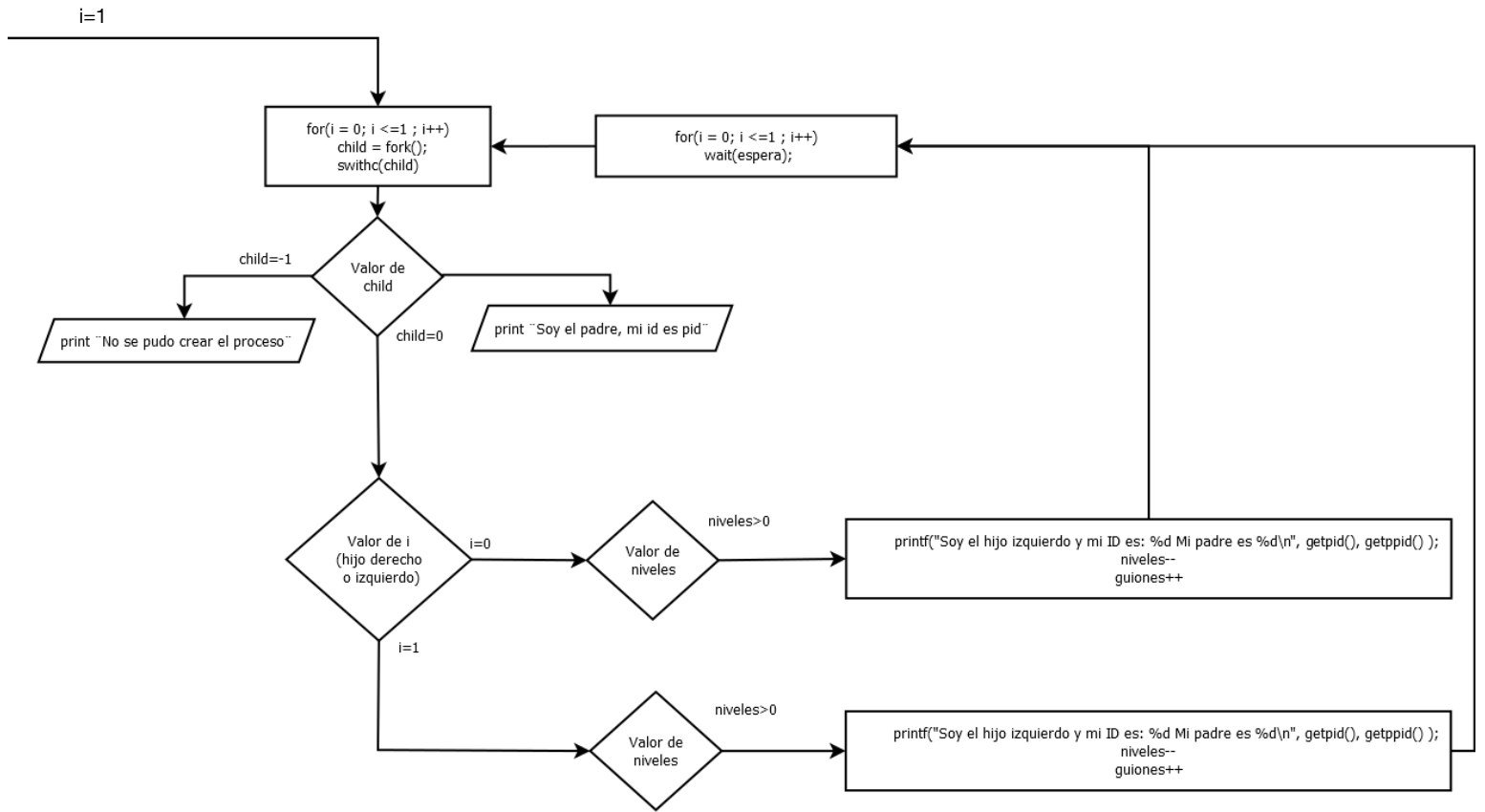
Introducción:

A diario utilizamos nuestros ordenadores y estos ejecutan montones de procesos para operar de la forma en que lo requiera el usuario. Normalmente se cree que todos los procesos involucrados se están ejecutando a la vez, es decir, todos al mismo tiempo, sin embargo sólo se ejecuta uno a la vez y esta es por esto que el trabajar con procesos puede llegar a parecernos ligeramente complicado. Para agilizar nuestra habilidad de usar procesos y adentrarnos más al tema, deberemos de crear un árbol en el cual todos los nodos(procesos) del lado derecho generen exactamente dos hijos, y que la parte de la izquierda genere la cantidad del nivel en el que se encuentra más un hijo. Además deberemos de mostrar la información de los nodos (procesos) con su id y que mencione si es un padre o un hijo, en caso de que sea un hijo también se deberá mostrar el id del padre.

Descripción de la solución:

Diagrama de flujo





Descripción

Una vez iniciado el programa se le pide al usuario ingresar la cantidad de niveles que tendrá el árbol, es importante recordar que los niveles comenzarán desde el nivel cero con el proceso padre(primer nodo).

Luego de que se haya ingresado el número de niveles se creará el primer hijo(de dos en total), este será el izquierdo debido a que ya sabemos que es la primera vez que creamos un hijo y mandaremos llamar una función que reciba dos parámetros: uno para la cantidad de hijos que debe crearse para ese nodo, el cual coincide con el nivel en el que se encuentra más uno, y otro que nos indicará la cantidad de niveles que faltan por crearse.

Dentro de esa función se crearán tantos hijos como lo haya indicado el parámetro y por cada hijo indicaremos el nivel en el que se encuentra, así como su pid y el de su padre. Además si aún no llegamos a la cantidad de niveles faltantes volveremos a llamar a esa misma función con el parámetro de niveles menos uno, y con la cantidad de hijos por crear incrementado en uno. Es importante indicar que se deberá de esperar a los hijos creados en cada una de las funciones.

A la vez, se creará al hijo derecho y se llamará a la función que creará sólo dos hijos por cada nodo, el saber que es hora de mandar llamar a esta función así como de indicar que se trata del hijo derecho se debe a que es de nuestro conocimiento que es la segunda vez que creamos un proceso hijo. A nuestra función también le mandaremos el parámetro de la cantidad de niveles que faltan por crearse y el nivel en el que se encuentra, sin embargo esta función sólo se encargará de crear siempre un hijo izquierdo y luego un hijo derecho. Cada vez que creamos uno de los hijos no sólo deberemos preguntar si se trata de un derecho un izquierdo, sino que también debemos de preguntar si ya hemos llegado a la cantidad de niveles necesarios, en caso de que no entonces volveremos de mandar llamar a esta misma función para crear a los hijos de dicho nodo.

Conclusiones:

Martín Eduardo Barriga Vargas: Para poder completar la práctica se tuvo que hacer uso de la instrucción fork, pues ésta nos ayudará a crear a los hijos. Sin embargo, mientras más se avanzaba con la práctica, más se complicaba. Esto debido a que se copiaba el código (proceso) y teníamos que cuidar que este nuevo proceso generado no se metiera en secciones de nuestro código que no le correspondían. Para esto se ingeniaron unos ifs que constantemente cuestionaban si el proceso generado era un padre, un hijo izquierdo, derecho, etcétera, para que nos imprimiera los pid's correspondientes y que en caso de ser hijo y que aún no se hayan completado los niveles del árbol pedidos por el usuario, entonces mande llamar a una función para generar la sección del árbol donde cada proceso sólo genera dos hijos (en caso de que haya sido el hijo derecho de la función main), en caso de que haya sido el hijo izquierdo de la función main entonces mandamos llamar a nuestra función que se encargará de generar la cantidad de nodos hijos correspondientes a su nivel.

Jose Manuel Ramirez Vives: La práctica nos ayudó bastante a comprender el funcionamiento de los procesos en la máquina y cómo se generan los procesos hijos mediante la función fork. Además, se puede observar de cierta manera la ejecución de cada proceso en el procesador, pues nos daos cuenta que no se ejecuta todo a la vez como es la idea que generalmente se tiene. Considero que lo que más se nos complicó de la práctica fue precisamente el entender la lógica de la función fork, el entender que se ejecutarían las instrucciones siguientes al llamado y poner las condiciones distintas para definir en qué parte del proceso se encontraba y, entonces, qué acciones debía ejecutar, también haciendo uso de la recursión y entendiendo dónde era el punto en que se debía hacer esperar a los procesos hijos para cada padre, evitando así que muriesen antes que los hijos.