SISTEMAS OPERATIVOS MEMORIA PRÁCTICA 1

Ángel Manuel Martin Canto

Darío Adrián Hernández Barroso

Grupo 2273

Introducción

APRENDIZAJE

1,2,3

CODIGO

Ejercicio 4:

En este ejercicio se pedía estudiar el árbol de procesos de un código dado. Dicho código generaba procesos hijo en un bucle, pero al carecer de una función wait() los procesos hijo se quedaban huérfanos.

Una vez modificado el código para no dejar huérfanos, usamos las funciones getpid() y getppid() para saber los pid de los procesos generados en el bucle.

Ejercicio 5:

Para controlar la creación de procesos de forma secuencial en este ejercicio, es importante tener en cuenta el pid del proceso, dado que si es el del padre no queremos generar un proceso. Con una simple condición como ésta, es fácil generar x procesos de forma secuencial.

En el segundo caso, sólo queremos que el primer padre genere procesos hijo, por lo que ésta vez la condición antes de la llamada fork() será comprobar el pid del primer proceso. Lo importante a tener en cuenta es que al final del código, no puede haber sólo una llamada a wait() Dado que eso generaría procesos huérfanos. En su lugar tiene que haber un bucle for con tantas vueltas como forks se han hecho antes, y en su interior la llamada a la función wait().

Ejercicio 6:

Cuando un proceso reserva memoria y después llama a la función fork(), la fotocopia creada para el proceso hijo incluye la reserva de memoria, por lo que los 2 tienen que liberar memoria al final del programa o habrá fallos de memoria.