INSTITUTO TECNOLÓGICO Y DE ESTUDIOS SUPERIORES DE OCCIDENTE

FUNDAMENTOS DE SISTEMAS OPERATIVOS

Actividad 03 – Ejemplos de Programas

Presentan

Andrea Lizeth Arevalo Solis, 752109

Hannah Chenoa Puente Rosales, 706413

Profesora: José Luis Elvira Valenzuela

1 de junio del 2025

Podemos decir que con esta actividad algunos ejercicios se nos hicieron un poco complicados por la lógica a la que queríamos llegar y también para refrescar nuestros conocimientos de la clase y la programación. Además, fue de ayuda preguntarnos a nosotras mismas y consultar a compañeros sobre cómo se podría llegar a la solución, para así entenderlo más profundamente y lograrlo. También se reforzó el aprendizaje sobre los procesos hijos mediante la llamada a fork(), entendiendo la diferencia de ejecuciones del padre y el PID. Logramos implementar pausas con sleep() y esperas del programa con wait(), esto para que el proceso hijo pudiera finalizar antes de que el padre continuara.

Otra parte complicada fue entender los niveles en un árbol, pues en cada iteración se duplicaba la cantidad de procesos, claro siempre empezando en el nivel 0. Cada uno de ellos imprimía el número de nivel por el que pasaba, lo que ayudó a visualizar cómo se formaba el árbol y a calcular los procesos que se debían generar para n niveles, donde existían 2^n procesos que están activos. También nos basamos en ejercicios que vimos en clase o en videos similares a lo que se nos pedía, como el ejercicio de generar un conjunto de hijos en un único nivel, donde se imprime cada hijo con su propio identificador mientras el padre espera en wait() hasta que todos los hijos hayan terminado.

También el programa creamos un intérprete de comandos nos retó pues tuvimos que investigar cómo hacerlo. Lo logramos con un Shell muy básico: empleamos fgets() para leer la línea de entrada del usuario y usamos la llamada system() para ejecutar el programa. Con esto vimos que el shell bloqueaba su ejecución hasta que el comando invocado terminaba, antes de volver a mostrar la línea de ingreso. Después hicimos lo mismo, pero sin system().

En conjunto, estos programas nos hicieron retarnos sobre lo que hemos aprendido, también utilizar ejemplos de las clases para poder resolver nuestra actividad, al igual que aplicando la lógica de los procesos, la sincronización y la jerarquía que existe entre padre e hijos. También investigamos cómo interpretar comandos en sistemas y comprendimos mejor el funcionamiento de un shell en Unix.