



Universidad Nacional Autónoma de México
Facultad de Ingeniería



Práctica 02

Procesos

Ing. Computación
Sistemas operativos
M.I Edgar Tista García

Elaborado por: Martínez Soto Mariana

Objetivo

Implementar de manera práctica los conceptos del curso relacionados con procesos realizando un programa donde sea capaz de simular la ejecución de procesos en un sistema operativo.
Que el alumno pueda retomar sus habilidades aplicándolo a un concepto fundamental del curso.

Introducción

Un proceso se puede definir de diferentes formas ya que es una de las unidades fundamentales de un trabajo de un sistema operativo.

Un proceso básicamente es un programa que se encuentra en ejecución, aunque de manera más formal se puede definir como cualquier secuencia de operaciones que está ejecutándose en memoria principal que realiza una o varias acciones sobre ciertos datos.

La estructura de un proceso cuenta con 4 elementos básicos:

- Sección de texto
- Sección de datos
- Pila
- Heap

El estado de un proceso depende de la actividad que esté realizando

- Nuevo: El proceso está siendo creado
- En ejecución: El cpu ejecuta las instrucciones
- Espera: Se encuentra en un estado que depende de un suceso externo
- Preparado: Listo para que se le asigne un procesador
- Terminado: Ha concluido la ejecución

Solo puede haber un proceso ejecutándose en cualquier procesador en un instante de tiempo pero puede haber muchos proceso preparados y es espera

Informe

Uno de los primeros pasos importantes para la realización de esta práctica fue comprender los conceptos sobre el funcionamiento, además como analizar que opciones serían mejor para representar algunas de las partes de un sistema operativo durante la implementación de procesos.

En primera instancia es importante entender que el sistema operativo es el que realiza dicha planificación, es decir, el que decide que va a suceder con cada proceso nuevo, en ejecución, en espera o finalizado, y que tiene el control total sobre ellos, y en este caso las decisiones que toma son las que toma el usuario. En otras palabras, que lo que el usuario decida hacer, ya sea eliminar, pasar o mover sería lo que en realidad el S.O debe analizar y realizar por su cuenta.

Otro punto importante es que la creación de procesos se mantiene en una cola principal, por lo que utilicé la clase ArrayList que me permite manejar con mayor facilidad los objetos que se guardan en ella, en este caso éstos últimos son los procesos los que nos interesa almacenar temporalmente en ésta queue. Al principio había decidido usar PriorityQueue, sin embargo la limitante fue que necesitaba indicar una comparación entre elementos por algún atributo que los distinguiera; Para el caso real, es decir que para lo que hace el S.O en la realidad hubiera sido muy bueno, ya que sí se utiliza la prioridad en los procesos, pero para los fines de la práctica, como no se requirió ninguna ventaja entre objetos me dediqué al uso de una cola común, sin prioridad.

Cabe aclarar que los procesos implementados no cambian su espacio de memoria, y que las localidades, para el caso en cuestión, no se liberan hasta terminadas completamente sus instrucciones. La diferencia que no lograba distinguir al principio era entre la cola de los procesos y la memoria siendo éstos dos entidades diferentes. De hecho podría decirse que los procesos reales son los que se encuentran en la cola, mientras que los que están en memoria, ocupando espacio son imágenes de dichos procesos, que no se están ejecutando precisamente por estar en memoria, sino más bien apartando su respectivo espacio, hasta que el S.O con su control permite ejecutar a través de la cola los procesos.

Conclusiones

El sistema operativo debe implementar de la forma más óptima posible la creación, ejecución y finalización de los procesos. Uno de los puntos importantes es desperdiciar la menor cantidad de memoria posible; En el curso anterior de EYPC se abordaron distintas formas de asignación de memoria, las cuales creo que juegan muy importante en este tema, sin embargo en la práctica solo poner en práctica la asignación de memoria contigua o en bloques consecutivos. Sobre los demás puntos parece funcionar correctamente, aunque sí, el espacio en memoria queda desperdiciado cuando uno o muchos de los procesos primeros finaliza.

El hecho de analizar con qué elementos se puede simular la actividad de planeación es un punto importante para la comprensión de su funcionamiento. De lo contrario no se puede avanzar mucho. En lo personal el algoritmo que debe seguir cualquier programa es esencial, porque en muchas ocasiones los errores no se generan por cuestiones de programación, sino por la forma en la cuál está estructurado el flujo. Y el hecho de separar mi programa en dos clases importantes, que es programa y memoria me dió más claridad.

Para finalizar me gustaría decir que la documentación de cualquier lenguaje será siempre una de las mejores herramientas que podamos usar como programadores, y por ellos la importancia de recurrir a ella más seguido. Mi elección con Java fue porque su framework es bastante amigable y precisamente está hecha para el nivel usuario. Creo que dependiendo el contexto cada programa tiene más valor que otros, en este caso lo consideré un camino relativamente práctico.