**Tutorial 24**

En este tutorial se aprenderá a manejar procesos ya en nuestro sistema operativo. Cuando un sistema operativo es capaz de ejecutar otro código dentro de si, ya se puede volver auto sostenible y supone un gran paso puesto que ya se puede desarrollar dentro de si mismo. Para esto es necesario definir varios componentes del manejo de procesos. Un proceso no es nada mas que un programa en ejecución en el procesador y que tiene un hilo (thread) en ejecución, siendo sí mismo. IPC o InterProcess Communication es una forma de implementación de comunicación entre procesos. Ya se han visto varios que pueden ser “pipes”, archivos compartidos, paso de mensajes, semáforos, señales, sockets, etc. Cada implementación de IPC puede ser distinta pero es esencial en cada sistema operativo.

Para poder empezar a cargar programas y crear procesos, es necesario también proteger estos procesos de otros procesos en si. Un proceso no debe poder corromper la memoria y datos de otro proceso en memoria ni tampoco puede acceder a memoria reservada para el kernel. Para esto, se implementa la memoria virtual, la cual asigna a cada proceso un espacio de memoria “infinito” y el kernel es el encargado de poder asignar adecuadamente un espacio en memoria real a cada proceso. Los programas contienen diferentes secciones que son creados por el linker y estos son en general : .text, .data, .bss y .rodata. La sección .text contiene todo el código ejecutable y corresponde al code segment. .data contiene todas las variables globales y estaticas. La sección .bss contiene la sección de variables nulas. Y por ultimo, la sección .rodata contiene data de variables de solo lectura.

Cuando el kernel crea un proceso, este le asigna un espacio de stack para todas sus variables y este es un tamaño fijo y también asigna una dirección y espacio de heap. El heap puede cambiar dinámicamente cuando el proceso requiera de mas espacio en heap pero el stack se mantiene igual siempre. El sistema operativo debe implementar un Program Control Block el cual contiene el nombre del proces, PID, tiempo de ejecución requerido en CPU o prioridad y nivel de protección de ejecución. Esta estructura depende de cada sistema operativo pero es esencial poder tener esta información en una lista o tabla para poder reconocer los proceso ejecutándose en nuestro sistema. Una vez que un proceso es cargado en memoria y tengamos esta información, podremos alocar el espacio de memoria virtual y espacio en memoria real y simplemente hacer un jump hacia la rutina de entrada del proceso para poder ejecutar el programa. Como no se cuenta con un planificador o Task Scheduler, el sistema operativo solo será capaz de ejecutar un proceso a la vez pero ya es un avance.