

## Síntesis

Como ya es común, los sistemas operativos son capaces de correr diferentes programas al mismo tiempo. Esto lo logran con una eficiente administración de los procesos de cada programa. Para lograrlo, en sistemas operativos como Linux se han definido estructuras como la `task_struct`, que contiene información esencial del proceso de referencias a los procesos adyacentes con el fin de una eficiente admiración de estos. Además, se cuenta con diferentes llamadas al sistema y funciones del Kernel para el ciclo de vida de un proceso. Los mecanismos adyacentes son similares tanto para el contexto de usuario como de Kernel, en ambos casos se termina ejecutando la función `do_fork` del Kernel que internamente se encarga de reservar un nuevo PID (id de proceso), memoria, copiar información del proceso padre e inicialización del nuevo proceso. La situación para la eliminación de procesos es similar, en ambos casos se termina llamada a la función `do_exit` que se encarga de cambiar ciertas banderas de estado del proceso, liberación de memoria, recursos y finalmente eliminación del proceso en cuestión.

## Preguntas

**¿Qué es el process id, para que se usa y donde se almacena?**

Desde la perspectiva del usuario el process id es un número, almacenado en la caché, que identifica de forma única a un proceso. Este no cambia durante la vida de un proceso, pero se pueden reutilizar después que un proceso muere.

**¿Qué información tipo de información se puede obtener de task\_struct con respecto al uso de recursos del computador?**

Se puede saber procesos adyacentes con el campo **tasks**, debido a que provee una lista enlazada con un puntero a la tarea anterior y otro a la siguiente. Los campos como **mm** y **active\_mm** nos dice el espacio de direcciones de memoria, si son direcciones altas esto indica que se está ocupando bastante espacio en la memoria.

**Enumere las funciones de Kernel involucradas en la creación de procesos**

La función para crear procesos es `do_fork`, esta a su vez hace uso de:

1. `alloc_pidmap`
2. `copy_process`
3. `dup_task_struct`
4. `copy_files`

5. `copy_sighand`
6. `copy_signal`
7. `copy_mm`
8. `copy_thread`
9. `cpus_allowed`
10. `wake_up_new_task`

**¿Explique un cambio recién del Kernel de Linux para mejorar el manejo de procesos, coloque referencia?**

Reciente mente se ha añadido un nuevo campo en `task_struct`, que son `mm` y `active_mm`. Como menciona la página [Kernel.org](https://lwn.net/Articles/lwn2019-05-01), el primero representa los descriptores de memoria del proceso actual, mientras que `active_mm` los descriptores de memoria del proceso previo. Esto mejorar los cambios de contexto en los procesos.

## Vocabulario

- **Thread:** según Computer Hope, un thread (hilo) es un pequeño set de instrucciones diseñada para ser administradas y ejecutadas por el CPU independientemente del proceso padre.
- **Debugger:** según Computer Hope, es un software de utilidad diseñado con el fin de localizar errores en el código fuente de programas.
- **File Descriptors:** según Computer Hope, es un número único que identifica a un archivo abierto en el sistema operativo de la computadora.
- **Process:** según TechTerms, un proceso es un programa, o una parte de él, que esta corriendo en la computadora.
- **Memory pages:** según Computer Hope, se refiere a una cantidad de memoria almacenada, lo que permite un uso no contiguo del espacio de direcciones físicas. Al acceder a la memoria en la misma fila, la computadora mantiene la dirección de la fila y solo cambia la columna.
- **Flags:** según TechTerms, es un valor que actúa como señal para una función o proceso, para cambiar el comportamiento de este o dar mas información de su estado.
- **Pointer:** según Computer Hope, un puntero comúnmente se refiere a una referencia hecha a una dirección en la memoria.
- **Address Space:** según Computer Hope, es un rango lógico de espacio de alguna parte de la computadora o hardware donde se almacena información.

- Hash: según TutotialsPoint, es una estructura de datos que almacena datos de manera asociativa. Cada valor de la tabla tiene asociada una clave única, esta clave también se usa para ubicar al dato en la tabla.
- Task: según Computer Hope, se utiliza para describir un programa de software o una sección de un programa que se ejecuta en un entorno multitarea.

## Referencias

(5 de Marzo de 2019). Obtenido de Computer Hope:

<https://www.computerhope.com/jargon/t/thread.htm>

*Computer Hope*. (s.f.). Obtenido de <https://www.computerhope.com/jargon/p/pointer.htm>

*Computer Hope*. (s.f.). Obtenido de <https://www.computerhope.com/jargon/a/address-space.htm>

*Computer Hope*. (s.f.). Obtenido de <https://www.computerhope.com/jargon/t/task.htm>

*Computer Hope*. (17 de Octubre de 2017). Obtenido de <https://www.computerhope.com/jargon/t/task.htm>

*Computer Hope*. (4 de Mayo de 2017). Obtenido de <https://www.computerhope.com/jargon/d/debugger.htm>

*Computer Hope*. (11 de Noviembre de 2019). Obtenido de <https://www.computerhope.com/jargon/f/file-descriptor.htm>

*Computer Hope*. (5 de Abril de 2020). Obtenido de <https://www.computerhope.com/jargon/p/paging.htm>

*Kernel.org*. (s.f.). Obtenido de [https://www.kernel.org/doc/html/latest/vm/active\\_mm.html](https://www.kernel.org/doc/html/latest/vm/active_mm.html)

*TechTerm*. (s.f.). Obtenido de <https://techterms.com/definition/process>

*TechTerms*. (s.f.). Obtenido de <https://techterms.com/definition/flag>

*TutorialsPoint*. (s.f.). Obtenido de [https://www.tutorialspoint.com/data\\_structures\\_algorithms/hash\\_data\\_structure.htm](https://www.tutorialspoint.com/data_structures_algorithms/hash_data_structure.htm)