

Centro Universitario de Ciencias Exactas e Ingenierías

DIVISIÓN DE TECNOLOGÍAS PARA LA INTEGRACIÓN CIBER-HUMANA

Seminario de Solución de Problemas de Uso, Adaptación, Explotación de

Sistemas Operativos

Profesora: Violeta del Rocío Becerra Velázquez

Ingeniería Informática

Sección: D02

Conociendo los procesos, hilos y su entorno

Equipo:

- Andrea Guadalupe Herrera Espinoza

Código: 215662794

- José Pedro Reyes Álvarez

Código: 222790897

- José Roberto Escobedo Quezada

Código: 213548145

16 / 03 / 2025

I. Introducción	3
II. Contenido	3
1.1. Definición de procesos y su jerarquía	3
1.2. Estados de un proceso	5
1.3. Descripción de los procesos y sus elementos	6
1.4. Control de los procesos (planificación).	9
1.5. Definición de hilo	11
1.6. Hilo vs. Proceso	14
1.7. Gestión de Hilos	15
III. Conclusión.....	17
IV. Bibliografía.....	17

I. Introducción

El proceso se puede definir como un programa en ejecución. El sistema operativo mantiene por cada proceso una serie de estructuras de información que permiten identificar las características de éste, así como los recursos que tiene asignados. Una parte muy importante de estas informaciones se encuentra en el llamado bloque de control del proceso (BCP). Por razones de eficiencia, la tabla de procesos se construye normalmente como una estructura estática, que tiene un determinado número de BCP, todos ellos del mismo tamaño.

Cada proceso se encuentra, en un instante dado en uno de los estados de ejecución que incluyen Listo, Ejecución y Bloqueado. El sistema operativo sigue el camino de estos estados de ejecución y gestiona el movimiento. Además, debe realizar operaciones de planificación y proporcionar servicios para la compartición y sincronización entre procesos. Por lo que esta actividad centra sus actividades en los puntos importantes de la gestión de procesos.

II. Contenido

1.1. Definición de procesos y su jerarquía

Un proceso es un programa en ejecución, el cual requiere de recursos del sistema para ser ejecutado, entre los cuales se incluyen:

- Tiempo de CPU.
- Memoria.
- Archivos.
- Dispositivos de Entrada/Salida.

Entre otros.

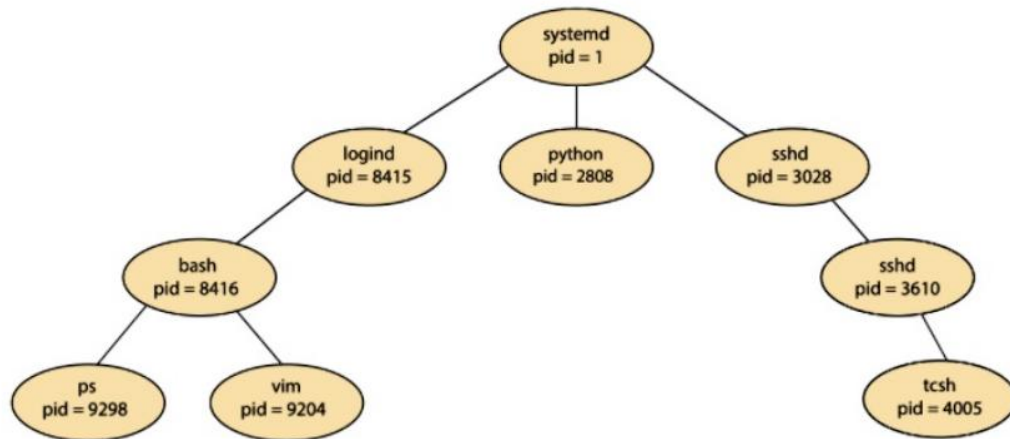
Puesto de forma simple, un programa está escrito en un archivo de texto que, al ser ejecutado se convierte en proceso que realiza todas las tareas mencionadas en él.

Una vez que el programa pasa a ser un proceso puede dividirse en cuatro secciones: La pila (Stack), el Heap (Montículo o memoria dinámica), texto y datos, los cuales son descritos completamente en la sección “Descripción de un proceso y sus elementos”.

No obstante, los procesos en los sistemas operativos modernos pueden crear otros procesos con los que se relacionan directamente, creando una estructura jerárquica compleja entre ellos, que se asemeja a la de un árbol.

Siguiendo los conceptos estandarizados de conceptualización de árboles como método descriptivo, puede decirse que un proceso padre (nodo padre del árbol) puede crear procesos hijos (nodos hijos), que a su vez, pueden crear otros procesos hijos.

Lo anterior puede ejemplificarse en la imagen a continuación, que muestra un árbol de proceso en un sistema Linux común:

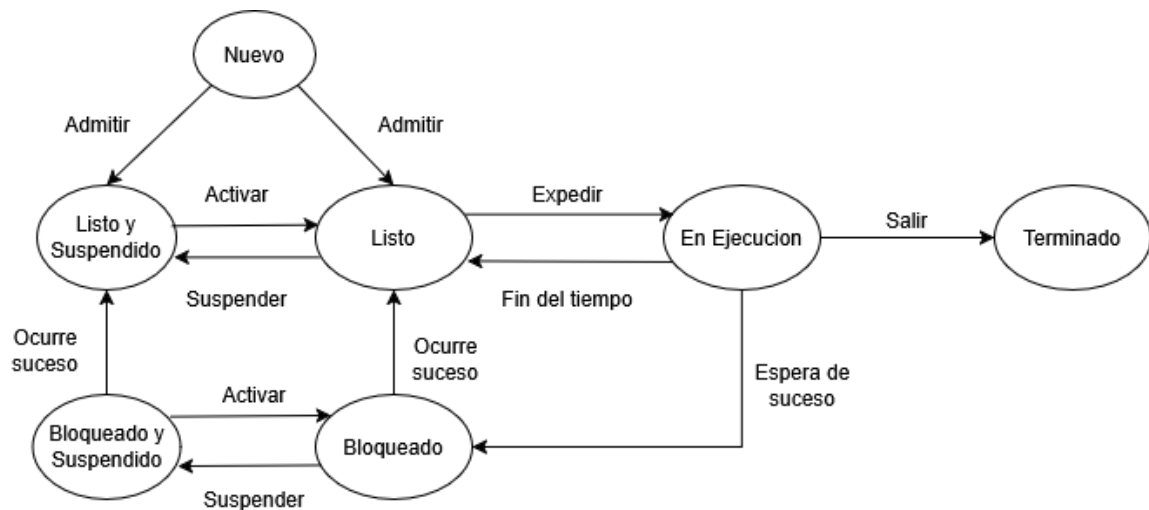


Por ende, en un árbol de jerarquía de proceso es común encontrarse con distintos tipos de proceso:

- **Proceso padre.** Proceso original que crea otros procesos.
- **Proceso hijo.** Creado por otro proceso (su padre). Puede heredar recursos del padre.
- **Proceso huérfano.** Un proceso cuyo padre ha terminado antes de que el hijo terminase su ejecución.
- **Proceso zombi.** Un proceso que ha terminado, pero cuya entrada en la tabla de procesos no ha sido eliminada porque el padre aún no ha leído su estado de terminación.

1.2. Estados de un proceso

a) Diagrama de estado.



b) Definición de cada estado y sus transiciones válidas.

Cada proceso posee un estado, que pueden ejemplificarse como nuevo, listo, en ejecución, bloqueado (o en espera) y terminado.

En el caso de un proceso nuevo, se hace referencia al estado inicial de un programa que recién ha sido creado o iniciado.

Los procesos en estado de “Listo”, están a la espera de ser asignados a un procesador. Pueden llegar a este estado después de haber pasado por el estado de “Nuevo” o tras ser interrumpidos por el asignador de proceso, luego de haber estado en ejecución, por necesidad de ejecutar procesos más importantes en el procesador.

Por otro lado, el estado “En ejecución”, ocurre cuando los procesos ya han sido asignados a un procesador por el organizador (también llamado asignador) del proceso y sus instrucciones están siendo ejecutadas.

El estado “Bloqueado” (también conocido como “En espera”) puede darse cuando un proceso requiere de esperar a un recurso que aún no posee, ejemplos de recursos comunes que llevan a este estado son la entrada de datos de un usuario, conexión con un dispositivo externo o un archivo que no está disponible en el momento por cualquier razón.

Por último, el estado “Terminado” ocurre cuando un proceso ha terminado su ejecución o ha sido finalizado por el sistema operativo. En este punto, ya solo es necesario que el proceso sea liberado de la memoria para que termine su ciclo de ejecución.

1.3. Descripción de los procesos y sus elementos

Un proceso está compuesto por varias partes que resultan fundamentales para su ejecución y gestión por parte del sistema operativo:

Código del programa.

Contiene las instrucciones ejecutables del programa en un archivo de texto ejecutable. Puede ser simplemente una sola línea de código o millones de ellas. Como se sabe, un programa de computadora es escrito por un programador en un lenguaje de programación.

Por ejemplo, a continuación se muestra un programa sencillo escrito en el lenguaje de programación C:

```
#include <stdio.h>

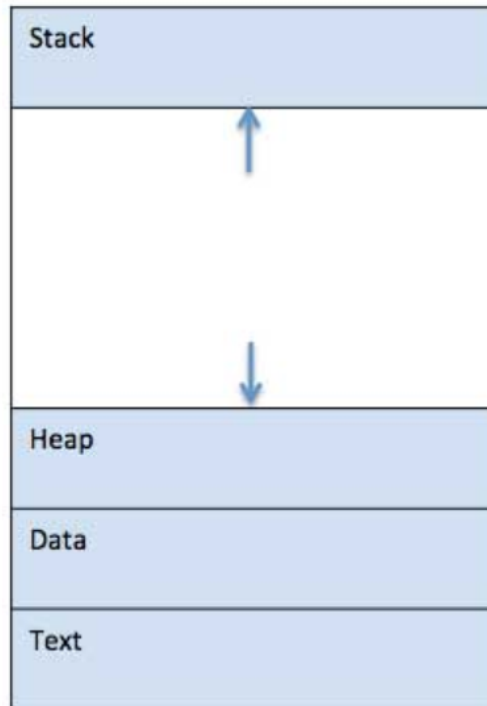
int main() {
    printf("Hello, World! \n");
    return 0;
}
```

Dentro de estos programas, se encuentra una colección de instrucciones que realizan una tarea específica al ser ejecutadas por la computadoras.

En este sentido, puede hacerse la aclaración de que un proceso es una instancia dinámica de un programa.

Memoria del proceso.

Que almacena distintas partes del proceso en secciones conocidas como la sección de datos, texto, Heap y la Pila y que puede observarse de manera gráfica en la siguiente imagen:

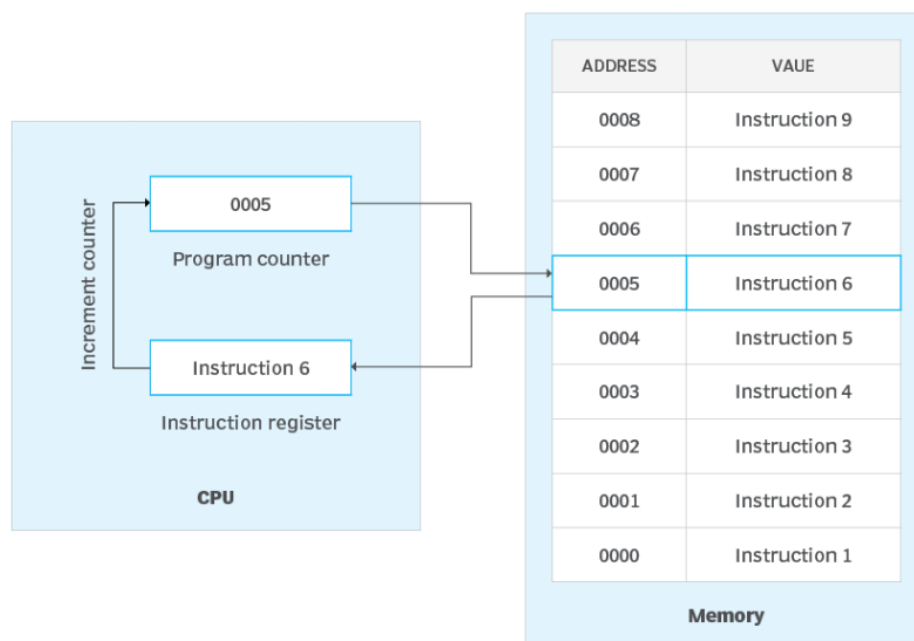


Cada sección puede definirse como:

- **Stack.** Contiene datos temporales como parámetros de métodos y/o funciones, direcciones y variables locales.
- **Heap.** Se trata de la memoria en donde el proceso se almacena de manera dinámica durante su tiempo de ejecución solamente.
- **Texto.** Incluye la actividad actual representada por el valor del contador del programa y los registros del procesador.
- **Datos.** Esta sección contiene las variables estáticas y globales requeridas por el sistema.

Contador del programa (PC - Program Counter).

Indica la dirección de la siguiente instrucción a ejecutar, de manera que la estructura y secuencia de ejecución de procesos padres e hijos pueda realizarse de manera adecuada.



Registros del CPU.

Guardan información del estado actual del proceso, como registros de uso general, registros de pila y registros de estado.

Estado del proceso.

Que puede ser Nuevo, Listo, En ejecución, Bloqueado o Terminado. Estos ya se han descrito anteriormente en la sección “Estados de un proceso”.

Estructura de control del Proceso (PCB - Process Control Block).

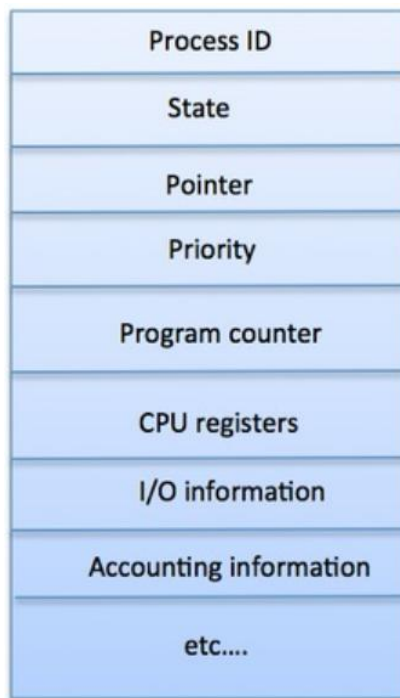
Es una estructura de datos del sistema operativo que almacena toda la información necesaria para gestionar el proceso.

Identificado por un id de proceso (conocido como PID), el PCB posee la siguiente información:

- Estado de proceso.
- Privilegios de acceso y modificación de proceso.
- ID de proceso.
- Puntero a procesos padres.
- Contador del programa.

- Registros del CPU.
- Información de organización dentro del CPU, que incluyen su orden de prioridad, entre otros datos.
- Información de organización de memoria.
- Información de recursos necesarios para que el proceso sea ejecutado.
- Información de estado de dispositivos de entrada/salida.

Esta estructura varía en forma y tamaño dependiendo del sistema operativo que la gestione, no obstante, a continuación se muestra un diagrama PCB general simplificado:



1.4. Control de los procesos (planificación).

Cuando se trata de sistemas de tarea única o por lotes, la gestión de procesos es sencilla porque solo hay un proceso activo a la vez. Sin embargo, en sistemas multiprogramados o multitarea, la gestión se vuelve más compleja porque varios procesos deben compartir recursos y sincronizarse adecuadamente.

De manera general, puede decirse que gestión de procesos es clave en sistemas operativos multitarea y el flujo de proceso realizado para dicha gestión, es el siguiente:

1. **Creación y terminación de procesos.** Se asigna un ID y un PCB (Process Control Block). Los procesos pueden ser terminados por el sistema o el proceso padre.

2. **Planificación de la CPU:** El sistema operativo decide qué proceso ejecutará la CPU para garantizar eficiencia.
3. **Manejo de interbloqueos (Deadlocks):** Previene situaciones donde los procesos quedan bloqueados esperando recursos entre sí.
4. **Comunicación entre procesos (IPC):** Se usan memoria compartida y paso de mensajes para la cooperación entre procesos.
5. **Sincronización de procesos:** Garantiza acceso controlado a recursos compartidos para evitar condiciones de carrera (race conditions), es decir, escenarios donde dos procesos acceden de manera simultánea a un recurso compartido y pueden producir resultados erróneos a causa de ello.

Los sistemas operativos actuales pueden llevar a cabo la gestión de procesos mediante distintos algoritmos de organización.

A continuación se mencionan algunos de los algoritmos de planificación de procesos más importantes.

FCFS - First-Come, First-Served (Primero en entrar, primero en servirse)

Es el algoritmo de organización más simple en donde los procesos son ejecutados en orden de llegada, simulando el comportamiento de una cola. Dicho de otra forma, los procesos son despachados mediante el concepto de Primero en Entrar, Primero en Salir (FIFO por sus siglas en inglés).

Es importante mencionar que en este algoritmo un proceso no puede interrumpirse bajo ningún motivo hasta que termina completamente su ejecución.

SJF - Shortest Job First (Tiempo de ejecución más corto primero)

Es un algoritmo de planificación proactivo que selecciona el proceso con el tiempo de ejecución más corto. Este proceso minimiza el tiempo promedio de espera de cada proceso.

RR - Round Robin

Es un algoritmo de planificación proactivo que permite la ejecución de procesos por bloques de tiempo determinados por cada “ronda” de ejecución, a dichos bloques de

ejecución se les conoce como “Quantums”. Si un proceso no termina su ejecución en el bloque de tiempo establecido, es bloqueado y se le coloca al final de la cola de espera de procesos.

Este algoritmo asegura una distribución de tiempo de uso de CPU equitativa entre todos los procesos.

Priority Scheduling (Planificación por prioridad)

Este algoritmo asigna una prioridad a cada proceso, de manera que el proceso con la prioridad más alta sea el que se debe ejecutar primero.

La prioridad puede ser dada de acuerdo con el tipo de proceso, importancia o recursos requeridos.

Multilevel Queue (Cola multinivel)

Este algoritmo divide la cola de procesos con estado “Listo” en varias colas distintas, en donde cada cola tiene un nivel de prioridad diferente. Los procesos son organizados en las colas dependiendo de su prioridad, y cada cola usa su propio sistema de planificación.

Este algoritmo de planificación resulta útil en escenarios donde diferentes tipos de procesos, poseen diferentes prioridades.

1.5. Definición de hilo

De la misma manera que un Sistema Operativo puede ejecutar varios procesos al mismo tiempo bien sea por concurrencia o paralelismo, dentro de un proceso puede haber varios hilos en ejecución. Por lo tanto, un hilo, o también llamado subproceso, es básicamente un flujo de ejecución o una secuencia de instrucciones manejada de forma independiente. Un hilo vive dentro del contexto de un proceso y este utiliza todos sus recursos; como por ejemplo, la memoria o los registradores.

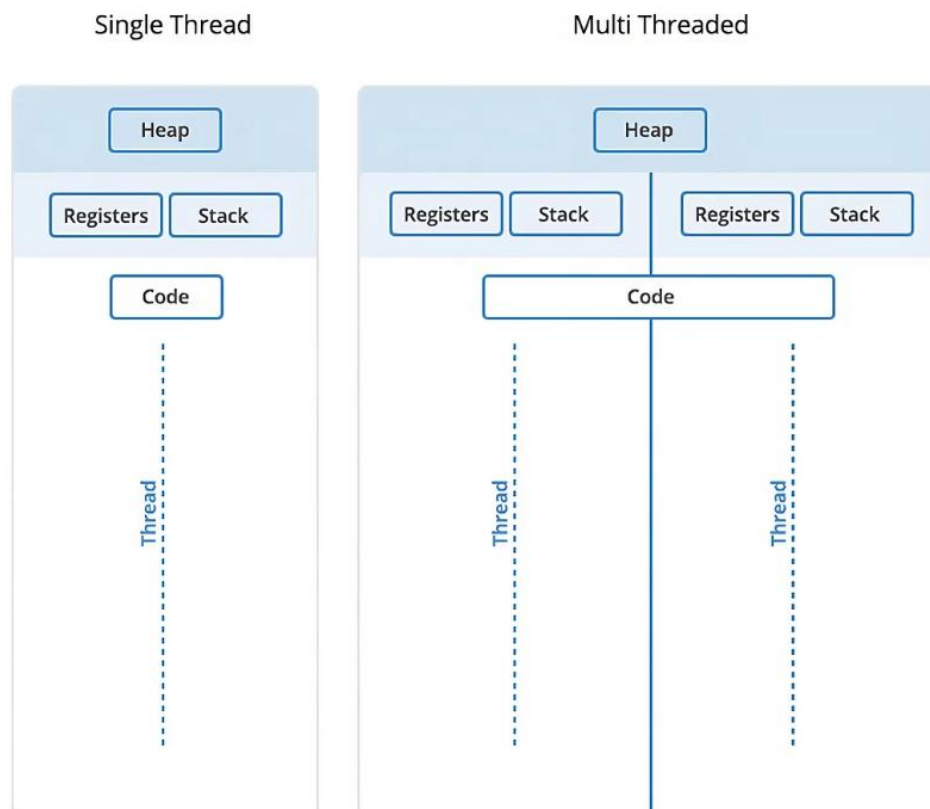
Debemos destacar que si un hilo, altera un elemento de memoria de segmento, todos los demás hilos se percatan de aquello (en el caso de multihilo).

¿Qué es un Multihilo?

Multihilo o también conocido como multithreading, se refiere a la capacidad de un sistema o programa de ejecutar múltiples hilos de forma concurrente dentro de un proceso.

En un entorno de ejecución de un solo hilo, el programa se ejecuta secuencialmente, es decir, una instrucción se ejecuta después de la otra y en orden. En cambio, en un entorno de ejecución multihilo, el programa puede dividirse en múltiples hilos, cada uno de los cuales se ejecuta simultáneamente y realiza un conjunto específico de tareas.

Los hilos comparten recursos y memoria comunes dentro de un proceso, lo que les permite comunicarse y cooperar entre sí. Cada hilo tiene su propio contador de programa, pila de ejecución y variables locales, pero también pueden acceder a variables compartidas y recursos compartidos dentro del proceso.



○

Características de los hilos:

- Los hilos son unidades de ejecución ligeras en comparación con los procesos. La creación, la finalización y el cambio de contexto entre hilos son más rápidos y requieren menos recursos que los procesos.

- Los hilos pueden ejecutarse concurrentemente, lo que significa que múltiples hilos pueden estar en ejecución al mismo tiempo dentro de un proceso. Esto permite aprovechar mejor los sistemas multiprocesador o multinúcleo, puesto que, los hilos pueden distribuirse en los diferentes núcleos de procesamiento.

En la figura 5 puede verse cómo sobre el hardware subyacente (una o varias CPU's) se sitúan el Sistema Operativo. Sobre éste se sitúan los procesos (P_i) que pueden ejecutarse concurrentemente y dentro de estos se ejecutan los hilos (h_i) que también pueden ejecutarse de forma concurrente dentro del proceso. Es decir, tenemos concurrencias a dos niveles, una entre procesos y otra entre hilos de un mismo proceso. Si por ejemplo tenemos dos procesadores, se podrían estar ejecutando al mismo tiempo el hilo 1 del proceso 1 y el hilo 2 del proceso 3. Otra posibilidad podría ser el hilo 1 y el hilo 2 del proceso 1.

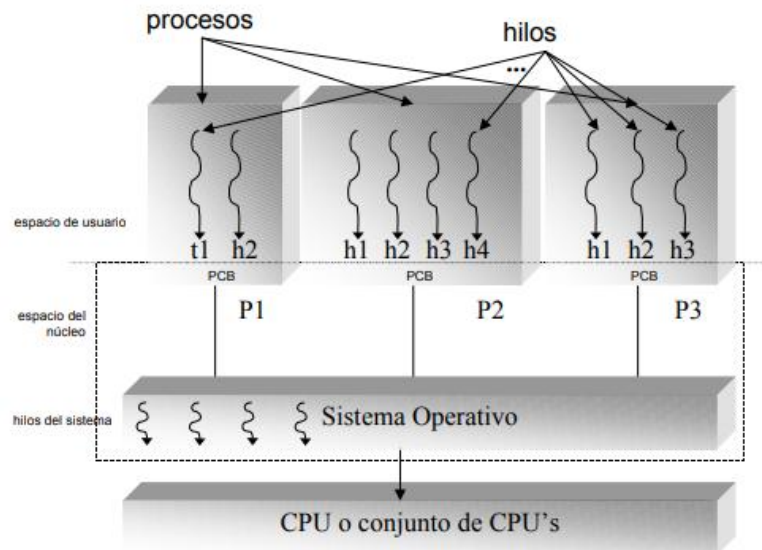


Figura 5 Concurrencia a dos niveles: procesos e hilos.

Estados de un hilo

Los principales estados de un hilo son: ejecución, preparado y bloqueado; y hay cuatro operaciones básicas relacionadas con el cambio de estado de los hilos:

Creación: En general, cuando se crea un nuevo proceso se crea también un hilo para ese proceso. Posteriormente, ese hilo puede crear nuevos hilos dándoles un puntero de instrucción y algunos argumentos. Ese hilo se colocará en la cola de preparados.

Bloqueo: Cuando un hilo debe esperar por un suceso, se le bloquea guardando sus registros. Así el procesador pasará a ejecutar otro hilo preparado.

Desbloqueo: Cuando se produce el suceso por el que un hilo se bloqueó pasa a la cola de listos.

Terminación: Cuando un hilo finaliza, se liberan su contexto y sus pilas.

Un punto importante es la posibilidad de que el bloqueo de un hilo lleve al bloqueo de todo el proceso. Es decir, que el bloqueo de un hilo lleve al bloqueo de todos los hilos que lo componen, aún cuando el proceso está preparado.

1.6. Hilo vs. Proceso

Diferencias importantes entre hilo y proceso

Parámetro	Proceso	Hilo
Definición	Proceso significativo que un programa está en ejecución.	Hilo significa un segmento de un proceso.
Ligeros	El proceso no es ligero.	Los hilos son ligeros.
hora de terminación	El proceso tarda más en finalizar.	El hilo tarda menos en terminar.
Tiempo de creación	Se necesita más tiempo para la creación.	Se necesita menos tiempo para la creación.
Comunicación	La comunicación entre procesos necesita más tiempo en comparación con los subprocesos.	La comparación entre subprocesos requiere menos tiempo en comparación con los procesos.
Tiempo de cambio de contexto	Se necesita más tiempo para cambiar de contexto.	Se necesita menos tiempo para cambiar de contexto.
Soporte Envolvente	El proceso consume más recursos.	Los subprocesos consumen menos recursos.
Tratamiento por Sistema Operativo	El sistema operativo realiza diferentes procesos por separado.	El sistema operativo trata todos los subprocesos del mismo nivel como una sola tarea.

Salud Cerebral	El proceso es mayoritariamente aislado.	Los hilos comparten memoria.
Compartir	No comparten datos.	Los hilos comparten datos entre sí.

Semejanzas: Los hilos operan, en muchos sentidos, igual que los procesos.

- Pueden estar en uno o varios estados: listo, bloqueado, en ejecución o terminado.
- También comparten la CPU.
- Sólo hay un hilo activo (en ejecución) en un instante dado.
- Un hilo dentro de un proceso se ejecuta secuencialmente.
- Cada hilo tiene su propia pila y contador de programa.
- Pueden crear sus propios hilos hijos.

Diferencias: Los hilos, a diferencia de los procesos, no son independientes entre sí.

- Como todos los hilos pueden acceder a todas las direcciones de la tarea, un hilo puede leer la pila de cualquier otro hilo o escribir sobre ella. Aunque pueda parecer lo contrario la protección no es necesaria ya que el diseño de una tarea con múltiples hilos tiene que ser un usuario único.

Ventajas: de los hilos sobre los procesos.

- Se tarda mucho menos tiempo en crear un nuevo hilo en un proceso existente que en crear un nuevo proceso.
- Se tarda mucho menos tiempo en terminar un hilo que un proceso.
- Se tarda mucho menos tiempo en conmutar entre hilos de un mismo proceso que entre procesos.
- Los hilos hacen más rápida la comunicación entre procesos, ya que al compartir memoria y recursos, se pueden comunicar entre sí sin invocar el núcleo del SO.

1.7. Gestión de Hilos

La gestión de hilos es la capacidad del sistema operativo para controlar la creación, sincronización y terminación de los hilos. Los principales aspectos de la gestión de hilos incluyen:

1. Creación de Hilos

- Se pueden crear hilos dentro de un proceso utilizando diferentes métodos, como las bibliotecas de hilos (POSIX Threads en UNIX, Java Threads en Java, etc.).
- En sistemas operativos modernos, cada proceso puede contener múltiples hilos.

2. Planificación de Hilos

- Similar a la planificación de procesos, el sistema operativo debe asignar tiempo de CPU a cada hilo.
- Existen dos tipos de planificación:
 - Planificación a nivel de usuario: La gestión de hilos es realizada por la aplicación sin intervención del sistema operativo.
 - Planificación a nivel de kernel: El sistema operativo gestiona los hilos directamente.

3. Sincronización de Hilos

- Cuando varios hilos comparten recursos, se pueden generar problemas de condiciones de carrera.
- Para evitarlo, se utilizan mecanismos de sincronización como:
 - Mutexes: Permiten que solo un hilo acceda a un recurso a la vez.
 - Semáforos: Controlan el acceso a múltiples recursos.
 - Variables de condición: Permiten la comunicación entre hilos.

4. Comunicación entre Hilos

- Dado que los hilos comparten memoria, pueden comunicarse de manera eficiente sin necesidad de mecanismos complejos como tuberías o memoria compartida entre procesos.

5. Terminación de Hilos

- Un hilo puede terminar cuando:
 - Completa su ejecución.
 - Es cancelado por otro hilo o por el proceso padre.
 - Se produce un error o una excepción no manejada.
- Algunos sistemas operativos permiten la terminación controlada de hilos mediante funciones específicas.

III. Conclusión

En esta investigación donde se llevó a cabo el estudio de los procesos, hilos y su gestión dentro de los sistemas operativos me parece que es fundamental para comprender el funcionamiento eficiente de un sistema informático. Los procesos representan la ejecución de programas y requieren una gestión adecuada para optimizar el uso de recursos, garantizar la estabilidad del sistema y mejorar el rendimiento general.

Por otro lado, los hilos permiten una ejecución más ágil y concurrente dentro de un mismo proceso, lo que facilita la multitarea y el aprovechamiento de arquitecturas multiprocesador. La implementación de algoritmos de planificación y sincronización de procesos e hilos es clave para evitar problemas como condiciones de carrera y bloqueos, asegurando una ejecución eficiente de tareas.

IV. Bibliografía

Operating System - Processes. (s. f.). Tutorialspoint. Recuperado 15 de marzo de 2025, de https://www.tutorialspoint.com/operating_system/os_processes.htm

Concepto de proceso. (s. f.). Recuperado 15 de marzo de 2025, de <https://misistema-operativo.blogspot.com/p/unidad-2.html>

Process Management - Part V. (s. f.). Recuperado 15 de marzo de 2025, de <https://www.gchamirpur.org/wp-content/uploads/2023/07/Unit-II-Lecture-8-Process-Management-Part-V.pdf>

Sheldon, R. (s. f.). What is a program counter? TechTarget. Recuperado 15 de marzo de 2025, de <https://www.techtarget.com/whatis/definition/program-counter>

Introduction of Process Management. (2024, 30 diciembre). GeeksforGeeks. Recuperado 15 de marzo de 2025, de <https://www.geeksforgeeks.org/introduction-of-process-management/>

D. (2023, Junio 22). Entendiendo los procesos, hilos y multihilos. Medium. <https://medium.com/@diego.coder/entendiendo-los-procesos-hilos-y-multihilos-9423f6e40ca7>

Cerón, M. (s.f.). *Capítulo 2: Diseño de sistemas operativos*. Benemérita Universidad Autónoma de Puebla. Recuperado de https://www.cs.buap.mx/~mceron/cap2_dis.pdf

Procesos vs Hilos. (n.d.). Blogspot.com. Retrieved March 16, 2025, from <https://systope.blogspot.com/2012/05/procesos-e-hilos.html>

Brooks, N. (2024, Agosto 12). Proceso versus hilo: diferencia entre ellos. Guru99.
<https://www.guru99.com/es/difference-between-process-and-thread.html>

González, F. J. (s.f.). Tema 4: Gestión de procesos e hilos. Universidad de Valladolid. Recuperado de <https://www.infor.uva.es/~figonzalez/apuntes/Tema4.pdf>