**Procesos de los sistemas operativos**

**Procesos**

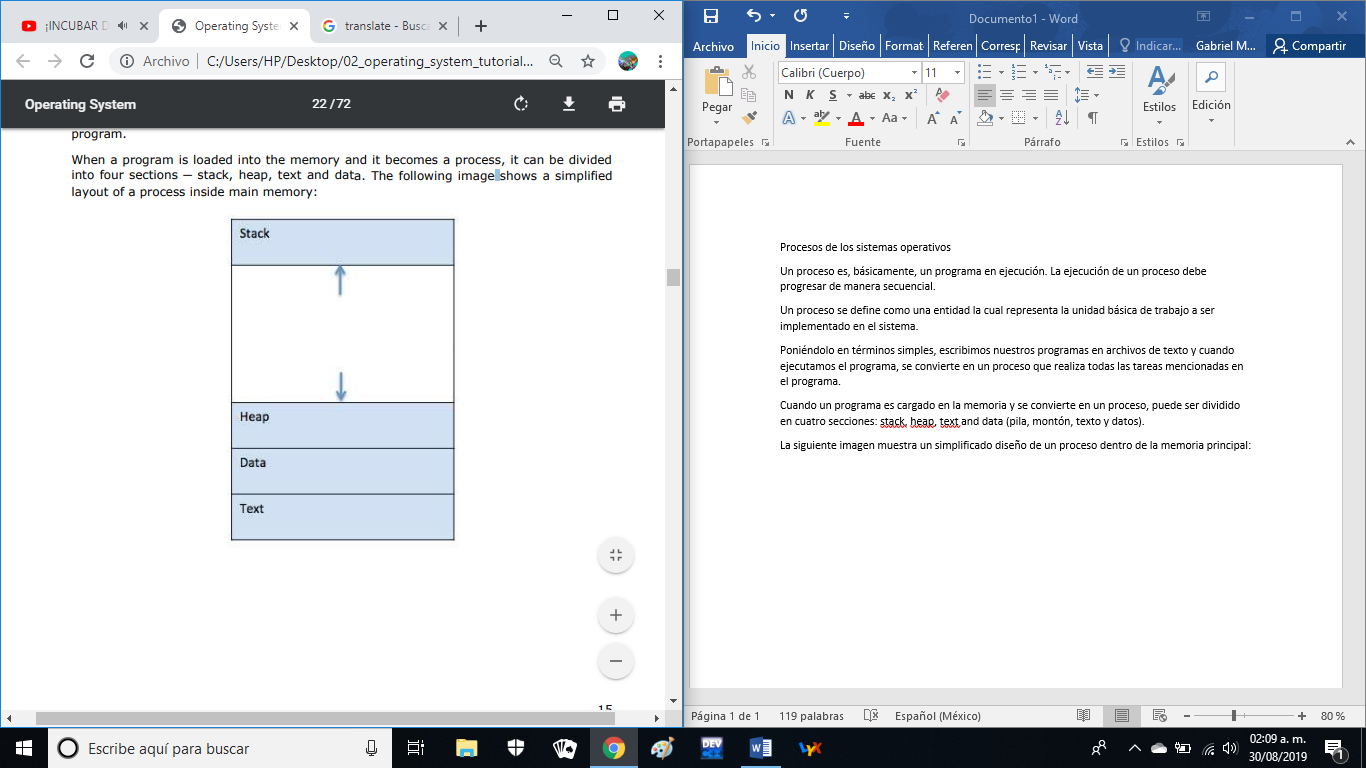
Un proceso es, básicamente, un programa en ejecución. La ejecución de un proceso debe progresar de manera secuencial.

Un proceso se define como una entidad la cual representa la unidad básica de trabajo a ser implementado en el sistema.

Poniéndolo en términos simples, escribimos nuestros programas en archivos de texto y cuando ejecutamos el programa, se convierte en un proceso que realiza todas las tareas mencionadas en el programa.

Cuando un programa es cargado en la memoria y se convierte en un proceso, puede ser dividido en cuatro secciones: stack, heap, text and data (pila, montón, texto y datos).

La siguiente imagen muestra un simplificado diseño de un proceso dentro de la memoria principal:



|  |  |
| --- | --- |
| S.N. | Componente y descripción |
| 1 | **Stack (pila):**  El proceso Stack contiene la información temporal tal como parámetros de método/función. |
| 2 | **Heap (montón):**  Esta es una memoria asignada dinámicamente a un proceso durante su tiempo de ejecución. |
| 3 | **Text (texto):**  Este incluye la actividad actual representada por el valor del Contador del Programa (Program Counter) y el contenido de los registros del procesador. |
| 4 | **Data (datos)**  Esta sección contiene las variables globales y estáticas. |

**Programa**

Un programa es una pieza de código que puede ser una sola línea o millones de líneas. Un programa computacional es usualmente escrito por un programador en un lenguaje de programación. Por ejemplo, aquí un simple programa escrito en el lenguaje de programación C:

#include <stdio.h>

int main() {

printf("Hello, World! \n");

return 0;

}

Un programa es una colección de instrucciones que realiza una específica tarea cuando es ejecutado por un computador.

Una parte de un programa que realiza una tarea bien definida se conoce como algoritmo. Una colección de programas, librerías y datos relacionados es conocida como software.

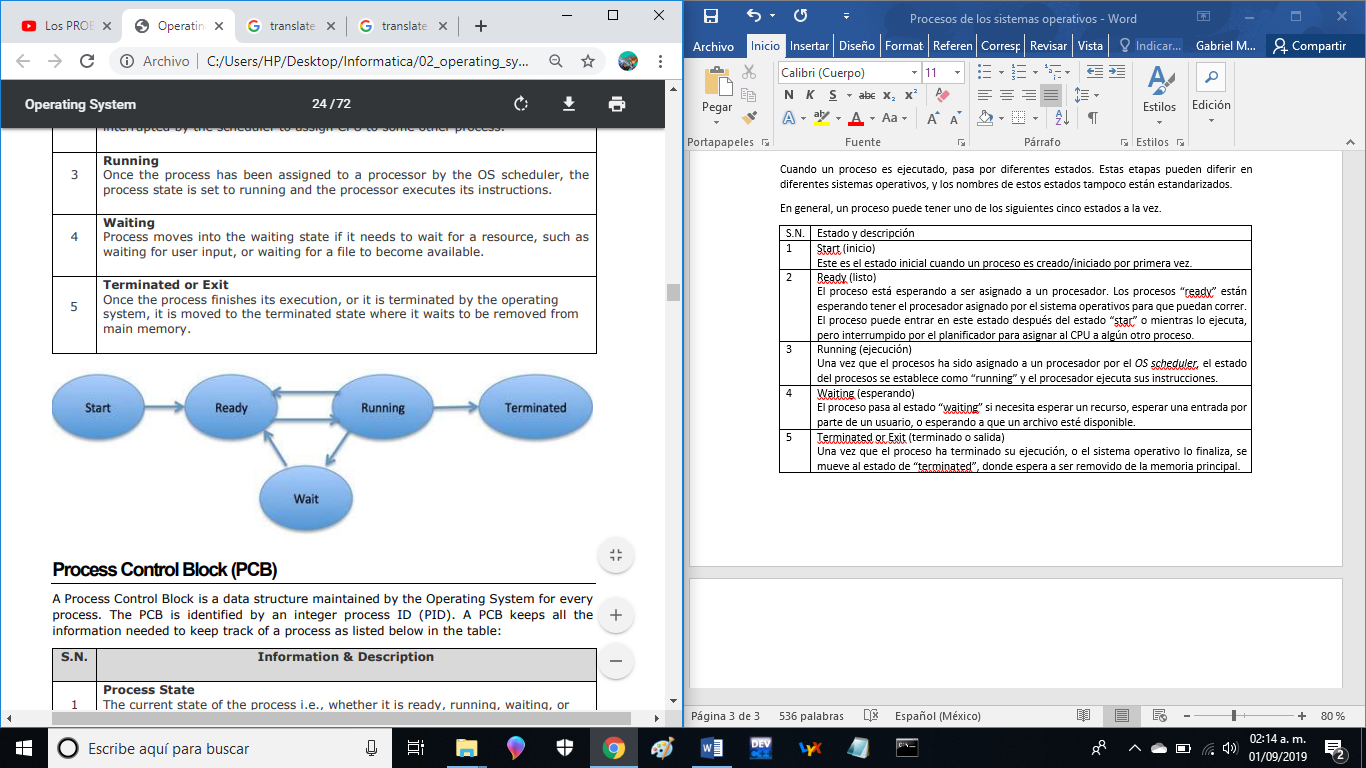
Cuando comparamos un programa con un proceso, podemos concluir que un proceso es una instancia dinámica de un programa.

**Ciclo de Vida del Proceso**

Cuando un proceso es ejecutado, pasa por diferentes estados. Estas etapas pueden diferir en diferentes sistemas operativos, y los nombres de estos estados tampoco están estandarizados.

En general, un proceso puede tener uno de los siguientes cinco estados a la vez.

|  |  |
| --- | --- |
| S.N. | Estado y descripción |
| 1 | **Start (inicio)**  Este es el estado inicial cuando un proceso es creado/iniciado por primera vez. |
| 2 | **Ready (listo)**  El proceso está esperando a ser asignado a un procesador. Los procesos “ready” están esperando tener el procesador asignado por el sistema operativos para que puedan correr.  El proceso puede entrar en este estado después del estado “star” o mientras lo ejecuta, pero interrumpido por el planificador para asignar al CPU a algún otro proceso. |
| 3 | **Running (ejecución)**  Una vez que el procesos ha sido asignado a un procesador por el *OS scheduler (planificador) (El****planificador****(en*[*inglés*](https://es.wikipedia.org/wiki/Idioma_ingl%C3%A9s)*scheduler) es un componente funcional muy importante de los*[*sistemas operativos*](https://es.wikipedia.org/wiki/Sistema_operativo)[*multitarea*](https://es.wikipedia.org/wiki/Multitarea)*y*[*multiproceso*](https://es.wikipedia.org/wiki/Multiproceso)*, y es esencial en los sistemas operativos de*[*tiempo real*](https://es.wikipedia.org/wiki/Tiempo_real)*. Su función consiste en repartir el tiempo disponible de un*[*microprocesador*](https://es.wikipedia.org/wiki/Microprocesador)*entre todos los*[*procesos*](https://es.wikipedia.org/wiki/Proceso_(inform%C3%A1tica))*que están disponibles para su ejecución),* el estado del procesos se establece como “running” y el procesador ejecuta sus instrucciones. |
| 4 | **Waiting (esperando)**  El proceso pasa al estado “waiting” si necesita esperar un recurso, esperar una entrada por parte de un usuario, o esperando a que un archivo esté disponible. |
| 5 | **Terminated or Exit (terminado o salida)**  Una vez que el proceso ha terminado su ejecución, o el sistema operativo lo finaliza, se mueve al estado de “terminated”, donde espera a ser removido de la memoria principal. |

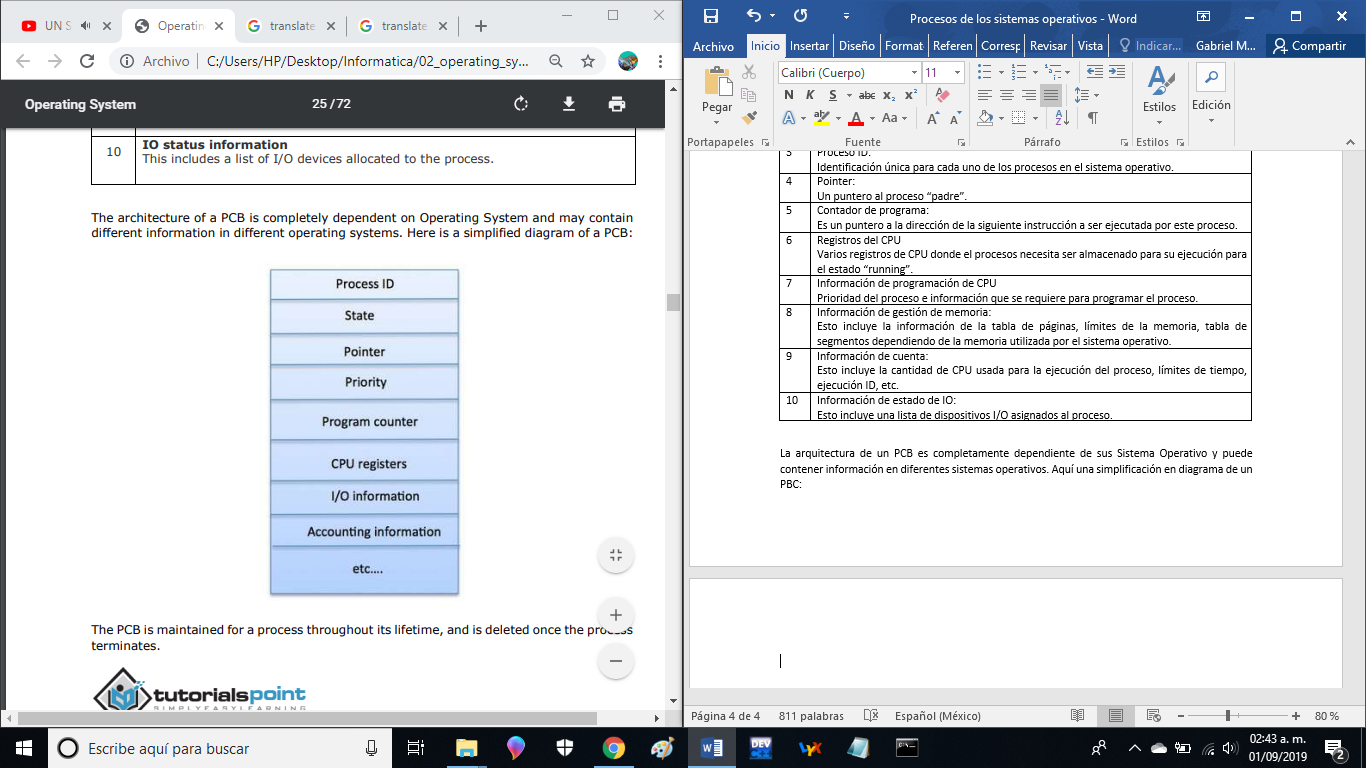


**Process Control Block (PCB)**

Un Bloque de Control de Proceso es una estructura de datos mantenida por el sistema operativo para cada proceso. El PCB se identifica mediante un proceso entero ID (PID) (En computación, **PID** es una abreviatura de process ID, o sea, ID del **proceso** o bien identificador de **procesos**. El identificador de **procesos** es un número entero usado por el kernel de algunos sistemas operativos (como el de Unix o el de Windows NT) para identificar un **proceso** de forma unívoca.). Un PCB guarda toda la información necesaria para realizar un seguimiento de un proceso como se detalla en la siguiente tabla:

|  |  |
| --- | --- |
| S.N. | Información y Descripción |
| 1 | **Estado del proceso**  El estado en cuestión del proceso, es decir, si está “ready”, “running”, “waiting” o el que sea. |
| 2 | **Privilegios del proceso:**  Estos son requeridos para permitir o no el acceso a los recursos del sistema. |
| 3 | **Proceso ID:**  Identificación única para cada uno de los procesos en el sistema operativo. |
| 4 | **Pointer:**  Un puntero al proceso “padre”. |
| 5 | **Contador de programa:**  Es un puntero a la dirección de la siguiente instrucción a ser ejecutada por este proceso. |
| 6 | **Registros del CPU**  Varios registros de CPU donde el procesos necesita ser almacenado para su ejecución para el estado “running”. |
| 7 | **Información de programación de CPU**  Prioridad del proceso e información que se requiere para programar el proceso. |
| 8 | **Información de gestión de memoria:**  Esto incluye la información de la tabla de páginas, límites de la memoria, tabla de segmentos dependiendo de la memoria utilizada por el sistema operativo. |
| 9 | **Información de cuenta:**  Esto incluye la cantidad de CPU usada para la ejecución del proceso, límites de tiempo, ejecución ID, etc. |
| 10 | **Información de estado de IO:**  Esto incluye una lista de dispositivos I/O asignados al proceso. |

La arquitectura de un PCB es completamente dependiente de sus Sistema Operativo y puede contener diferente información en diferentes sistemas operativos. Aquí una simplificación en diagrama de un PBC:



El PCB se mantiene para un proceso durante toda su vida útil y se elimina una vez que proceso termina UnU.