

CURSO PARA LA OBTENCIÓN DEL

DIPLOMA DE INFORMÁTICA MILITAR

(59130)

Teoría de Sistemas Operativos

PORFOLIO

Alumno: Xavier Guerrero Fernández

Fecha de cierre del porfolio: 31 de octubre de 2023

Un porfolio es una colección de documentos del trabajo del estudiante que exhibe su esfuerzo, progreso y logros. Es una forma de recopilar la información que demuestra las habilidades y logros de los estudiantes. Puede servir como forma de evaluación y de autoevaluación.

[1 Capítulo 1 3](#_Toc148024836)

[1.1 Preguntas 3](#_Toc148024837)

[1.1.1 Explicar el propósito principal de un sistema operativo 3](#_Toc148024838)

[1.1.2 Características de la vista de usuario del SO. ¿Qué herramientas utiliza el usuario en esta vista para enviar órdenes al sistema operativo? Algunas computadoras tienen poca o ninguna vista de usuario. Pon ejemplos 3](#_Toc148024839)

[1.1.3 Al parecer no hay una estricta separación entre los programas kernel y los programas de aplicación. Exponga cuatro tipos de programas que podíamos incluir en el SO 3](#_Toc148024840)

[1.1.4 Comentar las peculiaridades de los SO en general en cuanto a su dimensionamiento en líneas de código, su prolongado ciclo de vida y su complejidad 3](#_Toc148024841)

[1.1.5 Modelo conceptual del sistema operativo en una perspectiva de arriba hacia abajo. Ídem de abajo hacia arriba. 4](#_Toc148024842)

[1.1.6 Aunque hay muchos profesionales de las ciencias de la computación, solo un pequeño porcentaje de ellos estará involucrado en la creación o modificación de un sistema operativo. Indique cuatro razones para estudiar la teoría de los sistemas operativos en este curso. 4](#_Toc148024843)

[1.1.7 En 1968 Ken Thompson con una pequeña minicomputadora PDP-7 abandonada se propuso escribir una versión simplificada de MULTICS para un solo usuario. ¿Por qué tenía la capacidad de desarrollar un SO? ¿Qué importancia tuvo este hecho en la historia de los SO? 4](#_Toc148024844)

[1.1.8 Enunciado falso: “1960 Doug Engelbart en el Instituto de Investigación de Stanford inventó la interfaz gráfica de usuario, completa con ventanas, íconos, menús y ratón, encargado por Apple. Pero Steve Jobs, quien coinventó la computadora de Xerox PARC compró la interface gráfica de Engelbart para sus máquinas. En 1999, Apple adoptó un núcleo derivado de BSD UNIX que reemplazó al micronúcleo Mach de la Universidad Carnegie Mellon. Por eso el macOS de Apple no es un sistema operativo basado en UNIX”. 5](#_Toc148024845)

[Rectifique adecuadamente ese falso enunciado 5](#_Toc148024846)

[1.1.9 Explicar las diferencias claras entre driver de dispositivo y controlador de dispositivo. 5](#_Toc148024847)

[1.1.10 Mediante el esquema que se acompaña, explicar el concepto IRQ en el SO 5](#_Toc148024848)

[1.1.11 Diferencia entre trampa (TRAP) e interrupción utilizando la siguiente tabla: 6](#_Toc148024849)

[1.1.12 Respecto a la estructura de almacenamiento en niveles, sustituya las manchas rojas por los conceptos: rápido, lento, pequeño, grande, memoria volátil, memoria no volátil, almacenamiento primario, secundario y terciario 6](#_Toc148024850)

[1.1.13 Qué estructura de datos es la de la figura y ponga ejemplos donde el SO la emplea 7](#_Toc148024851)

[1.1.14 Qué estructura de datos es la de la figura y ponga ejemplos donde el SO la emplea 7](#_Toc148024852)

[1.1.15 Un árbol rojo-negro es un árbol binario de búsqueda en el que cada nodo tiene un atributo de color cuyo valor es rojo o negro. Además de los requisitos impuestos a los árboles binarios de búsqueda convencionales, enuncie cuatro reglas para tener un árbol rojo-negro válido. Linux utiliza un árbol rojo-negro (rbtree) en muchos lugares del kernel, indique tres: 7](#_Toc148024853)

# Capítulo 1. Introducción al Sistema Operativo

## Preguntas

### Explicar el propósito principal de un sistema operativo

Los sistemas operativos existen para dos propósitos principales. Una es que está diseñado para asegurarse de que un sistema informático funcione bien mediante la gestión de sus actividades informáticas. Otra es que proporciona un entorno para el desarrollo y ejecución de programas.

El propósito principal de un sistema operativo es permitir al usuario entenderse con el Hardware, llevando a cabo la administración de manera eficiente del mismo y proporcionando al usuario herramientas para los programas de aplicación.

### Características de la vista de usuario del SO. ¿Qué herramientas utiliza el usuario en esta vista para enviar órdenes al sistema operativo? Algunas computadoras tienen poca o ninguna vista de usuario. Pon ejemplos

Características de la vista de usuario del SO:

* Desde una vista usuario se dispone del modo usuario para manejar un subconjunto de instrucciones del SO.
* La vista del usuario de la computadora varía según la interfaz que se utilice.
* Actúan de forma ergonómica, directa y sencilla para sacar el mejor partido de los recursos.
* Es el programa de interfaz de usuario, shell o GUI, entendido como el nivel más bajo de software en modo de usuario que permite al usuario iniciar otros programas.

Herramientas para enviar ordenas al SO: navegador web, un lector de correo electrónico o un reproductor de música. Estos programas también hacen un uso intensivo del sistema operativo.

Computadores con poco o ninguna vista de usuario: Microcontroladores empotrados en domótica y automóvil.

### Al parecer no hay una estricta separación entre los programas kernel y los programas de aplicación. Exponga cuatro tipos de programas que podíamos incluir en el SO

Programas de ayuda, programas de administración, programas de entorno gráfico de gestión, y programas intérpretes de comandos.

### Comentar las peculiaridades de los SO en general en cuanto a su dimensionamiento en líneas de código, su prolongado ciclo de vida y su complejidad

Un SO implica un desarrollo de gran complejidad. Esto supone numerosas líneas de código, así como numerosos recursos materiales y humanos, que también se traducen en términos económicos. Por ello, su elaboración desde cero supone un despliegue de tiempo, presupuesto, medios y personas que no muchas empresas están dispuestas a asumir. Es por ello que un SO se mantiene modificando e implementando funcionalidades a medida que aparecen nuevas tecnologías para poder utilizar periféricos y protocolos de comunicación entre otros. Por ello, se concluye que, aunque estos sistemas operativos tengan en un momento dado un cambio en la arquitectura como el caso de Windows NT, el kernel a penas cambia con el paso del tiempo en muchos de los SSOO que conocemos a día de hoy.

### Modelo conceptual del sistema operativo en una perspectiva de arriba hacia abajo. Ídem de abajo hacia arriba.

El concepto de que un sistema operativo proporciona principalmente abstracciones a los programas de aplicación es una **visión de arriba hacia abajo**.

Una visión alternativa, **de abajo hacia arriba**, sostiene que el sistema operativo está ahí para administrar todas las piezas de un sistema complejo.

### Aunque hay muchos profesionales de las ciencias de la computación, solo un pequeño porcentaje de ellos estará involucrado en la creación o modificación de un sistema operativo. Indique cuatro razones para estudiar la teoría de los sistemas operativos en este curso.

Obtener el máximo rendimiento de los recursos que ofrece el hardware y software, eludiendo la configuración en serie y adaptándolos a las necesidades de cada usuario.

Fortalecer la seguridad del propio SO, dificultando accesos no autorizados, ya que el SO instalado de serie ofrece vulnerabilidades.

Es una herramienta fundamental en el fortalecimiento de la seguridad de los servidores.

Permite crear un SO propio, partiendo de distribuciones existentes, aumentando las funcionalidades de protección frente a ataques.

### En 1968 Ken Thompson con una pequeña minicomputadora PDP-7 abandonada se propuso escribir una versión simplificada de MULTICS para un solo usuario. ¿Por qué tenía la capacidad de desarrollar un SO? ¿Qué importancia tuvo este hecho en la historia de los SO?

Ken Thomson trabajaba en los laboratorios Bell, desarrollando el SO MULTICS. Se considera el padre del lenguaje B en el que estaba escrito dicho SO y continuó desarrollando este sistema hasta que se abandonó debido a su creciente complejidad. En 1969, por su cuenta y junto a Dennis Ritchie trató de convertir el SO MULTICS abandonado en un sistema monousuario más simple y que tuviera mucho más rendimiento.

Este sistema se acabó transformado formalmente en UNIX en 1970 cuando consiguieron el apoyo económico de los laboratorios Bell. En 1972 decidieron reescribir completamente el código usando lenguaje C haciendo que se pudiera modificar fácilmente para trabajar en múltiples computadoras y convirtiéndose en el padre de una gran familia de sistemas operativos multisistema y multiusuario como System V, XENIX, BSD y LINUX, entre otros.

### Enunciado falso: “1960 Doug Engelbart en el Instituto de Investigación de Stanford inventó la interfaz gráfica de usuario, completa con ventanas, íconos, menús y ratón, encargado por Apple. Pero Steve Jobs, quien coinventó la computadora de Xerox PARC compró la interface gráfica de Engelbart para sus máquinas. En 1999, Apple adoptó un núcleo derivado de BSD UNIX que reemplazó al micronúcleo Mach de la Universidad Carnegie Mellon. Por eso el macOS de Apple no es un sistema operativo basado en UNIX”.

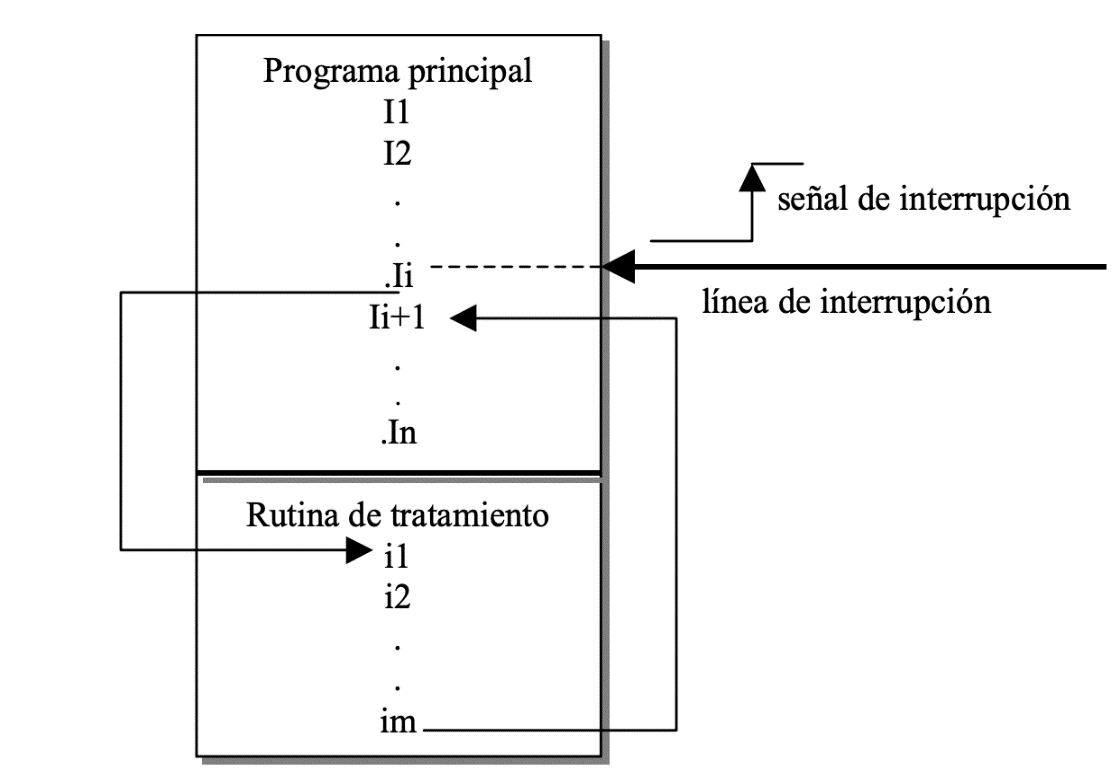
## Rectifique adecuadamente ese falso enunciado

“1960 Doug Engelbart en el Instituto de Investigación de Stanford inventó la interfaz gráfica de usuario, completa con ventanas, íconos, menús y ratón, encargado por Apple (NO FUE ENCARGADA POR APPLE, FUE POR XEROX PARCK). Pero Steve Jobs, quien coinventó la computadora de Xerox PARC (no coinventó la computadora de XEROC PARC) compró (NO LO COMPRÓ, CONSTRUYÓ UNO PROPIO) la interface gráfica de Engelbart para sus máquinas. En 1999, Apple adoptó un núcleo derivado de BSD UNIX que reemplazó al micronúcleo Mach de la Universidad Carnegie Mellon. Por eso el macOS de Apple no es (SI QUE ES) un sistema operativo basado en UNIX”.

### Explicar las diferencias claras entre driver de dispositivo y controlador de dispositivo.

Un controlador de dispositivo se encuentra en el propio dispositivo, es una parte de él, y tiene una parte de hardware y de programa (software). El driver de dispositivo, por el contrario, es un programa que sirve para comunicar ese dispositivo con el Sistema Operativo y que se pueda ejecutar.

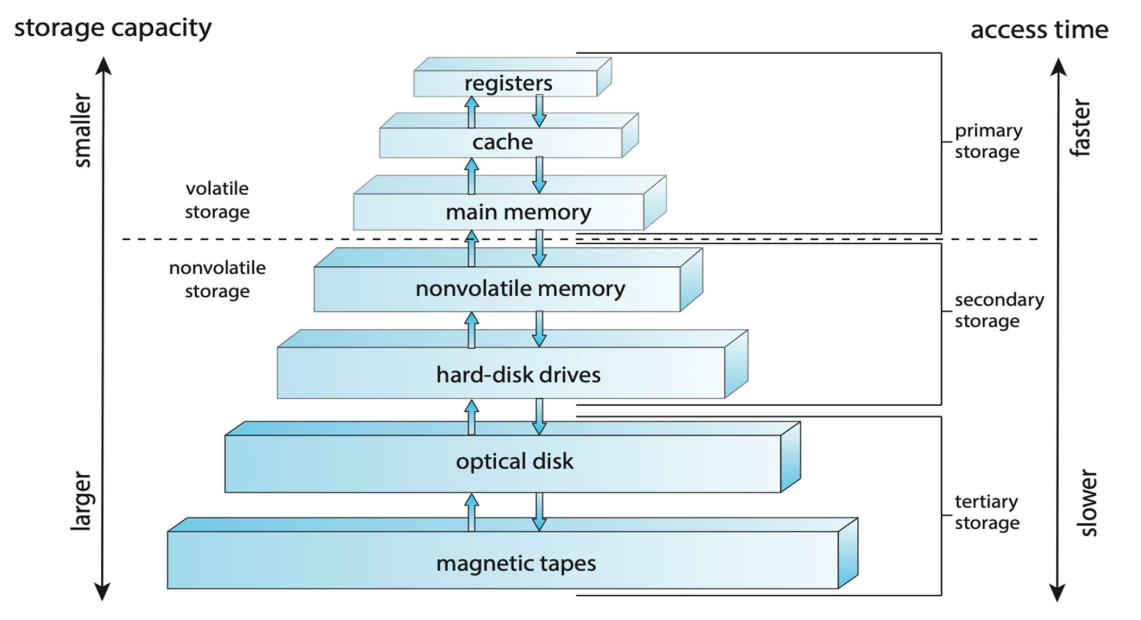
### Mediante el esquema que se acompaña, explicar el concepto IRQ en el SO

Una interrupción es una suspensión temporal de la ejecución de un proceso, para pasar a ejecutar una subrutina de servicio de interrupción, que, por lo general, no forma parte del programa, sino que pertenece al sistema operativo. Una vez finalizada dicha subrutina, se reanuda la ejecución del programa. Las interrupciones son generadas por los dispositivos periféricos habilitando una señal del CPU (llamada IRQ "interrupt request") para solicitar atención del mismo. Las interrupciones son una parte importante de la arquitectura de una computadora.

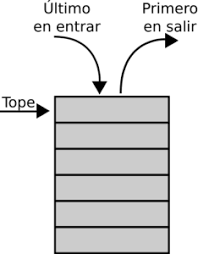
### Diferencia entre trampa (TRAP) e interrupción utilizando la siguiente tabla:

|  |  |  |
| --- | --- | --- |
|  | Trampa | Interrupción |
| Método de Generación | surge como evento durante la ejecución de una instrucción en el programa | generada como un evento por dispositivos de hardware |
| Funcionalidad principal | invoca la funcionalidad del SO quien transfiere su control al manipulador de trampas | activa el procesador, quien transfiere el evento al controlador de interrupciones |
| Ocurrencia | síncrona, tras la ejecución de una instrucción | asíncrona, no depende de la ejecución de instrucciones, al producirse por hardware no está directamente relacionada con una secuencia de instrucciones |
| Sinónimos | interrupción de software | interrupción de hardware |

### Respecto a la estructura de almacenamiento en niveles, sustituya las manchas rojas por los conceptos: rápido, lento, pequeño, grande, memoria volátil, memoria no volátil, almacenamiento primario, secundario y terciario

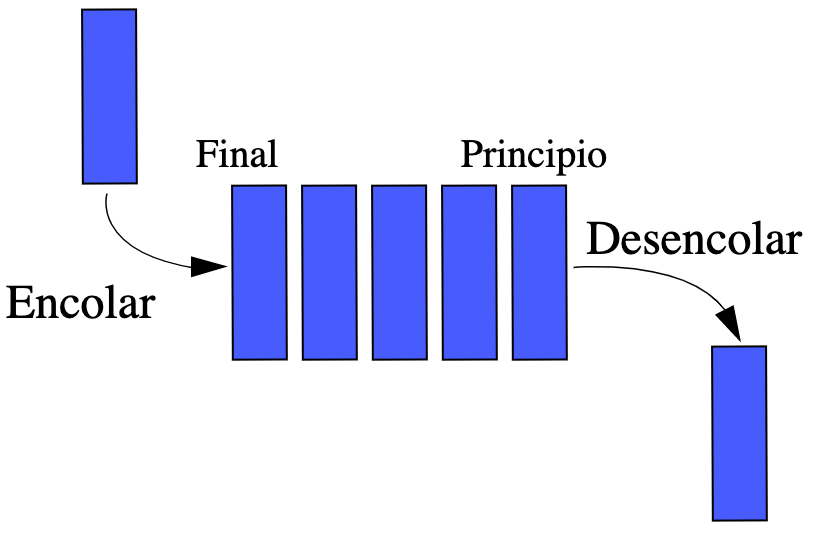


### Qué estructura de datos es la de la figura y ponga ejemplos donde el SO la emplea



Esta estructura de datos se trata de una pila ordenada secuencialmente y que utiliza el principio de último en entrar, primero en salir (LIFO). Un sistema operativo a menudo usa una pila cuando invoca llamadas a funciones. Los parámetros, las variables locales y la dirección de retorno se colocan en la pila cuando se llama a una función; regresar de la llamada a la función saca esos elementos de la pila.

### Qué estructura de datos es la de la figura y ponga ejemplos donde el SO la emplea



Es una estructura de datos ordenada en cola que utiliza el principio de primero en entrar, primero en salir (FIFO): los elementos se eliminan de una cola en el orden en que se insertaron.

### Un árbol rojo-negro es un árbol binario de búsqueda en el que cada nodo tiene un atributo de color cuyo valor es rojo o negro. Además de los requisitos impuestos a los árboles binarios de búsqueda convencionales, enuncie cuatro reglas para tener un árbol rojo-negro válido. Linux utiliza un árbol rojo-negro (rbtree) en muchos lugares del kernel, indique tres:

Además de los requisitos impuestos a los árboles binarios de búsqueda convencionales, se deben satisfacer las siguientes reglas para tener un árbol rojo-negro válido:

* Todo nodo es o bien rojo o bien negro.
* La raíz es negra.
* Todas las hojas (NULL) son negras.
* Todo nodo rojo debe tener dos nodos hijos negros.

Linux utiliza un árbol rojo-negro (rbtree) en muchos lugares del kernel, por ejemplo:

* El programador de E/S utilizan rbtree para realizar un seguimiento de las solicitudes.
* En el control de las unidades de datos en bloque de CD / DVD.
* El código del temporizador de alta resolución usa rbtree para organizar las solicitudes del temporizador.

# Capítulo 2: Llamadas y tipos de sistemas operativos.

## Preguntas

### ¿Cuál es el propósito del intérprete de comandos? ¿Por qué suele estar separado del núcleo?

La función principal del intérprete de comandos es obtener y ejecutar el comando especificado por el usuario.Suele estar separado del núcleo porque la mayoría de los sistemas operativos, incluidos Linux, UNIX y Windows, tratan al intérprete de comandos como un programa especial que se ejecuta cuando se inicia un proceso o cuando un usuario inicia sesión por primera vez (en sistemas interactivos), existiendo sistemas operativos con múltiples intérpretes de comandos para elegir, que se conocen como shells.

### Ventajas de CLI

* Tiene un rendimiento más rápido: Si se sabe cómo usar CLI y se está familiarizado con diferentes comandos, el ciclo pregunta-respuesta se completará más rápido en comparación con el uso de GUI.
* Consume menos memoria: CLI usa menos memoria en comparación con la GUI.
* Se necesita un monitor de baja resolución para su uso a diferencia del GUI.
* Un procesador lento puede funcionar: ya que no necesitan potencia de procesamiento adicional.
* No necesita instalar complejos programas para gestionar entornos gráficos. Todos los SO importantes admiten CLI.
* Autocompletar: Cuando escribe comandos en la CLI, la mayoría de los comandos se completan automáticamente al hacer clic en el botón TAB en el teclado.
* Funciones avanzadas: En CLI se tiene posibilidades de acceso a más funciones avanzadas que en GUI.
* Trabajo remoto: CLI es una opción ágil para comunicación entre dispositivos y operar diferentes comandos CLI en computadoras remotas.
* Historial de comandos: Al ejecutar comandos en CLI, las órdenes se guardan en un recurso de la memoria. Al usar las flechas arriba/abajo del teclado se puede volver a usar esos comandos.

### Ventajas de GUI

Una interfaz gráfica de usuario o GUI es fácil de usar. En lugar de ingresar textos que expresan comandos directamente a través de una CLI, los usuarios emplean un sistema de ventanas y menús basado en ratón caracterizado por una metáfora de escritorio. El usuario mueve el dispositivo ratón para colocar su puntero sobre imágenes o iconos en la pantalla (el escritorio) que representan programas, archivos, directorios y funciones del sistema gráfico. Dependiendo de la ubicación del puntero, hacer clic en un botón del ratón puede invocar un programa, seleccionar un archivo o directorio, conocido como carpeta, o desplegar un menú que contiene comandos.

### ¿Cuál es el propósito de las llamadas al sistema?

Una llamada al sistema es una rutina que permite a una aplicación de usuario solicitar acciones que requieren privilegios especiales. La adición de llamadas al sistema es una de varias maneras de ampliar las funciones proporcionadas por el kernel.

### Diferencias entre la llamada a una función en el dominio de aplicaciones de usuario y una llamada al sistema.

* Una llamada al sistema difiere de una función de usuario de varias formas:
* Una llamada al sistema tiene más privilegios que una subrutina normal.
* Una llamada al sistema se ejecuta con el privilegio modo de kernel en el dominio de protección de kernel.
* El código de llamada del sistema y los datos se encuentran en la memoria global del kernel.
* Las rutinas de llamada al sistema pueden crear y utilizar procesos de kernel para realizar el proceso asíncrono.
* Las llamadas del sistema no pueden utilizar bibliotecas compartidas ni ningún símbolo que no se encuentre en el dominio de protección del kernel.

### Indique los seis tipos de llamadas al sistema que generalmente tienen los SO

En general las llamadas al sistema se pueden agrupar aproximadamente en seis categorías principales: control de procesos, gestión de archivos, gestión de dispositivos, mantenimiento de la información, comunicaciones y protección.

### ¿Qué es POSIX?

POSIX ([acrónimo](https://es.wikipedia.org/wiki/Acrónimo) de Portable Operating System Interface, y X viene de [UNIX](https://es.wikipedia.org/wiki/UNIX) como seña de identidad de la [API](https://es.wikipedia.org/wiki/Application_Programming_Interface)) es una norma escrita por la [IEEE](https://es.wikipedia.org/wiki/IEEE), que define una interfaz estándar del [sistema operativo](https://es.wikipedia.org/wiki/Sistema_operativo) y el entorno, incluyendo un [intérprete de comandos](https://es.wikipedia.org/wiki/Intérprete_de_comandos) (o "shell").

El término fue sugerido por [Richard Stallman](https://es.wikipedia.org/wiki/Richard_Stallman) en la década de 1980, en respuesta a la demanda del [IEEE](https://es.wikipedia.org/wiki/IEEE), que buscaba un nombre fácil de recordar. La traducción del acrónimo es "Interfaz de Sistema Operativo Portable".

### ¿Cuáles son los tipos de SO según estructura de kernel? De una explicación de una línea de cada uno de ellos

* **Sistemas operativos de kernel monolítico/monolítico con extensiones cargables**: Los sistemas operativos monolíticos se caracterizan por implementar en el núcleo los cuatro componentes fundamentales del sistema operativo.
* **Sistemas operativos de microkernel:** Se caracterizan por disponer de un núcleo que implementa únicamente Planificación de proceso, Mecanismo de comunicación entre procesos y la Gestión de interrupciones. Además, existen procesos servidores que, cuando un proceso cualquiera solicita un servicio a través de una llamada al sistema, el micronúcleo canaliza la petición al proceso servidor correspondiente.
* **Sistemas operativos de kernel híbrido**: El kernel de este tipo de Sistema Operativo el núcleo en cuestión usa conceptos de arquitectura tanto del diseño monolítico como del micronúcleo.

### Comparación de un kernel monolítico de un microkernel

Los sistemas operativos monolíticos se caracterizan por implementar en el núcleo los cuatro componentes fundamentales del sistema operativo, que son la planificación de procesos, la administración de la memoria principal, la administración de ficheros y la gestión de los dispositivos de entrada/salida. Algunos de ellos admiten la inclusión de módulos compilados en tiempo de ejecución del kernel del sistema operativo sin necesidad de recompilar completamente el mismo.

Por otra parte, los microkernel se caracterizan por disponer de un núcleo que implementa únicamente las funciones básicas de Planificación de proceso, Mecanismo de comunicación entre procesos y la Gestión de interrupciones.

### ¿Qué tipo de núcleo llevan los siguientes SO?

|  |  |
| --- | --- |
| Ubuntu | Monolítico con extensiones. |
| Windows 10 | Híbrido. |
| MacOS | Monolítico con extensiones. |
| MacOSX | Híbrido. |
| Android | Monolítico con extensiones. |
| IOS | Híbrido. |

### Utilizando la tabla a continuación, establecer las diferencias entre el kernel de un S.O. monolítico con módulos recargable y un S.O. híbrido

|  |  |  |
| --- | --- | --- |
|  | Monolítico con Módulos Recargables | Híbrido |
| Arquitectura |  |  |
| Espacio del S.O. en modo usuario |  |  |
| Espacio del S.O. en modo kernel |  |  |
| Modularidad |  |  |
| Modo de operación de los módulos |  |  |
| Riesgo |  |  |
| Coste de diseño |  |  |
| Coste o facilidad de mantenimiento |  |  |
| Consumo de recursos computacionales |  |  |
| Ejemplos |  |  |

# Capítulo 3. Procesos

## Preguntas

### ¿Qué es un proceso del sistema operativo? ¿Qué recursos del sistema necesita un proceso para realizar satisfactoriamente su tarea?

El proceso es un programa en ejecución. El estado de la actividad actual de un proceso está representado por el valor del contador del programa y el contenido de los registros del procesador. Por último, los procesos necesitan recursos para ejecutarse como son:

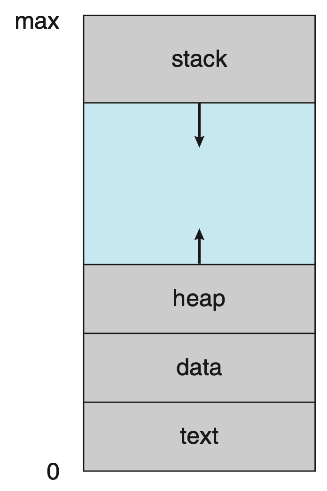
* Tiempo de CPU.
* Memoria.
* Archivos.
* Dispositivos E/S

### ¿Cuales son las obligaciones del S.O. como gestor de procesos?

Las obligaciones del SO como gestor de procesos son:

* Creación y eliminación de procesos
* Planificación de procesos
* Establecimiento de mecanismos para la sincronización y comunicación de procesos.
* Manejo de bloqueos mutuos

### ¿Qué suele almacenarse en cada una de las secciones de la memoria ocupada por un proceso?

 El diseño de la memoria que un proceso necesita generalmente se divide en varias secciones:

* Sección de texto: el código ejecutable
* Sección de datos: variables globales
* Sección Heap: memoria que se asigna dinámicamente durante el tiempo de ejecución del programa
* Sección stack (de pila): almacenamiento temporal de datos al invocar funciones (como parámetros de función, direcciones de retorno y variables locales.

### Se ha ejecutado el comando “size /bin/more”. Explicar la salida obtenida

[root@localhost home]# size /bin/more

   text    data     bss     dec     hex filename

  29527    1824      24   31375    7a8f /bin/more

### Respecto a los estados de un proceso, rellene adecuadamente la tabla

| **Estado** | **Descripción** |
| --- | --- |
| Nuevo | El proceso se está creando. |
| Preparado | El proceso está esperando a ser asignado a un procesador. |
| Activo | Se están ejecutando instrucciones. |
| Bloqueado | El proceso está esperando que ocurra algún evento (como una finalización de E/S o la recepción de una señal). |
| Terminado | El proceso ha terminado la ejecución. |

### ¿Qué es un PCB? ¿Qué información generalmente se suele guardar en un PCB?

Cada proceso está representado en el sistema operativo por un bloque de control de procesos (PCB), también llamado bloque de control de tareas. Contiene muchos campos de información asociados a un proceso específico como:

* **Estado del proceso.** El estado puede ser nuevo, listo, en ejecución, en espera, detenido, etc.
* **Contador de programa** indica la dirección de la siguiente instrucción a ejecutar para este proceso
* **Registros de la CPU**. Los registros varían en número y tipo, dependiendo de la arquitectura de la computadora. Incluyen acumuladores, registros de índice, punteros de pila y registros de propósito general, además de cualquier información que condicione la ejecución del proceso. Junto con el contador del programa, esta información de estado debe guardarse cuando ocurre una interrupción, para permitir que el proceso continúe correctamente cuando se reanude la ejecución.
* **Información de planificación (scheduling) de CPU**: prioridad de proceso, punteros a colas de scheduling, etc.
* **Información de gestión de memoria:** valor de los registros base y límite, y las tablas de páginas o de segmentos, según el sistema de memoria utilizado por el sistema operativo.
* **Información de contabilidad de CPU**: la cantidad de CPU y el tiempo real utilizado, los límites de tiempo, la identificación de tareas o de proceso, etc.
* **Información de estado de E/S:** la lista de dispositivos de E/S asignados al proceso, una lista de archivos abiertos, etc.

## Ejercicios