

Teoria Parcial

Si te llega este papiro del mar muerto a tus manos y tienes dudas/consultas mira la última página concha de tu madre

Temas

Temas según lo que averigüe por parciales viejos y clases del tetoides

- **Introducción a los Sistemas Operativos:** -
 - Objetivos y Funciones de los SO
 - Evolución de los SO
 - El SO como interfaz
- **System Calls:**
 - Agrupación de las llamadas al sistema
 - Gestión de procesos
 - Gestión de memoria
 - Gestión del sistema de archivos
 - Gestión del almacenamiento masivo
 - Gestión de sistemas de E/S
- **Procesos:**
 - Concepto
 - Diagrama de 5 Estados
 - Elementos/PCB
- **Planificadores**
 - Concepto
 - Planificador de Procesos
 - Colas de Planificación
 - Planificadores
 - Algoritmos de planificación
- **Hilos**
 - Conceptos
 - Procesos e Hilos
 - Funcionalidades de los Hilos
- **Concurrencia, Exclusión mutua y sincronización**
 - Conceptos
 - Principios
 - Semaforos

Introduccion a los Sistemas Operativos

Objetivos y Funciones de un SO

Un sistema Operativo es un programa que controla la ejecucion de aplicaciones y programas que actua como interfaz entre las aplicaciones y el hardware del computador.

Sus **objetivos** son los siguientes:

- **Facilidad de uso:** Un SO facilita el uso de una computadora.
- **Eficiencia:** Permite la utilizacion eficiente de los recursos en la computadora.
- **Capacidad para evolucionar:** Se debe contruir de forma en la que este se pueda desarrollar.

Entre sus servicios los SO ofrecen el desarrollo de programas, ejecucion de programas, acceso a dispositivos E/S, acceso a ficheros, al sistema, entre otros.

El SO como interfaz (Estructura Generica de un Sistema)

Lo podemos analizar como una capa que envuelve al hardware, formado por capas como una cebolla, en el centro se encotrnraria el Hardware, luego el Nucleo, el SO, y finalmente en la capa externa el Nivel de Usuario.

La interfaz es para el usuario en general, administradores y desarrolladores. Este puede aparecer de varias maneras, como CLI, GUI, API.

El SO como Gestor de Recursos

- Que es un proceso? Una entidad dinamica
- Que es un recurso? Puede ser algo fisico o virtual, pero requiere un proceso, y estos son escasos. Por eso el SO los administra.

El SO se encarga ademas de la **gestion de recursos** llevados a cabo en un computador, pero para que el procesador pueda realizar esto, el sistema operativo debe dejar paso a la **ejecucion de otro programas**. Por lo tanto el SO deja el control para que el **procesador** pueda realizar este trabajo **util** y de nuevo retoma el control para permitir al procesador que realice la siguiente pieza de trabajo.

Una porcion del SO se encuentra en **memoria principal**. Esto incluye al **Kernel** (Nucleo principal del SO), que contiene las funciones del SO mas frecuentemente utilizadas y otras en uso.

El resto de **memoria principal** contiene programas y datos de usuario, la asignacion de este es controlada por el SO y el hardware de gestion de memoria del procesadro.

El SO decide cuando un programa en ejecucion puede utilizar un dispositivo de E/S y controla el acceso y uso de los ficheros.

La Evolution de los SO

La primera computadora era puramente mecanica, no tenia SO, una vrg en resumen

La Generacion 0:

Se manejaban con 0 y 1, lo que representaba una valvula, no tenian SO, solo podia hacer calculos numericos y la memoria e manejaba por registros.

A principios de 1950 se mejoran las rutinas por la aparicion de las **tarjetas perforadas**.

La Generacion 1 y 2: Transistores

Mas memoria, pero surgia la necesidad de cargar:

- Rutinas
- Controladores
- Drivers Impresora

Se usan los **Monitores**, un SO basico que incluia controladores y rutinas especificas.

La Generacion 3

Primer SO lanzado por IBM que trabajaba con el punto fijo como flotante.

Nacio el concepto de **Multiprocesamiento** y las **Unidades de Control**, o como lo conocemos como controladores, e cual controla el flujo de datos a traves del procesador y coordina actividades con las otras unidades, tiene su propio hardware especifico y puede ejecutar instrucciones.

Estas computadoras tienen electronica y permiten ejecutar intrucciones

Otro concepto naciente en esta generacion es la **Concurrencia**.

La Generacion 4

Aparecen mas modelos comerciales por lo que se empezo a vender de manera masiva y se generan tres grandes familias de software:

- Mainframe (IBM): Sistemas propios.
- Servidores (HP, Lenovo, Dell): Unix/Linux.
- WorkStation (Varias): Windows/Linux.

Experiencia de resumir esto? malisima, solido 2/10, no lo recomiendo si encima lo resumen en un banio para discapacitados

Estructuras de los SO

Operaciones del SO

Estos se **controlan por medio de interrupciones**, si no hay ningun proceso a ejecutar, ningun dispositivo E/S al que dar servicio y ningun usuario al que responder, un SO debe permanecer **inactivo**, esperando a que ocurra.

Los sucesos casi siempre se indican mediante la ocurrencia de una interrupcion o una excepcion. Una **excepcion** es una **interrupcion generada por un software**, debido a un error (division por cero o crear un

acceso a memoria no valido) o a una solicitud especificada de un programa de usuario de que se realice un servicio al sistema operativo determinan que accion hay que llevar a cabo. Una rutina de servicio a la interrupcion sera proveida para tratar a la misma.

Operacion en Modo Dual

Diferenciamos dos modos de ejecucion, el **modo usuario** y el **modo kernel**, un bit denominado **bit de modo** se ania al hardware de la computadora para indicar el modo actual, en caso de elegir el modo kernel sera un 0 en caso de modo de usuario sera un 1.

-Mucha data- ``Cuando empezamos el SO basicamente se empieza en modo kernel, se carga el SO y los programas del usuario en modo usuario, en caso de que hay una interrupcion que carajos pasa? el hardware conmuta del modo usuario al modo kernel. El sistema siempre cambia a modo usuario antes de pasar el control del programa e usuario.

System Calls

Las llamadas al sistema permiten que un programa de usuario solicite al SO realizar **tareas especificas** reservadas para el mismo, se **invocan de varia formas**, pero generalmente como **excepciones**, transfiriendo el control del programa de usuario al SO, osea pasamos del **modo usuario al kernel** (ez). Lo que hara el kernel es verificar y ejecutar la solicitud, luego devuelve el control al programa.

- Como actuan estas llamadas?

Actuan como **interfaz entre el usuario y el SO** y estan disponibles como rutinas en lenguajes como C, aunque requiera un ensamblador proque son de bajo level, cada SO tiene diferentes nombres para las llamadas.

- **Una API watafak**

Application Programming Interface para las gordas, permite a diferentes **aplicaciones comunicarse y compartir funcionalidades**. Las funciones de una API **generalmente invocan llamadas al sistema**, **facilitando la portabilidad del software** entre diferentes sistemas que soportan la misma API.

El **sistema de soporte** en tiempo de ejecucion de la mayoria de los lenguajes de programacion proporcionan una interfaz que actua como **enlace entre las funciones del API y las llamadas al sistema** directamente, **mejorando la portabilidad y la facilidad de desarrollo**.

Agrupacion de las llamadas al sistema

- **Control de procesos**: Un programa en ejecucion necesita poder interrumpir dicha ejecucion de manera normal (end) o de manera anormal (**abort**)
- **Manipulacion de archivos**: Crear, eliminar, modificar, leer archivos, etc.
- **Manipulacion de los dispositivos E/S**: Un proceso puede necesitar un recurso o varios recursos. Si estos estan disponibles entonces pueden ser concedidos de lo contrario el proceso debe esperar a que haya suficientes recursos disponibles. (**request, release, open, close, etc**)
- **Mantenimiento de la Informacion**: Llamadas para transferencia de informacion entre programa del usuario y el SO (**date, time, uname, free, etc.**)

- **Comunicación** (entre procesos): Pasaje de mensajes (Utilización del kernel) o memoria compartida (no hay utilización del kernel).

Gestión de procesos

Proceso: entidad dinámica que se genera cuando se ejecuta un programa; programa en ejecución; unidad de trabajo en un sistema.

Un proceso necesita para llevar a cabo su tarea ciertos recursos, entre los que se incluyen tiempo de CPU, memoria, archivos y dispositivos de E/S. Estos se asignan al proceso en el momento de crearlo o mientras se está ejecutando. Además de los diversos recursos físicos y lógicos, pueden pasársele diversos datos de inicialización.

Cada sistema consta de una colección de procesos, siendo algunos procesos del sistema operativo y el resto procesos de usuario. Todos estos procesos pueden, potencialmente, ejecutarse en forma concurrente.

El sistema operativo es responsable de las siguientes actividades en lo que se refiere a la gestión de procesos:

- Crear y borrar los procesos de usuario y del sistema.
- Suspender y reanudar los procesos.
- Proporcionar mecanismos para la sincronización de procesos.
- Proporcionar mecanismos para la comunicación entre procesos.
- Programar procesos y subprocesos en las CPUs.

Gestión de memoria

El sistema operativo es responsable de las siguientes actividades en lo que se refiere a la gestión de memoria:

- Controlar qué partes de la memoria están actualmente en uso y por parte de quién.
- Decidir que datos y procesos (o partes de procesos) añadir o extraer de la memoria.
- Asignar y liberar la asignación de espacio de memoria según sea necesario.

Gestión del sistema de archivos

Un **archivo** es una colección de información relacionada definida por su creador. Comúnmente los archivos representan programas (tanto en formato fuente como objeto) y datos. Los archivos normalmente se organizan en directorios para hacer más fácil su uso.

El sistema operativo es responsable de las siguientes actividades en lo que se refiere a la gestión de archivos:

- Creación y borrado de archivos.
- Creación y borrado de directorios para organizar los archivos.
- Soporte de primitivas para manipular archivos y directorios.
- Asignación de archivos a los dispositivos de almacenamiento secundario.

- Copia de seguridad de los archivos en medios de almacenamiento estables (no volátiles.)

Gestión del almacenamiento masivo

Almacenamiento secundario como respaldo de la memoria principal. El sistema operativo es **responsable de las siguientes actividades** en lo que se refiere a la gestión de disco:

- Gestión del espacio libre
- Asignación del espacio de almacenamiento.
- Planificación del disco.

Gestión de sistemas de E/S

Uno de los propósitos del sistema operativo es **ocultar al usuario las peculiaridades de los dispositivos hardware** específicos. En UNIX, p.e., estas se **ocultan mediante el subsistema de E/S**. El subsistema de E/S consta de varios componentes:

- Un componente de gestión de memoria que incluye almacenamiento en búfer, gestión de cache y gestión de colas.
- Una interfaz general para controladores de dispositivo.
- Controladores para dispositivos de hardware específicos.

Solo el controlador del dispositivo conoce las peculiaridades del dispositivo específico al que está asignado.

no puedes romper algo que ya está roto

Procesos

Concepto

Podríamos definir a un **proceso** como una **entidad dinámica generada al ejecutar un programa**, sería un **código ejecutable del programa**. Además incluye una actividad representada por el **contador del programa** y por los **contenidos del registro del procesador**, por lo general también incluye una **pila de procesos** (contiene datos temporales) y una **sección de datos** (variables globales). El proceso puede incluir un **cumulo de memoria**, que es la memoria que se le asigna dinámicamente al proceso en tiempo de ejecución.

- Es un programa un proceso?

Por sí mismo un programa no es un proceso, sino una **entidad estática**, un archivo con listas de instrucciones almacenadas en disco. Mientras que un proceso es dinámico con un contador de programa que especifica la siguiente instrucción que hay que ejecutar y un conjunto de recursos asociados

Un proceso se puede **crear, ejecutar y terminar**.

Diagrama de 5 Estados (Estado del proceso)

Cuando se ejecuta un proceso se cambia de estado, el estado de un proceso se define segun la actividad que realiza, los estados en los que puede encontrarse son los siguientes:

- **Nuevo:** El proceso esta siendo creado.
- **En ejecucion:** Se esta ejecutando las instrucciones.
- **En espera:** El proceso esta esperando a que se produzca un suceso (como la terminacion de una operacion E/S o la recepcion de una senial).
- **Preparado:** El proceso esta en espera de que se lo asigne a un procesador.
- **Terminado:** Termina la ejecucion.

Elementos/PCB

Medio raro esto, pero para entender la estructura del PCB viene joya

Se caracteriza por una seria de elementos, que se almacenan en el **bloque de control del proceso**

- **Identificador:** Para distinguirlo del resto de procesos.
Estado. Si el proceso está actualmente corriendo, está en el estado en ejecución.
- **Prioridad:** Nivel de prioridad relativo al resto de procesos.
- **Contador de programa.** La dirección de la siguiente instrucción del programa que se ejecutará.
- **Punteros a memoria.** Incluye los punteros al código de programa y los datos asociados a dicho proceso, además de cualquier bloque de memoria compartido con otros procesos.
- **Datos de contexto.** Estos son datos que están presenten en los registros del procesador cuando el proceso está corriendo.
- **Información de estado de E/S.** Incluye las peticiones de E/S pendientes, dispositivos de E/S (por ejemplo, una unidad de cinta) asignados a dicho proceso, una lista de los ficheros en uso por el mismo, etc.
- **Información de auditoría.** Puede incluir la cantidad de tiempo de procesador y de tiempo de reloj utilizados, así como los límites de tiempo, registros contables, etc.
- **Valores de registro de CPU.** Se utilizan también en el cambio de contexto.
- **Espacio de direcciones de memoria.**
- **Lista de recursos asignados** (incluyendo descriptores de archivos y sockets abiertos).
- **Estadísticas del proceso.**
- **Datos del propietario** (owner).
- **Permisos asignados.**

Planificadores

frase del dia: La muerte nos mata una vez, pero la preocupacion todos los dias...

Concepto, Tipos y Objetivo

El objetivo de la planificación de procesos es asignar procesos a ser ejecutados por el procesador a lo largo del tiempo, cumpliendo así los objetivos del sistema tales como tiempo de respuesta, rendimiento y eficiencia del procesador. Generalmente se divide en tres funciones, las cuales sus nombres indican el tiempo en el que se ejecutan:

- **Planificador a largo plazo:** Selecciona procesos de la cola de trabajos y los carga en memoria para su ejecución. Controla el grado de multiprogramación, o sea el número de procesos en memoria. Esto se hace cuando se crea el proceso.
- **Planificador a corto plazo:** Selecciona entre los procesos que están preparados para ser ejecutados y asigna la CPU a uno de ellos. Administra procesos en general y se ejecuta muy seguido sobre procesos cargados en memoria.
- **Planificador a mediano plazo:** Ventajoso para eliminar procesos de la memoria y reducir el grado de multiprogramación. Luego el proceso puede volver a cargarse en memoria, continuando su ejecución en el punto en el que se interrumpió. Denominado esquema de intercambio. El planificador de mediano plazo descarga y luego vuelve a cargar el proceso.

Algoritmos de Planificación

La gordita no lo dio en clase pero esto es fundamental logi

FCFS

Primeramente veremos el algoritmo **FCFS** (Primero en llegar, Primero en servirse / o en inglés first-come-first-served). En el momento en que un proceso pasa al estado de listo, se une a la cola de listos. Cuando el proceso actualmente en ejecución deja de ejecutar, se selecciona para ejecutar el proceso que ha estado más tiempo en la cola de listos. FCFS funciona mucho mejor para procesos largos que para procesos cortos.

RR

El maldito **RR** (Round Robin), se usa para asegurar un reparto justo de la CPU entre los procesos en ejecución. Su funcionamiento se basa en la asignación de un intervalo de tiempo fijo, conocido como "quantum" o "cuanto", a cada proceso de ejecución. Todos los procesos se colocan en una cola circular y se le asigna un tiempo de ejecución en orden, de forma secuencial.

Aunque el RR puede generar una sobrecarga adicional debido a los cambios frecuentes de contexto, sobre todo con procesos de larga duración o con un quantum muy pequeño.

En resumen asigna un intervalo de tiempo a cada proceso en una cola circular, dando posibilidad a todos los procesos tengan oportunidades equitativas de ejecución

SFJ

El algoritmo **SJF** (Shortest Job First), se usa para determinar procesos que se ejecutarán a continuación en el SO, se basa en el tiempo estimado de duración de los procesos para tomar decisiones de la planificación

Le asigna a la CPU al proceso que tiene el menor tiempo de ejecución pendiente. Es decir prioriza los procesos mas cortos o pequeños en terminos de ejecución.

Ejemplo: Si un proceso mas corto mientras se ejecuta uno mas largo, el procesos mas corto se ejecuta antes que el mas largo.

El objetivo de las SJF es minimizar el tiempo promedio de espera y mejorar el rendimiento del sistema en general. Al seleccionar los procesos mas cortos primero, se pueden completar mas procesos en menos tiempo, lo que conduce a una mayor eficiencia. (Mejora la tasa de procesamiento)

Compuesto por:

- ID del proceso.
- Estado del proceso.
- Contador de Programa.
- Registros de la CPU.
- Informacion del planificador.

Cambio de contexto

Las interrupciones provocan a un SO a abandonar su tarea actual para ejecutar una rutina de kernel, cuando pasa esto se debe guardar el contexto del proceso que se ejecuta en la CPU, de modo que puede restaurar dicho contexto cuando su procesamiento concluya, suspendiendo el proceso y reanudandolo despues. El contexto se almacena en el PCB del proceso e incluye el valor de los registros de la CPU, el estado del proceso y gestion de memoria.

La conmutacion de la CPU a otro proceso requiere una salvaguarda del estado del proceso y una restauracion del estado de otro proceso diferente. Esto se conoce como el cambio de contexto.

- Salva el proceso en la PCB, luego como hay otro proceso en espera, carga el estado del proceso que entrara en ejecución, luego en otro momento se produce otra interrupcion, eso es un cambio de contexto.
- Un proceso nunca va a sacar a otro proceso, sino que son procesos del sistema operativo.

Planificacion de procesos

- Procesos independientes
No es afectados por otros procesos en ejecución, es decir no depende de otros procesos.
- Procesos Cooperativos
Si pueden ser afectados por otros procesos en ejecución, Las razones pueden ser:
Compartir informacion, y para esto es necesario que los procesos se comuniquen entre ellos por medio de memoria compartida o pasaje de mensaje.

Comunicacion entre Procesos

Directa:

Tienen que estar referenciados:

- Sincronica (Ambos se nombran)
- Asincronica (Solo el emisor se nombra).

Indirecta

Se reciben a traves de buzones o puertos (Se produce un enlace entre procesos si tienen un buzón compartido). Pueden ser univocos entre dos procesos, Asociado a mas de dos procesos, Varios enlaces y cada uno a un buzón.

- Puede ser propiedad del Proceso:
Forma parte del espacio e direcciones del mismo.
- Puede ser propiedad del Sistema Operativo:
No esta ligado a un proceso en particular, El SO debe proporcionar mecanismos para que el proceso pueda crear, borrar o comunicar un buzón.

Comando para crear procesos

- Esto lo hacemos en C bajo nivel, (como crear otro proceso)*

Usamos la sentencia *fork* para procesos padre e hijo.

```
Declaracion: pid_t pid=fork(void)
```

- Crea un proceso llamado hijo.
- El proceso hijo es un duplicado del padre.
- Los dos tienen identificadores PIDS diferentes
- Corren en espacios de memoria diferentes. Cuando se crea el hijo los espacios de memoria como las variables tienen los mismos valores a partir de aca cada uno modifica las variables en forma independiente.

Retorno de fork

```
Al padre se le retorna el valor del PID del hijo, y al hijo se le retorna 0. Si hay  
algun problema en la creacion del hijo el valor que devuelve es -1.
```

Despachador

Es el modulo que proporciona el control de la CPU a los procesos seleccionados por el planificador a corto plazo, esto implica:

- Cambio de contexto.
- Cambio al modo usuario
- Salto a la posición correcta del programa de usuario para reiniciar dicho programa.

El despachador debe ser lo mas rapido posible con el tiempo que tarda en detener un proceso e iniciar la ejecución de otro se conoce como latencia de despacho.

Hilos, SMP y Micronucleos

"se te ve el hilo..." - Tetás 2024

Procesos e Hilos

Clase del tetás sobre Hilos

Literal esta en el libro de catedra pagina 189 del maldito Silberschartz Abraham

"Es una entidad dinamica, compuesto por recursos que tiene asignado. La ejecucion en si del mismo proceso esta dada por un proceso"

El concepto viene de lo que llamamos procedimientos o rutinas osea de la programacion estructurada, proceso que se comunican entre si. La progrrmacion estructurada en esa epoca se trabajaba asi.

A partir de que tenemos mas de una CPU, la pregunta es:

- Como potenciamos esto?

Hablamos del proceso cooperativo, simultaneamente puede ejecutar dos partes o tres partes del mismo proceso, tenemos un proceso, que simultaneamente generan subprocesso o hijos, eso porque tenemos mas de una CPU, en caso de tener solo una, todos competirian por esa sola CPU.

El orden en los que deben estar los hijos es importante, para saber si se ejecuta de manera paralela o no.

Un ejemplo que facilita la comprension:

Ejemplo de oficina (proceso) con empleados (hilos) una oficina donde varios empleados están trabajando en diferentes tareas. Cada empleado representa un hilo dentro del proceso general de la oficina. Comparten el mismo espacio de trabajo, recursos e información. Esto facilita la colaboración y la comunicación directa sin necesidad de herramientas externas.

En lugar de tener varos procesos, tendremos unicamenre un numero de proceso cooperativos. Es decir un solo proceso ejecutando varios **SUBPROCESOS**. Donde cada subprocesso es el **equivalente** a uno de los tantos proceso cooperativos (concepto mas basico de hilos).

Tnedremos otro concepto de procesos, Contenedor de recursos y multiples entidades dinamicas muchos mas pequenas denominadas HILOS.

Los beneficios de los Hilos son:

- Grado de respuesta: Mas rapido, cada hilo se puede ejecutar en CPU diferentes.
- Compartir Recursos: Por defecto comparte la memoria de los recursos del proceso que lo genero.
- Economia: Memoria y Recursos
- Utilizacion de arquitectura multinucleos.

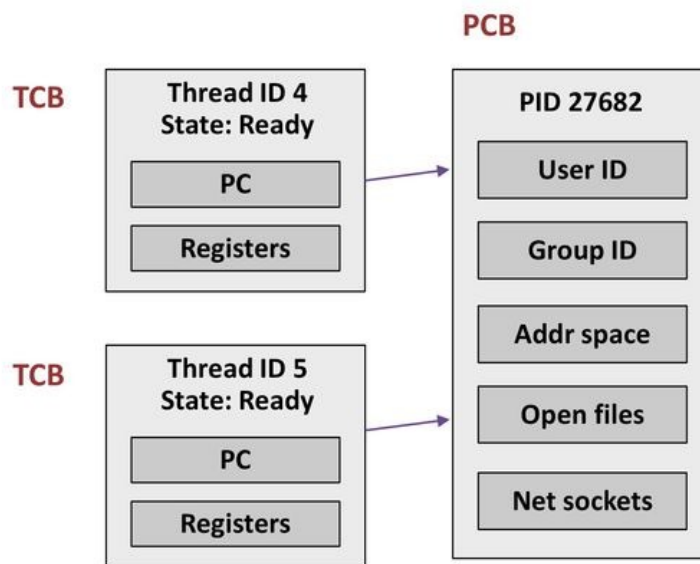
Dividimos a los hilos por la TCB (Thread Control Block): Dentro de la PCB tenemos la TCB, el estado del proceso padre y la información de los procesos hijos, recursos asignados, etc.

Para profundizar, la TCB es una estructura de datos en el núcleo del SO que tiene:

- Id del hilo
- Estado del hilo
- Información de la CPU
- Prioridad del hilo

Thread Control Block (TCB)

- **TCB contains info on a single thread**
 - Just processor state and pointer to corresponding PCB
- **PCB contains information on the containing process**
 - Address space and OS resources ... but NO processor state!



Modelos Uno-a-Uno, cada vez que nace uno a nivel de usuario nace uno a nivel de kernel (Cuando lo lanzó Microsoft década del 80)

Vamos a tener que cada hilo de usuario estará mapeado con un hilo de núcleo, este modelo permite mayor concurrencia y uso de múltiples procesadores

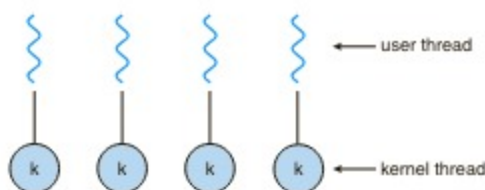


Figure 4.6 One-to-one model.

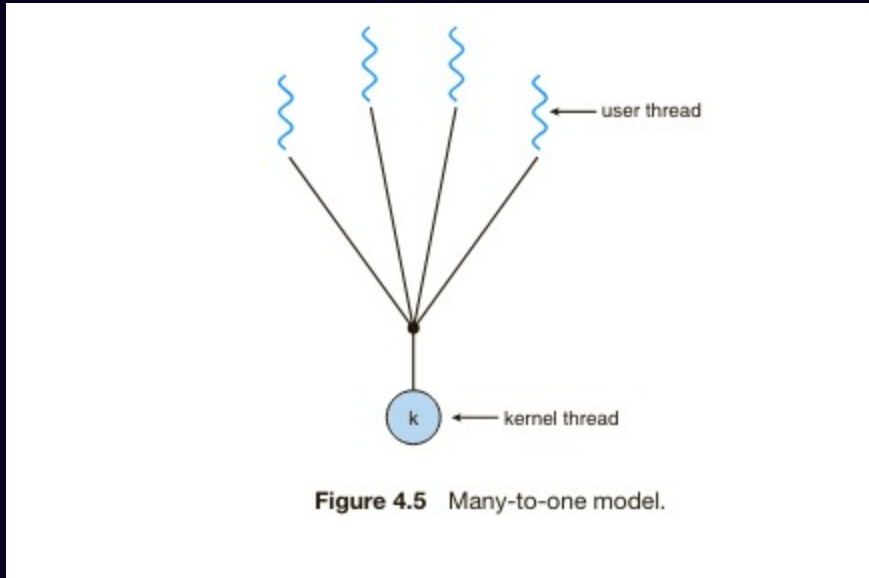
Ventajas: Cada hilo del K atiende a un hilo del U

Deventajas: Cuando se crea un hilo de usuario, va a requerir un hilo de kernell

Se pensaron en otras alternativas, como el **Muchos a Uno** (linux), donde cada hilo del proceso usuario es manejado por un hilo del kernel, empezo como una libreria portable (Pthread).

Basicamente:

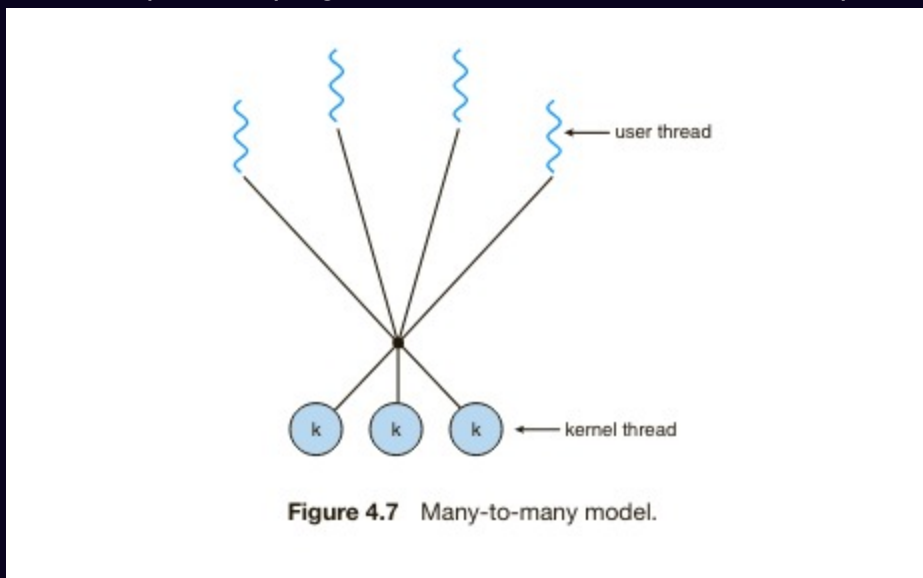
- Muchos hilso de usuario mapeados a un hilo de nucleo
- Eficiente pero bloquea todo si uno realiza una llamada de bloqueo y no aprovecha multiples nucleos.
- Usado en sistemas como Solaris



Tambien se implemento el **Muchos a Muchos**, varios hilos de usuario a un numero menor o igual de kernel.

Este modelo permite:

- Muchos hilos de usuario mapeados a un numero igual o menor de hilos de nucleo
- Permite crear muchos hilos de usuario sin limitar la concurrencia real
- El nucleo puede reprogramar otro hilos cuando uno se bloquea



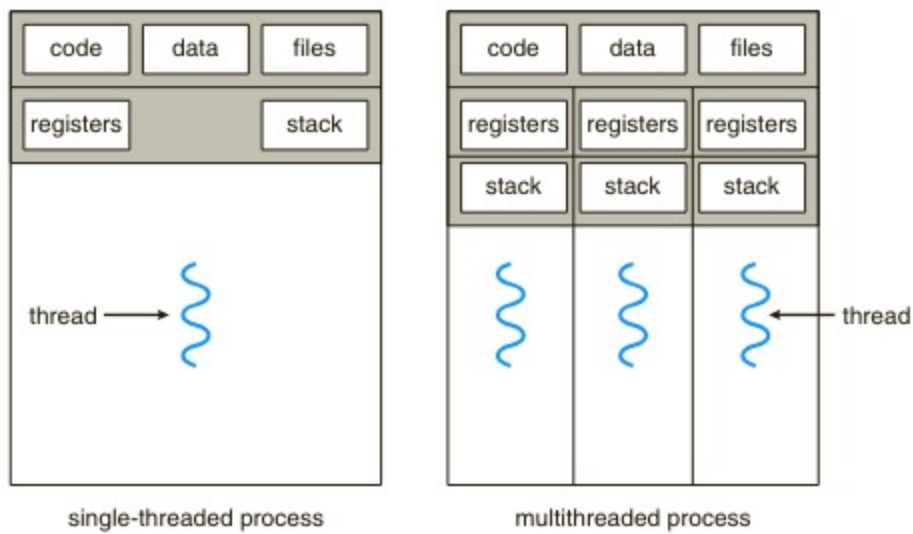


Figure 4.1 Single-threaded and multithreaded processes.

La pregunta del día, que poronga es eso? dsp de 30 min de invesigacion llegue a la siguiente conclusion:

En la imagen vemos como se comprara el proceso de un solo hilo, y un proceso multihilo
 Proceso de un solo hilo: Tiene una pila, registros, un código, datos y archivos, todo esto unicamente para un solo hilo

Proceso de multihilo: Este comparte el cdigo, los datos, archivos, etc. pero cada hilo tiene su propia pila y registros. Permite manejar la concurrencia dentro de un mismo proceso.

Lo que investigue yo:

esto seguramente no entre pero queda en tus manos si lo ves o no, quizas te sirve si quieres profundizar en conocimientos claves

Se presenta el concepto de un proceso cooposeedor de las siguietnes características:

- **Propiedad de recursos:**

Un proceso incluye direcciones virtuales para el manejo de la imagen del proceso (coleccion de programa, datos, pila y atributos definidos en el bloque de control del proceso). A un proceso se le puede asignar control o propiedad de recursos tales como la memoria principal, canales E/S, dispositivos E/S y archivos.

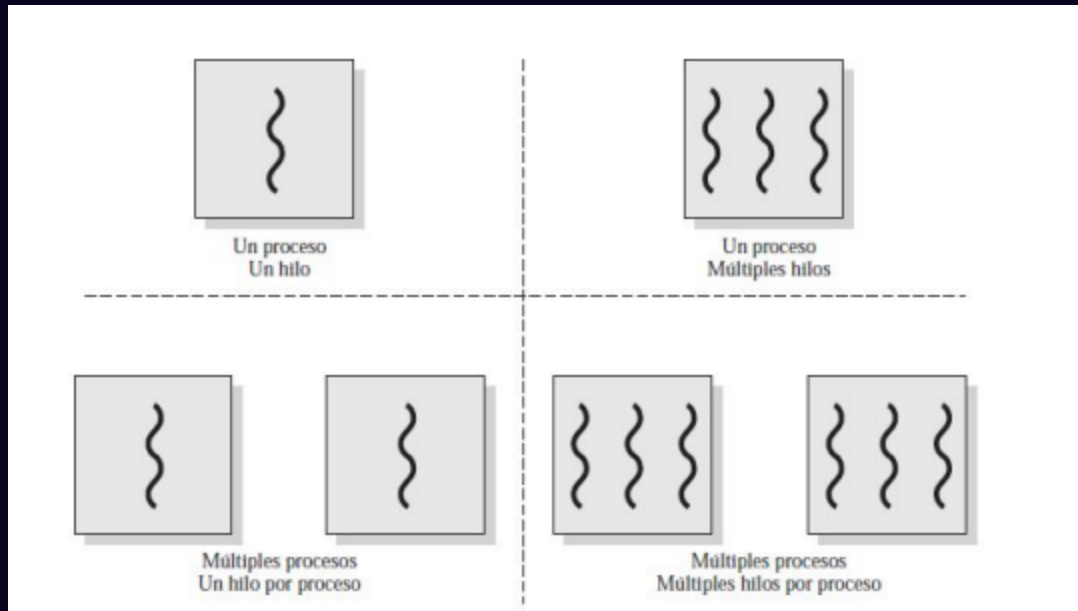
- **Planificacion/ejecucion:**

La ejecucion de un proceso sigue una ruta de ejecucion conocida como traza a traves de uno o mas programas.

En los SO tradicionales esta dos características son fundamentales, aunque ambas son independientes. Para distinguir estas características, la unidad que se activa se lo conoce como hilo (thread), o proceso ligero, mientras que la unidad de propiedad de recursos se suele denominar procesos o tarea 1.

Multihilo

Se refiere a la capacidad de un sistema operativo de dar soporte a múltiples hilos de ejecución en un solo proceso. El enfoque tradicional de un solo hilo de ejecución por proceso, en el que no se identifica con el concepto de hilo, se conoce como **estrategia monohilo**.



En un entorno multihilo, un proceso se define como la unidad de asignación de recursos y una unidad de protección. Se asocian con procesos los siguientes:

- Un espacio de direcciones virtuales que soporta la imagen del proceso.
- Acceso protegido a procesadores, otros procesos (para comunicación entre procesos), archivos y recursos de E/S (dispositivos y canales).

Dentro de un proceso puede haber uno o más hilos, cada uno con:

- Un estado de ejecución por hilo (Ejecutando, Listo, etc.)
- Un contexto de hilo que se almacena cuando no está en ejecución, una forma de ver a un hilo es como un contador de programa independiente dentro de un proceso.
- Una pila de ejecución
- Espacio de almacenamiento para variables locales (por cada hilo)
- Acceso a la memoria y recursos de su proceso, compartido con todos los hilos de su mismo proceso.

En un modelo de proceso **monohilo** (no existe el concepto de hilo), la representación de un proceso y el espacio de direcciones de usuario, además de pilas de usuario y núcleo para gestionar el comportamiento de las llamadas en la ejecución de los procesos.

Mientras el proceso está ejecutando, los registros del proceso se controlan por ese proceso, y cuando el proceso no está ejecutando, se almacena el contenido de estos registros.

En un entorno multihilo, sigue habiendo un único bloque de control del proceso y un espacio de direcciones de usuario asociado al proceso, pero ahora hay varias pilas separadas para cada hilo, así

como un bloque de control para cada hilo que contiene los valores de los registros, la prioridad, y otra informacion relativa al estado del hilo.

Por ende todos los hilos de un proceso comparten: El estado y los recursos de ese proceso, residen en el mismo espacio de direcciones y tienen acceso a los mismos datos. Cuando un hilo cambia determinados datos en memoria, otros hilos ven resultados cuando acceden a estos datos. Si un hilo abre un archivo con permisos de lectura, los demás hilos del mismo proceso pueden también leer ese archivo.

Las ventajas que tiene son:

- Lleva menos tiempo crear un nuevo hilo en un proceso existente que crear un proceso totalmente nuevo.
- Lleva menos tiempo finalizar un hilo que un proceso.
- Lleva menos tiempo cambiar entre dos hilos dentro del mismo proceso.
- Los hilos mejoran la eficiencia de la comunicación entre diferentes programas que están ejecutando. En los SO, la comunicación entre procesos independientes requiere del núcleo para proporcionar protección y los mecanismos necesarios de comunicación. Aunque los hilos dentro de los mismos procesos comparten memoria y archivos, se pueden comunicar entre ellos sin necesidad de invocar el núcleo.

A veces los hilos también pueden ser útiles en un solo procesador ya que ayudan a simplificar la estructura de programas que realizan varias funciones diferentes

Funcionalidad de los Hilos

Los hilos poseen estados de ejecución y se pueden sincronizar entre ellos.

Los estados de los hilos al igual que los procesos son: Ejecutando - Listo - Bloqueado.

Las operaciones básicas relacionadas con los hilos están asociadas con un cambio de estado del hilo:

- Creación: Cuando se crea un proceso, se crea un hilo de dicho proceso, este hilo posteriormente puede crear otro hilo dentro del mismo proceso dando un puntero a las instrucciones y los argumentos para el nuevo hilo. Al nuevo hilo se proporciona su propio registro de contexto y espacio de pila y se coloca en la cola de Listos.
- Bloqueo: Cuando un hilo debe esperar por un evento, se bloquea, almacenando los registros de usuario, contador de programa y punteros de pila. El procesador puede pasar a ejecutar otro hilo en estado Listo, dentro del mismo proceso o en otro diferente.
- Desbloqueo: Cuando sucede el evento por el que el hilo está bloqueado, el hilo se pasa a la cola de Listos.
- Finalización: Cuando se completa un hilo, se liberan su registro de contexto y pilas.

Sincronización de hilos:

Todos los hilos de un proceso comparten el mismo espacio de direcciones y otros recursos, como los archivos abiertos. Cualquier alteración de un recurso por cualquiera de los hilos, afecta al entorno del resto de los hilos del mismo proceso. Por lo que se debe sincronizar las actividades de los hilos para que no interfieran entre ellos o corrompan estructuras de datos.

Hilos a Nivel de Usuario y a Nivel de Nucleo

Dos categorías de implementación de hilos:

- Hilos de nivel de usuario (USL - user-level-threads)
- Hilos de nivel de núcleo (KLT - kernel-level-threads)

Hilos de Nivel de Usuario

- En un entorno ULT puro, la aplicación gestiona todo el trabajo de los hilos y el núcleo no es consciente de la existencia de los mismos.
- Cualquier aplicación puede programarse para ser multihilo a través del uso de una biblioteca de hilos, que es un paquete de rutinas para la gestión de ULT.
- Las ventajas de los ULT incluyen que el cambio de hilo no requiere privilegios de modo núcleo y que todas las estructuras de datos de gestión de hilos están en el espacio de direcciones de usuario de un solo proceso.

Hilos de Nivel de Nucleo

- En un entorno KLT puro, el núcleo gestiona todo el trabajo de gestión de hilos. No hay código de gestión de hilos en la aplicación, solo una interfaz de programación de aplicación (API) para acceder a las utilidades de hilos del núcleo.
Un ejemplo es Windows
- En un entorno KLT, el núcleo asigna el proceso a un solo procesador al mismo tiempo y se encarga de la planificación de hilos, el intercambio de procesos, las excepciones, el manejo de interrupciones y la sincronización de multiprocesadores.

En resumen, los hilos de nivel de usuario son gestionados por la aplicación y no requieren privilegios de modo núcleo, mientras que los hilos de nivel de núcleo son gestionados por el sistema operativo y tienen acceso directo a las funcionalidades del núcleo.

Concurrencia

frase del día: el honor es como el fuego, una vez prendido es fácil que perdure, pero si se apaga es difícil de recuperar

Concepto

Esto tampoco lo dio a profundidad, pero lo mismo que lo anterior si quieres darle bola queda en vos

Los diseños de los SO están relacionados con la gestión de procesos e hilos:

- **Multiprogramación**: Gestión de múltiples procesos dentro de un sistema monoprocesado.
- **Multiprocesamiento**: Gestión de múltiples procesos dentro de un multiprocesador.
- **Procesamiento distribuido**: Gestión de múltiples procesos que ejecutan sobre múltiples sistemas de cómputo distribuidos.

La concurrencia aparece en tres contextos diferentes:

- Múltiples aplicaciones: La multiprogramación fue ideada para compartir dinámicamente el tiempo de procesamiento entre varias aplicaciones activas.
- Aplicaciones Estructuradas: Algunas aplicaciones pueden ser programadas eficazmente como un conjunto de procesos concurrentes.
- Estructura del SO: Las mismas ventajas son aplicables a la programación de sistemas y, de hecho, los sistemas operativos son implementados en sí mismo como un conjunto de procesos o hilos.

Palabras que deberías saber...

Procesador: Controla el funcionamiento del computador y realiza sus funciones de procesamiento de datos. Cuando solo hay un procesador, se denomina usualmente unidad central de procesos.

Memoria Principal: Almacena Datos y programas. Esta memoria es habitualmente volátil. En contraste el contenido de la memoria del disco se mantiene.

Modulos E/S: Transfieren los datos entre el computador y su entorno externo. El entorno externo está formado por diversos dispositivos, incluyendo dispositivos de memoria secundaria (discos por ej).

Bus del sistema: Proporciona comunicación entre los procesadores, la memoria principal y módulos de E/S.

Shell: Interpretador de comandos, donde los usuarios generalmente interactúan con el sistema, cuando está basado en texto.

GUI: Interfaz Gráfica del Usuario, donde usa iconos o elementos gráficos.

CLI: Command Line Interface.

API: Servicios de programación.

Modo Kernel/Supervisor: Cuando el SO tiene acceso completo al hardware y puede ejecutar cualquier instrucción.

Modo Usuario: Subconjunto de las instrucciones de máquina el cual tiene permitido usar.

Parciales

Miren la parte de mi drive que dice Respuesta Parciales, ahí está todo y más

Parciales Tipo 1

Este es de UTN FRBA así que no se asusten si ven cosas raras. Las rta están en el drive tmb, pero no sean bolas y practiquen con esto`

El conjunto de system calls

Seleccione una o más de una:

- a. Ninguna opción es válida.
- b. Ninguna está regulada por normas ISO.
- c. Puede estar regulado por normas ISO.
- d. Es independiente del SO.
- e. Define el poder del SO.

En interbloqueo, la exclusión mutua es

Seleccione una o más de una:

- a. Independiente del interbloqueo.
- b. Una condición de espera lógica.
- c. Una condición necesaria más.
- d. Ninguna opción es válida.
- e. La única condición necesaria.

Un modo Dual

Seleccione una o más de una:

- a. Tiene dos modos de ejecución.
- b. Tiene dos CPU.
- c. Tiene una CPU y dos núcleos.
- d. Tiene dos terminales.
- e. Ninguna opción es válida.

El módulo objeto es generado por

Seleccione una o más de una:

- a. Ninguna opción es válida.
- b. El compilador.
- c. El editor de montaje.
- d. El Sistema Operativo.
- e. El editor de enlace.

La interfaz con el usuario puede ser

Seleccione una o más de una:

- a. Gráfica.
- b. Online.
- c. Ninguna opción es válida.
- d. Oral.
- e. Texto.

Un sistema está en estado seguro cuando

Seleccione una o más de una:

- a. Existe una secuencia segura.
- b. Evade los interbloqueos.
- c. Detecta los interbloqueos y los soluciona.

- d. Ninguna opción es válida.
- e. Hay bloqueos mutuos.

El modelo de conjunto de trabajo

Seleccione una o más de una:

- a. Es usado para administrar memoria real.
- b. Ninguna opción es válida.
- c. Es usado para planificación de CPU.
- d. Es para planificar procesos.
- e. Está basado en la suposición de la localidad.

La primera generación estaba basada en

Seleccione una o más de una:

- a. Circuitos integrados a pequeña escala.
- b. Válvulas.
- c. Transistores.
- d. Ninguna opción es válida.
- e. Circuitos lógicos.

Los recursos a un proceso nuevo le son asignados por:

Seleccione una o más de una:

- a. El proceso padre.
- b. El sistema operativo.
- c. El usuario.
- d. Ninguna opción es válida.

Una máquina virtual es

Seleccione una o más de una:

- a. Un kernel con funciones básicas.
- b. Un tipo de hardware.
- c. Una arquitectura de kernel.
- d. Una capa de software que abstrae el hardware.
- e. Ninguna opción es válida.

Un error genera

Seleccione una o más de una:

- a. Una máscara.
- b. Una interrupción por hardware.
- c. Una interrupción por software.
- d. Una excepción.
- e. Ninguna opción es válida.

Una interrupción propiamente dicho es

Seleccione una o más de una:

- a. Una interrupción producida por un error.
- b. Una interrupción DOS.

- c. Una interrupción por hardware.
- d. Una interrupción por software.
- e. Ninguna opción es válida.

Los controladores de periféricos se incorporaron para

Seleccione una o más de una:

- a. Para evitar bloqueos.
- b. Permitir la multiprogramación.
- c. Bajar los costos.
- d. Acelerar los tiempos de procesamiento.
- e. Ninguna opción es válida.

Un monitor

Seleccione una o más de una:

- a. Es una herramienta de sincronización de procesos.
- b. Es un hardware.
- c. Ninguna opción es válida.
- d. Es un tipo abstracto de datos.

La condición de retención y espera

Seleccione una o más de una:

- a. Implica que el recurso forma parte de un ciclo.
- b. Implica que el recurso no puede ser compartido.
- c. No es obligatoria para un interbloqueo.
- d. El proceso termina por tiempo.
- e. Ninguna opción es válida.

Una DMA sirve para

Seleccione una o más de una:

- a. Manejar impresoras.
- b. Acelerar los procesos.
- c. Ninguna opción es válida.
- d. Controlar la ROM.
- e. Evitar múltiples interrupciones de periféricos.

El planificador de corto plazo administra:

Seleccione una o más de una:

- a. Procesos Batch
- b. Las colas de Espera
- c. Las colas de Listo
- d. Ninguna opción es válida
- e. Los procesos a crear

La planificación por prioridad se usa para:

Seleccione una o más de una:

- a. Sistema Batch

- b. Sistemas Dual
- c. Sistemas con monoprocesamiento
- d. Sistemas interactivos
- e. Ninguna opción es válida

Un kernel monolítico contiene

Seleccione una o más de una:

- a. Múltiples módulos.
- b. Ninguna opción es válida.
- c. Un solo nivel.
- d. Dos niveles.
- e. Múltiples capas.

El tamaño de la PCB

Seleccione una o más de una:

- a. Es definido por el Hardware del procesador.
- b. Es variable durante la ejecución.
- c. Es definido por el SO.
- d. Es definido por el usuario.
- e. Ninguna opción es válida.

Un proceso pasa de ejecución a listo

Seleccione una o más de una:

- a. Ninguna opción es válida.
- b. Por una System Call.
- c. Nunca.
- d. Por espera de dispositivo.
- e. Por una interrupción de hardware.

Las minicomputadoras eran:

Seleccione una o más de una:

- a. Equipos de escritorio
- b. Equipos intermedios entre mainframes y PC
- c. Ninguna opción es válida
- d. Equipos portales
- e. Eran equipos para juegos

Un semáforo es

Seleccione una o más de una:

- a. Un tipo de variable.
- b. Parte del hardware.
- c. Ninguna opción es válida.
- d. Una herramienta de sincronización.
- e. Un administrador de procesos.

Las System Calls:

Seleccione una o más de una:

- a. Ninguna opción es válida
- b. Son las mismas en todos los S.O
- c. Existen distintos tipos
- d. Las ejecuta el usuario
- e. Son todas del mismo tipo

Linux usa la biblioteca de hilos:

Seleccione una o más de una:

- a. RMThread
- b. Phtread
- c. Ninguna opción es válida
- d. No usa bibliotecas
- e. Threading

El estado de un proceso se registra en:

Seleccione una o más de una:

- a. La PCB.
- b. Ninguna opción es válida.
- c. El port asignado al proceso.
- d. El vector de estados.

La detección de interbloqueos

Seleccione una o más de una:

- a. Ninguna opción es válida.
- b. Requiere un hardware específico.
- c. Es de simple ejecución.
- d. Tiene un alto costo asociado.
- e. Requiere de mucha memoria.

Los sistemas multi CPU son también llamados

Seleccione una o más de una:

- a. Sistemas de multiprogramación.
- b. Sistemas Multinúcleo.
- c. Sistemas Multiprocesador.
- d. Ninguna opción es válida.
- e. Sistemas Multiprocesamiento.

Los hilos (threads) los puede planificar.

Seleccione una o más de una:

- a. El propio proceso.
- b. El planificador a largo plazo.
- c. El usuario.

- d. Ninguna opción es válida.
- e. El sistema operativo.

La planificación por prioridad se usa para

Seleccione una o más de una:

- a. Sistemas con monoprocesamiento.
- b. Ninguna opción es válida.
- c. Sistemas Dual.
- d. Sistemas batch.
- e. Sistemas interactivos.

El planificador de Corto plazo administra:

Seleccione una o más de una:

- a. Periféricos.
- b. Procesos en wait.
- c. Ninguna opción es válida.
- d. Procesos.
- e. Programas.

El algoritmo SJF:

- a. Perjudica los procesos chicos
- b. Mejora los tiempos de ejecución
- c. Ninguna opción es válida
- d. Mejora la tasa de procesamiento
- e. Garantiza los tiempos de respuesta

El algoritmo SJF utiliza:

- a. No usa promedios
- b. Un promedio móvil
- c. Ninguna opción es válida
- d. Tiempos fijos
- e. Un promedio Exponencial

Los procesos pueden comunicarse:

- a. Mediante system calls
- b. Ninguna opción es válida
- c. En forma directa
- d. En forma indirecta
- e. No es una clasificación de comunicación

Los procesos pueden comunicarse:

- a. No es una clasificación de comunicación
- b. Ninguna opción es válida
- c. En forma sincrónica
- d. Mediante System Calls
- e. En forma asincrónica

Un Hilo se ejecuta:

- a. En modo kernel.
- b. En modo usuario.
- c. Ninguna opción es válida.
- d. En Microkernel con capas.
- e. Solamente en Microkernel.

Los sistemas de multi núcleo asimétrico:

- a. Ninguna opción es válida.
- b. Soportan solo procesos batch.
- c. Puede ejecutar varios procesos simultáneamente.
- d. Soportan instrucciones gráficas.
- e. Requieren múltiples procesadores.

En un cambio de contexto:

- a. Cambia el proceso en ejecución.
- b. Ninguna opción es válida.
- c. Cambian los parámetros del kernel.
- d. Cambian las variables de entorno.
- e. Los ID de procesos se guardan en la swap.

Los semáforos son:

- a. Un tipo de variable.
- b. Ninguna opción es válida.
- c. Hay muchos tipos de semáforos.
- d. Una variable de tipo entero.
- e. Parte del Hardware.

Un proceso puede pasar de wait a terminado:

- a. Mediante un comando del usuario.
- b. Nunca.
- c. Mediante una excepción.
- d. Mediante un comando del SO

El planificador de corto plazo administra

- A. Periféricos
- B. Procesos en wait
- C. Ninguna opción es válida
- D. Procesos
- E. Programas

Linux usa la biblioteca de Hilos

- A. RMThread
- B. Phtread
- C. Ninguna opción es válida

- D. No usa bibliotecas
- E. Threading

Los system calls

- A. Ninguna opción es válida
- B. Son las mismas en todos los SO
- C. Existen distintos tipos
- D. Las ejecuta el usuario
- E. Son todas del mismo tipo

Las minicomputadoras eran

- A. Equipos de escritorio
- B. Equipos intermedios entre mainframes y PC
- C. Ninguna opción es válida
- D. Equipos portátiles
- E. Eran equipos para juegos

La planificación por prioridad se usa para

- A. Sistemas batch
- B. Sistemas Dual
- C. Sistemas con monoprocesamiento
- D. Sistemas interactivos
- E. Ninguna opción es válida

El planificador de corto plazo administra

- A. Procesos batch
- B. Las colas de espera
- C. Las colas de listo
- D. Ninguna opción es válida
- E. Los procesos a crear

Los hilos de usuario pueden relacionarse con

- A. No se relacionan con el Kernel
- B. Ninguna opción es válida
- C. Procesos batch solamente
- D. Solo un hilo de kernel
- E. Múltiples hilos de Kernel

Los planificación Round Robin Se usa para

- A. Sistemas batch
- B. Usa un algoritmo exponencial
- C. Sistemas de tiempo real
- D. Ninguna opción es válida
- E. Sistemas interactivos

El procesamiento BATCH Simple

- A. Requiere la participación del usuario
- B. Es interactivo
- C. Ninguna opción es válida
- D. No requiere la participación del usuario
- E. Se usa en una terminal

El problema de sección crítica se puede solucionar

- A. No tiene solución
- B. A veces no tiene solución
- C. Usando hardware
- D. Ninguna opción es válida
- E. Usando Software

El tamaño de la PCB

- A. Ninguna opción es válida
- B. Es variable durante la ejecución
- C. Es definido por el usuario
- D. Es definido por el SO
- E. Es definido por el hardware del procesador

Los módulos objeto externo

- A. Son incorporados por el editor de montaje
- B. Ningun opción es válida
- C. Son parte del programa fuente
- D. Son incorporados por el compilador
- E. Son incorporados por el cargador

La detección de interbloqueos

- A. Tienen un alto costo asociado
- B. Requiere de mucha memoria
- C. Requiere un hardware específico
- D. Ninguna opción es válida
- E. Es de simple ejecución

Multiprocesamiento es

- A. Ninguna opción es válida
- B. Dos CPU en paralelo
- C. Varios procesos ejecutándose concurrentemente
- D. Varios procesadores en la misma placa madre
- E. Varias CPU en un solo procesador

Linux (x86) tiene una estructura

- A. Monolítica
- B. Ninguna opción es válida
- C. Por capas

- D. Microkernel con capas
- E. Microkernel

Los semáforos pueden ser de tipo

- A. Ninguna opción es válida
- B. Contador con lista de espera
- C. Contador de espera activa
- D. Espera limitada
- E. Binario

Los hilos (threads) los puede planificar

- A. El SO
- B. El planificador a largo plazo
- C. Ninguna opción es válida
- D. El usuario
- E. El propio proceso

Un monitor

- A. Es un tipo abstracto de datos
- B. Es una herramienta de sincronización de procesos
- C. Solo se utiliza en sistemas orientados a objetos
- D. Ninguna opción es válida
- E. Es un hardware

La planificación por prioridad puede sufrir de

- A. Inversión de prioridades
- B. Ninguna opción es válida
- C. Espera lógica
- D. Muerte por inanición (starvation)
- E. Muerte al esperar 3 quantum

Parcial Tipo 2

- 1:Un sistema Operativo es:
☐ A: Parte del Hardware del equipo
☒ B: Software que administra el hardware
☐ C: Un programa de usuario
☐ D: Las Opciones B y C son válidas
☐ E: Ninguna opción es válida

- 2:La primera generación estaba basada en
☒ A: Válvulas
☐ B: Transistores
☐ C: Circuitos integrados a pequeña escala
☐ D: Las Opciones A y B son válidas
☐ E: Ninguna opción es válida

- 3:La tercera generación incorpora:
☐ A: Transistores
☐ B: Circuitos integrados a pequeña escala
☐ C: Multiprogramación
☒ D: Las Opciones B y C son válidas
☐ E: Ninguna opción es válida

- 4: Las minicomputadoras eran
☐ A: Equipos de escritorio
☐ B: Equipos portátiles
☒ C: Equipos intermedios entre mainframes y PC
☐ D: Las Opciones A y B son válidas
☐ E: Ninguna opción es válida

- 5: Los sistemas de propósito general sirven para
☐ A: Cálculo Científico
☐ B: Sistemas comerciales
☐ C: Control de equipamientos por sensores
☒ D: Las Opciones A y B son válidas
☐ E: Ninguna opción es válida

- 6: Multiprocesamiento es:
☐ A: Varios procesadores en la misma placa madre
☒ B: Varios procesos ejecutándose concurrentemente
☐ C: Varios núcleos (CPU) en un solo procesador
☐ D: Las Opciones A y C son válidas
☐ E: Ninguna opción es válida

- 7: Una interrupción propiamente dicha es
☒ A: Una interrupción por hardware
☐ B: Una interrupción por software
☐ C: Una interrupción producida por un error
☐ D: Las Opciones A y C son válidas
☐ E: Ninguna opción es válida

- 8: Una excepción es
☐ A: Una interrupción por software
☒ B: Una interrupción producida por un error
☐ C: Una interrupción por hardware
☐ D: Las Opciones A y C son válidas
☐ E: Ninguna opción es válida

- 9: La memoria cache es
☐ A: Parte de la RAM
☐ B: una memoria para manejar impresoras
☒ C: una memoria intermedia rápida redundante
☐ D: Las Opciones B y C son válidas
☐ E: Ninguna opción es válida

- 10: Una DMA sirve para
☐ A: Manejar placas de video
☐ B: Manejar discos duros
☒ C: Evitar múltiples interrupciones de periféricos
☐ D: Las Opciones A y B son válidas
☐ E: Ninguna opción es válida

Parcialito 1° Parte teórica

Tema 1

- 1:Un sistema Operativo es:
☐ A: Parte del Hardware del equipo
☐ B: Un programa de usuario
☒ C: Software que administra el hardware
☐ D: Las Opciones B y C son válidas
☐ E: Ninguna opción es válida

- 2:La primera generación estaba basada en
☐ A: Transistores
☐ B: Circuitos integrados a pequeña escala
☒ C: Válvulas
☐ D: Las Opciones A y B son válidas
☐ E: Ninguna opción es válida

- 3:La tercera generación incorpora:
☐ A: Multiprogramación
☐ B: Circuitos integrados a pequeña escala
☐ C: Transistores
☒ D: Las Opciones A y B son válidas
☐ E: Ninguna opción es válida

- 4: Las minicomputadoras eran
☐ A: Equipos portátiles
☒ B: Equipos intermedios entre mainframes y PC
☐ C: Equipos de escritorio
☐ D: Las Opciones A y C son válidas
☐ E: Ninguna opción es válida

- 5: Los sistemas de propósito general sirven para
☐ A: Cálculo Científico
☐ B: Sistemas comerciales
☐ C: Control de equipamientos por sensores
☒ D: Las Opciones A y B son válidas
☐ E: Ninguna opción es válida

- 6: Multiprocesamiento es:
☐ A: Varias CPU en un solo procesador
☐ B: Varios procesadores en la misma placa madre
☒ C: Varios procesos ejecutándose concurrentemente
☐ D: Las Opciones A y B son válidas
☐ E: Ninguna opción es válida

- 7: Una interrupción propiamente dicha es
☐ A: Una interrupción por software
☒ B: Una interrupción por hardware
☐ C: Una interrupción producida por un error
☐ D: Las Opciones A y C son válidas
☐ E: Ninguna opción es válida

- 8: Una excepción es
☐ A: Una interrupción por software
☐ B: Una interrupción por hardware
☒ C: Una interrupción producida por un error
☐ D: Las Opciones A y C son válidas
☐ E: Ninguna opción es válida

- 9: La memoria cache es
☐ A: Parte de la RAM
☒ B: una memoria intermedia rápida redundante
☐ C: una memoria para manejar impresoras
☐ D: Las Opciones B y C son válidas
☐ E: Ninguna opción es válida

- 10: Una DMA sirve para
☐ A: Acelerar los procesos
☐ B: Evitar múltiples interrupciones de periféricos
☐ C: Manejar impresoras
☒ D: Las Opciones A y B son válidas
☐ E: Ninguna opción es válida

1: Un sistema Operativo es:

- ☐ A: Parte del Hardware del equipo
- ☒ B: Software que administra el hardware
- ☐ C: Un programa de usuario
- ☐ D: Las Opciones B y C son válidas
- ☐ E: Ninguna opción es válida

2: La primera generación estaba basada en

- ☒ A: Válvulas
- ☐ B: Transistores
- ☐ C: Circuitos integrados a pequeña escala
- ☐ D: Las Opciones A y B son válidas
- ☐ E: Ninguna opción es válida

3: La tercera generación incorpora:

- ☐ A: Transistores
- ☐ B: Circuitos integrados a pequeña escala
- ☐ C: Multiprogramación
- ☒ D: Las Opciones B y C son válidas
- ☐ E: Ninguna opción es válida

4: Las minicomputadoras eran

- ☐ A: Equipos de escritorio
- ☐ B: Equipos portátiles
- ☒ C: Equipos intermedios entre mainframes y PC
- ☐ D: Las Opciones A y B son válidas
- ☐ E: Ninguna opción es válida

5: Los sistemas de propósito general sirven para

- ☐ A: Cálculo Científico
- ☐ B: Sistemas comerciales
- ☐ C: Control de equipamientos por sensores
- ☒ D: Las Opciones A y B son válidas
- ☐ E: Ninguna opción es válida

6: Multiprocesamiento es:

- ☐ A: Varios procesadores en la misma placa madre
- ☒ B: Varios procesos ejecutándose concurrentemente
- ☐ C: Varios núcleos (CPU) en un solo procesador
- ☐ D: Las Opciones A y C son válidas
- ☐ E: Ninguna opción es válida

7: Una interrupción propiamente dicha es

- ☒ A: Una interrupción por hardware
- ☐ B: Una interrupción por software
- ☐ C: Una interrupción producida por un error
- ☐ D: Las Opciones A y C son válidas
- ☐ E: Ninguna opción es válida

8: Una excepción es

- ☐ A: Una interrupción por software
- ☒ B: Una interrupción producida por un error
- ☐ C: Una interrupción por hardware
- ☐ D: Las Opciones A y C son válidas
- ☐ E: Ninguna opción es válida

9: La memoria cache es

- ☐ A: Parte de la RAM
- ☐ B: una memoria para manejar impresoras
- ☒ C: una memoria intermedia rápida redundante
- ☐ D: Las Opciones B y C son válidas
- ☐ E: Ninguna opción es válida

10: Una DMA sirve para

- ☐ A: Manejar placas de video
- ☐ B: Manejar discos duros
- ☒ C: Evitar múltiples interrupciones de periféricos
- ☐ D: Las Opciones A y B son válidas
- ☐ E: Ninguna opción es válida