**TEORÍA**

1. Registros visibles para el usuario:

* Registro Índice
* Puntero de Segmento
* Puntero de pila

Registro de Control y Estado:

* Contador de Programa
* Registro Instrucción

1. El acceso a E/S mediante programa funciona cuando el procesador ejecuta un programa y encuentra una instrucción relacionada con la E/S, ejecuta esa instrucción generando un mandato al módulo de E/S apropiado. En este caso, el módulo de E/S realiza la acción solicitada y fija los bits correspondientes en el registro de estado de E/S, pero no realiza ninguna acción para avisar al procesador. Por lo tanto, después de la invocación de la instrucción de E/S, el procesador debe tomar un papel activo para determinar cuándo se completa la instrucción E/S.
2. Los objetivos de un S.O son:

* Facilidad de Uso: Un sistema operativo facilita el uso de un computador.
* Eficiencia: Un SO permite que los recursos de un sistema de computación se puedan utilizar de manera eficiente.
* Capacidad para evolucionar: Un SO se debe construir de tal forma que se puedan desarrollar, probar e introducir nuevas funciones en el sistema sin interferir con su servicio.

1. El modelo de ejecución por lotes fue desarrollado a mediados de los años ’50 por General Motors para el uso de un IBM 701. La idea central es el uso de una pieza de software denominado monitor. Con este tipo de SO, el usuario no tiene que acceder directamente a la máquina. El usuario envía un trabajo a través de una tarjeta o cinta al operador del computador, que crea un sistema por lotes con todos los trabajos enviados y coloca la secuencia de trabajos en el dispositivo de entrada, para que lo utilice el monitor. Cuando un programa finaliza su procesamiento, devuelve el control al monitor, punto en el cual dicho monitor comienza la carga del siguiente programa.

Ventajas:

* Eficiencia en uso de recursos.
* Planificación simplificada.
* Mayor rendimiento del sistema.

1. Avances logrados por el SO:

* Procesos
* Gestión de memoria
* Protección y seguridad de la información
* Planificación y gestión de los recursos
* Estructura del sistema

1. El SO está formado por 3 capas:

* Kernel
* Servicios
* Intérprete de mandatos (SHELL)

Gestor de Recursos (UCP, memoria, …)

* Asignación y recuperación de recursos
* Protección de los usuarios
* Contabilidad
* Soporte de usuario

Máquina Extendida (servicios)

* Ejecución de programas (procesos)
* Órdenes E/S
* Operaciones sobre archivos
* Detección y tratamiendo de errores

Interfaz de usuario

* Shell/GUI

1. –
2. Asignar un identificador de proceso único al proceso.
3. Reservar espacio para proceso.
4. Inicialización del bloque de control de acceso.
5. Establecer los enlaces apropiados.
6. Creación o expansión de otras estructuras de datos.
7. El estado “Bloqueado” es un proceso que no se puede ejecutar hasta que se cumpla un evento determinado o se complete una operación E/S. En cambio, el estado “Suspendido” es un proceso que se encuentra en un estado inactivo donde el proceso se retira de la memoria principal y se guarda en un almacenamiento secundario con el fin de liberar recursos.
8. El SO como un conjunto de procesos es una alternativa donde las principales funciones del núcleo se organizan como procesos independientes. Entre las ventajas de esta visión, es que refuerza el uso de sistemas operativos modulares con mínimas y claras interfaces entre los módulos. Además, otras funciones del SO que no sean críticas están separadas como otros procesos. Y también, se podría incrementar el rendimiento al enviar a procesadores dedicados determinados servicios.
9. El concepto de Multihilo se refiere a la capacidad de un sistema operativo de dar soporte a múltiples hilos de ejecución en un solo proceso.
10. Los beneficios que tiene el ULT (User Level Thread) son:

* El cambio de hilo no requiere privilegios de modo núcleo porque todas las estructuras de datos de gestión de hilos están en el espacio de direcciones de usuario de un solo proceso. Por consiguiente, el proceso no cambia a modo núcleo para realizar la gestión de hilos. Esto ahorra la sobrecarga de dos cambios de modo.
* La planificación puede especificarse por parte de la aplicación.
* Los ULT pueden ejecutarse en cualquier sistema operativo. No se necesita ningún cambio en el nuevo núcleo para dar soporte a los ULT.

1. Las desventajas de los sistemas de tipo MicroKernel son:

* La principal desventaja que tiene es la del rendimiento ya que lleva más tiempo construir y enviar un mensaje a través del micronúcleo, y aceptar y decodificar la respuesta, que hacer una simple llamada a un servicio.
* Complejidad en la implementación.
* Dificultades en la compatibilidad y portabilidad.

1. Los diferentes escalas de agendamiento de procesos son:

* Planificación a largo plazo: La decisión de añadir un proceso al conjunto de procesos a ser ejecutados.
* Planificación a medio plazo: La decisión de añadir un proceso al número de procesos que están parcialmente o totalmente en la memoria principal.
* Planificación a corto plazo: La decisión por la que un proceso disponible será ejecutado por el procesador.
* Planificación de la E/S: La decisión por la que un proceso que está pendiente de una petición de E/S será atendido por un dispositivo de E/S disponible.

1. Criterios de planificación orientados al usuario:

* Tiempo de estancia (turnaround time): Tiempo transcurrido desde que se lanza un proceso hasta que finaliza. Incluye el tiempo de ejecución sumado con el tiempo de espera por los recursos, incluyendo el procesador.
* Tiempo de respuesta (response time): Para un proceso interactivo, es el tiempo que trascurre desde que se lanza una petición hasta que se comienza a recibir la respuesta. A veces un proceso producir alguna salida al usuario mientras continúa el proceso de la petición.
* Fecha tope (deadlines): Cuando se puede especificar la fecha tope de un proceso, el planificador debe subordinar otros objetivos al de maximizar el porcentaje de fechas tope conseguidas.
* Previsibilidad: Un trabajo dado debería ejecutarse aproximadamente en el mismo tiempo y con el mismo coste a pesar de la carga del sistema.

1. El algoritmo First Come First Served (FCFS) es un tipo de algoritmo no apropiativo y penaliza a los procesos cortos. Este algoritmo asigna la CPU a los procesos en el orden en que llegan.

Tabla

Descripción generada automáticamente con confianza media

1. Los modos de decisión que existen son:
2. Cuando un proceso se bloquea en espera de un evento.
   * Realización de una llamada al sistema
3. Cuando se produce una interrupción
   * Interrupción del reloj
   * Interrupción de fin de E/S
4. Fin de Proceso
   * Planificación no apropiativa
   * Planificación apropiativa
5. Niveles de granularidad:
6. Fino: Paralelismo inherente en un único flujo de instrucciones.
7. Medio: Procesamiento paralelo dentro de una única aplicación.
8. Grueso: Multiprocesamiento de procesos concurrentes en un entorno multiprogramado.
9. Muy grueso: Procesamiento distribuido entre nodos de una red para conformar un único entorno de computación.
10. Independiente: Múltiples procesos no relacionados.
11. Los problemas que surgen al utilizar un programa concurrente en un solo procesador son:

* Consumo de tiempo en la creación y terminación de procesos.
* Consumo de tiempo en cambios de contexto.
* Problemas en la compartición de recursos.

1. Los sistemas operativos de tiempo real pueden ser caracterizados por:

* Cambio de proceso o hilo rápido.
* Pequeño tamaño (que está asociado con funcionalidades mínimas)
* Capacidad para responder rápidamente a interrupciones externas.
* Multitarea con herramientas para la comunicación entre procesos como semáforos, señales y eventos.
* Utilización de ficheros secuenciales especiales que pueden acumular datos a alta velocidad.
* Planificación expulsiva basada en prioridades.
* Minimización de los intervalos durante los cuales se deshabilitan las interrupciones.
* Primitivas para retardar tareas durante una cantidad dada de tiempo y para parar/retomar tareas.
* Alarmas y temporizaciones especiales.

1. La planificación de tiempo real depende de cuando el sistema realiza análisis de planificabilidad; si se realiza estática o dinámicamente; y de si el resultado del análisis produce un plan de planificación de acuerdo al cual se desarrollarán las tareas en tiempo de ejecución. Encontramos los siguientes tipos de algoritmos a partir de las consideraciones anteriormente mencionadas:
2. Enfoques estáticos dirigidos por tablas.
3. Enfoques estáticos expulsivos dirigidos por prioridad.
4. Enfoques dinámicos basados en un plan.
5. Enfoques dinámicos de mejor esfuerzo.