



Universidad de Guadalajara.

Centro Universitario de Ciencias Exactas e Ingenierías.

DIVISIÓN DE TECNOLOGÍAS PARA LA INTEGRACIÓN CIBER-
HUMANA.

DEPARTAMENTO DE CIENCIAS COMPUTACIONALES.

TEMA: Tarea 3

NOMBRE DEL ESTUDIANTE: Padilla Perez Jorge Daray.

NOMBRE DE LA MATERIA: Seminario de Sistemas operativos

NOMBRE DEL PROFESOR: Julio Esteban Valdés López

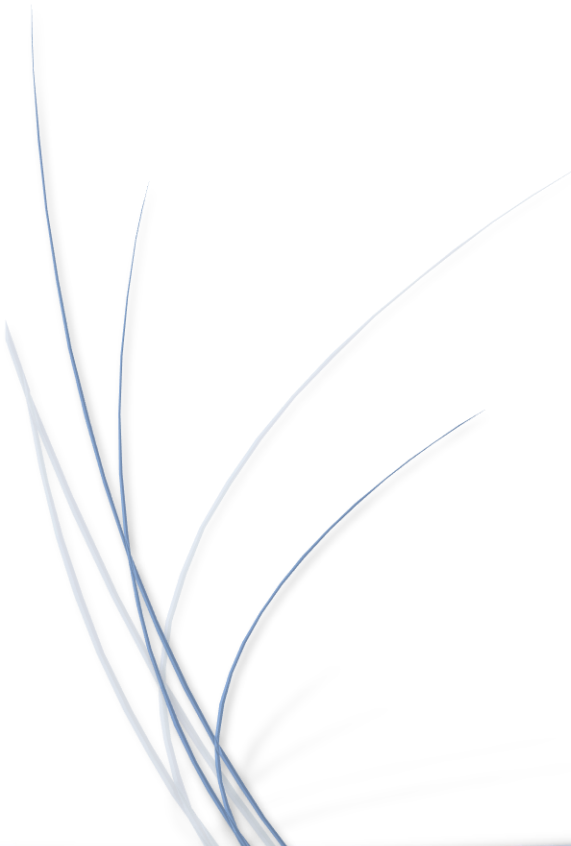


Table of Contents

Preguntas	¡Error! Marcador no definido.
¿Qué significa JCL?	¡Error! Marcador no definido.
Escriba la diferencia entre el procesamiento por lotes y el de procesamiento por lotes con multiprogramación.....	¡Error! Marcador no definido.
Según la evolución del Sistema Operativo, liste lo que siguió al proceso por lotes con multiprogramación.....	¡Error! Marcador no definido.
Escribe una de las utilidades de la interrupción int86 en C.	¡Error! Marcador no definido.
¿Para qué sirve la función Kbhit?	¡Error! Marcador no definido.
Investigue el equivalente de Kbhit (utilizado en c) en otros dos lenguajes de programación y escríbalos	¡Error! Marcador no definido.
Bibliografía	9

Que son los algoritmos de planificación

Los algoritmos de planificación son reglas o estrategias utilizadas por el sistema operativo para administrar la ejecución de procesos en un sistema de cómputo. Su objetivo principal es optimizar el uso de los recursos del sistema, como la CPU, la memoria y los dispositivos de entrada/salida.

Tiempo de espera: Es el tiempo que un proceso permanece en espera en la cola de ejecución.

Tiempo de retorno: Es el tiempo desde que se lanza un proceso hasta que finaliza.

Tiempo de respuesta: Representa el tiempo que un proceso bloqueado tarda en entrar en ejecución.

Uso de CPU: Indica el porcentaje de tiempo que la CPU está ocupada.

Productividad: Mide el número de procesos realizados en una unidad de tiempo.

Existen dos tipos principales de algoritmos de planificación:

- Apropiativos (o expropiativos): Estos algoritmos permiten la expulsión de procesos en ejecución para dar paso a otros. Por ejemplo:

Round Robin: Asigna a cada proceso un tiempo equitativo y los trata por igual.

SRTF (Shortest Remaining Time Next): Expulsa procesos largos para ejecutar otros más cortos.

- No apropiativos: En este caso, un proceso en ejecución no puede ser interrumpido. Solo cuando termina o se bloquea, se permite la entrada de un nuevo proceso. Ejemplos:

FCFS (First-Come, First-Served): El primer proceso que llega se ejecuta primero.

SJF (Shortest Job First): Prioriza los procesos más cortos, pero puede causar inanición de procesos largos.

Que son las políticas de planificación

Diversidad de Políticas:

Las políticas de planificación pueden variar según el contexto y el sistema en el que se aplican. Algunos ejemplos comunes incluyen:

Prioridad de Procesos: Algunos procesos pueden tener prioridad sobre otros. Por ejemplo, los procesos críticos para la seguridad del sistema pueden recibir una prioridad más alta.

Equidad: Se busca distribuir el tiempo de CPU de manera justa entre los procesos. Esto evita que un proceso acapare todos los recursos.

Tiempo Compartido: En sistemas interactivos, se asigna un tiempo de CPU a cada proceso en turnos. Esto permite que varios usuarios compartan el procesador.

Optimización de Rendimiento: Algunas políticas buscan minimizar el tiempo de respuesta o maximizar la productividad.

La elección de una política depende del tipo de sistema, sus objetivos y las necesidades específicas.

Implementación Técnica:

Las políticas de planificación se implementan mediante algoritmos en el sistema operativo. Estos algoritmos determinan cómo se asigna el tiempo de CPU a los procesos.

Los algoritmos pueden ser no apropiativos (como FIFO o SJF) o apropiativos (como Round Robin).

La elección del algoritmo afecta el rendimiento general del sistema y la experiencia del usuario.

Consideraciones Prácticas:

Las políticas deben equilibrar diferentes objetivos: rendimiento, equidad, eficiencia, etc.

Los administradores deben ajustar las políticas según las necesidades específicas del sistema y los usuarios.

La monitorización constante y la adaptación son esenciales para mantener un sistema eficiente.

Preguntas.

1. Describa en qué consisten los algoritmos de planificación No Apropiativos.

Planificación No Apropiativa:

En este enfoque, una vez que un proceso comienza a ejecutarse, no puede ser interrumpido por el sistema operativo hasta que termine o se bloquee esperando una operación de entrada/salida o al solicitar algún servicio del sistema:

FIFO (First In First Out): También conocido como “primero en entrar, primero en salir” (FCFS), procesa los trabajos en el orden en que llegan. Sin embargo, no considera el estado del sistema ni las necesidades de recursos individuales de los procesos, lo que puede llevar a un rendimiento deficiente.

SJF (Shortest Job First): Planifica los procesos según su tiempo de ejecución estimado. El proceso más corto se ejecuta primero. Aunque eficiente en términos de tiempo de respuesta, puede causar inanición de procesos más largos

2. Explique cómo obtener cada uno de los tiempos solicitados en el punto 9 del programa 3 (actividad de aprendizaje 6).

Para obtener el tiempo de los nuevos ingresados esta fácil, simplemente se debe hacer un random, con cada ingreso nuevo con un tiempo de 10 a 15 segundos en nuestro caso, también con su implementación de la pausa, se agrega un contador nuevo que al pausar agregue un contador para la pausa que es básicamente un iterador en 0 que tenga el tiempo que se congela el proceso. Y por último de este está cuando se termina un programa enviar el tiempo a 0 y no dejarlo esperar.

Para los procesos listos a ejecutarse simplemente nos basamos en la recopilación del tiempo anterior que tiene, y en cuanto acabe el programa ya sea si lo acabo en el tiempo que se dijo, tardo mas o simplemente se termino o pauso, este tenga el tiempo actualizado en todo momento.

Par el proceso en ejecución esta fácil, simplemente con el random generado anteriormente le asignamos un iterador que cada segundo le reste 1.

Para los bloqueados aplicamos casi lo mismo que la pausa solo que esta no se quita con la pausa si no hasta que se vuelva a unir a la cola de procesos.

Los terminados solo se imprime el tiempo que se estuvo ejecutando ya sea con pausa, con bloqueo o como haya sido, mientras estuviera ejecutándose se pone un temporizador el cual va guardando el tiempo global de este.

El tiempo de llegada se apeg a lo dicho en los procesos se genera un random de 10 a 15 segundos y se le asigna.

El tiempo de finalización esta fácil es un contador global del proceso desde que inicia su ejecución hasta que termina.

Tiempo de Retorno: Tiempo total desde que el proceso llega hasta que termina, como este dice básicamente el contador se inicia desde que se agrega al programa y hasta que se acaba deja de contar.

d. Tiempo de Respuesta: Tiempo transcurrido desde que llega hasta que es atendido por primera vez, este tiempo se obtiene si es el primero se le asigna nulo o 0 en todo caso, pero si es cualquier otro se le asigna el valor inicial de tiempo de llegada o ejecución, y se le suma dependiendo cuantos tenga por delante.

e. Tiempo de Espera: Tiempo que el proceso ha estado esperando para usar el procesador, este simplemente cuando se esta en espera se inicializa un contador al detectar la pausa, se detiene y reanuda el contador principal del proceso al volver a iniciar el proceso o darle a reanudar.

f. Tiempo de Servicio: Tiempo que el proceso ha estado dentro del procesador. (Si el proceso termino su ejecución normal es el TME), este es el contador global que no le importa si se pauso o termino abruptamente simplemente en cuanto se agrega al programa se inicializa su contador y hasta que se termine el proceso deja de contar.

3. ¿Qué significa BCP?

BCP (Plan de Continuidad del Negocio, por sus siglas en inglés) es un plan estratégico que permite a las organizaciones prevenir y recuperarse ante posibles amenazas o desastres.

Definición: El BCP es un plan de acción que permite a la empresa responder de manera adecuada ante situaciones de emergencia, minimizando su impacto en la organización y garantizando la continuidad del negocio.

Objetivos:

Prevenir: El BCP busca anticiparse a posibles riesgos y amenazas.

Recuperar: En caso de desastre, el BCP habilita las operaciones continuas antes y durante la ejecución del plan de recuperación ante desastres.

Componentes:

Análisis Integral: El BCP considera mucho más que solo la infraestructura tecnológica. Examina recursos, procesos y funciones críticas para volver a la operación normal.

Plan Documentado: El BCP es un documento que establece los pasos a seguir cuando una organización se ve afectada por escenarios inesperados.

Reducción del Tiempo de Inactividad: Un buen BCP reduce la cantidad de tiempo de inactividad, permitiendo una recuperación más rápida

4. Liste los elementos que conforman un BCP.

Identificación de Amenazas:

Se debe crear una lista de incidentes o amenazas que puedan interrumpir la actividad normal de la compañía. Esto incluye no solo desastres naturales, sino también otros eventos como interrupciones en la cadena de suministro o caídas en la demanda.

Análisis de Impacto:

Determinar las áreas críticas para la supervivencia de la empresa. ¿Qué partes de la organización son esenciales para seguir funcionando? Esto ayuda a priorizar la respuesta y la recuperación.

Plan de Respuesta y Recuperación:

Crear un plan detallado que establezca cómo la empresa responderá ante las emergencias. Esto incluye procedimientos paso a paso para mantener la continuidad del negocio.

Identificar contactos de emergencia, tanto dentro de la organización como con proveedores de sitios de respaldo.

Administradores del Plan:

Designar a personas responsables de implementar y mantener el BCP. Estos administradores deben estar familiarizados con los procesos críticos y las estrategias de recuperación.

Activos y Tecnología Necesarios:

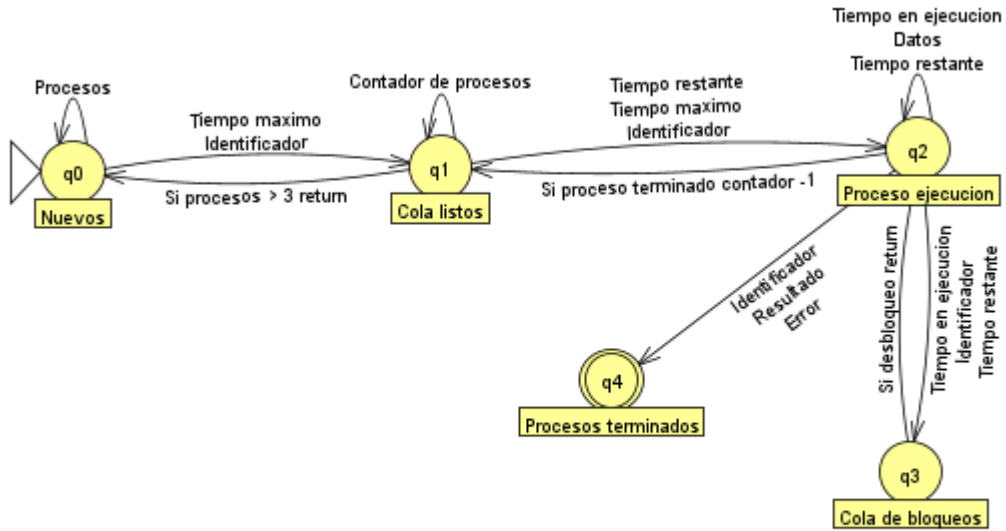
Identificar los recursos críticos necesarios para la operación continua. Esto puede incluir tecnología, insumos, personal esencial, etc..

Pruebas y Evaluación:

Probar regularmente el BCP para asegurarse de que funcione correctamente.

Evaluar y actualizar el análisis de riesgos y las estrategias según sea necesario

5. Dibuje el diagrama de 5 estados y sus transiciones válidas.



Bibliografía

La Red Martínez, D. L. (2001). Sistemas Operativos. Universidad Nacional del Nordeste. ISBN: 9874342994, 978987434299712

Castellanos, L. (2014). Sistemas Operativos: una guía de estudios

Jaramillo Morales, C. O. (s. f.). Compilación unidad temática: sistemas operativos

¿Cómo puedo usar la función kbhit en getch, cuando estoy usando la biblioteca getch en Python y usando Linux?. (2024, 18 de febrero). Recuperado de

<https://stackoverflow.com/questions/.../how-can-i-use-kbhit-function-in-getch-when-i-am-using-getch-library-in-python>

Autor desconocido. (2024). Pregunta sobre código antiguo de C en OpenWatcom. Stack Overflow. Recuperado de <https://stackoverflow.com/questions/12345678/old-c-code-in-openwatcom>

La evolución histórica de los sistemas operativos: un recorrido por la tecnología informática. (s. f.). Recuperado el 18 de febrero de 2024, de <https://www.example.com/evolucion-sistemas-operativos>