Actividad Optativa

Programación de Sistemas Distribuidos

José María Fernández Gómez

INDICE

UNIVERSIDAD NEBRIJA

Sistemas Operativos Distribuidos

1.1 INTRODUCCIÓN

Los sistemas operativos distribuidos guardan muchas similitudes en común con los GPOS o sistemas operativos locales, como pueden ser la capa de abstracción entre hardware e interfaces, capa de transparencia entre procesos y usuarios y gestión , automatización de tareas y seguridad y gestión de acceso y permisos.

1.2 RELACIÓN CON CONCEPTOS VISTOS

Los sistemas operativos en red buscan crear una capa de abstracción para favorecer la transparencia del sistema global, ya que posee muchas Apis propias de cada sistema y se encarga de poder hacer un uso compartido de recursos. Por otro lado, se ataja el parámetro de diseño de la concurrencia, al ser capaz de gestionar las tareas a través de procesos e hilos que gestionen las peticiones realizadas al sistema. Por otro lado, desde el punto de vista de la heterogeneidad, al estar actuando sobre un mismo sistema distribuido global, los sistemas operativos subyacentes o los que usen máquinas externas para comunicarse con este no deberían entrar en conflicto con este. En cuanto a seguridad, el sistema está dotado de mecanismos seguros, como la encapsulación de tareas en hilos y procesos que gestiona el S.O.

2. QUÉ PROPORCIONAN LOS SO DISTRIBUIDOS

   Los sistemas Operativos distribuidos proporcionan esa capa de abstracción entre hardware e interfaces que también proporcionan los sistemas operativos clásicos, pero con la diferencia de tener que atajar también las problemáticas y desafíos de los sistemas distribuidos, y atender a sus parámetros de diseño.

3. QUÉ REQUIEREN LOS SISTEMAS OPERATIVOS DISTRIBUIDOS
   3.1. Gestor de procesos
   3.2. Gestor de hilos
   3.3. Gestor de comunicaciones
   3.4. Gestor de memoria
   3.5. Supervisor
4. EJEMPLOS DE SISTEMAS OPERATIVOS EN RED

   - Windows server
   - Cisco IOS
   - SONIC
   - Fortinet

5. EJERCICIO DE RAZONAR ¿Es posible depurar un programa multihilo?
   Sí, es posible pero con restricciones, se puede hacer de manera externa, con programas que faciliten el acceso a cada hilo en todo momento, pero siempre teniendo en cuenta que el propio monitoreo afectará al rendimiento del sistema , y al ser un programa multihilo, el planificador del sistema operativo tendrá que tener en cuenta el propio monitoreo y debbugging, ya que si hacemos este mediante breakpoints, la ejecución de un hilo dependiente de otro podría verse afectada, al tener bloqueado y en espera a ese primero.

6. HILOS Y PROCESOS

Los hilos son menos costosos de crear que los procesos, pero el acceso a memoria es compartido, por lo que este se debe controlar bien, ya que podrían interactuar entre si, corrompiendo datos y accediendo a zonas críticas concurrentemente.

7. MODELOS DE SISTEMA

- Estación de trabajo
  - Sencillo
  - Poder de computo fijo
  - Usuarios autónomos por nodo
  - Recursos escalables
  - No es eficiente
- Pila de procesadores
  - Modelo centralizado
  - Usuarios ven un único recurso
  - Adaptativo
  - No se pueden usar nodos por separado

8. ALGORITMOS DE ASIGNACIÓN DE PROCESADORES

8.1. Modelaje

8.1.1. Deterministas -> Conocimiento total

8.1.2. Heurísticos-> No modelables

8.2. Ubicación

8.2.1. Centralizados->Gestión por tablas de prioridad

8.2.2. Remantes->Cada proceso anuncia su peso en archivo compartido

8.3. Eficiencia

8.3.1. Óptimos

8.3.2. Subóptimos

9. EJERCICIO.

Imaginemos que estamos intentando pensar en un algoritmo de asignación de procesadores en base al nivel de carga de los distintos procesadores. Si tuvieras que definir en una fórmula el índice de carga, ¿a qué lo igualarías? Intenta iterar sobre la fórmula para ver si eres capaz de mejorarla

---

P1->60%
P2->80%
P3->20%

Podemos hacerlo de manera estadística, reasignando tareas a distintos procesadores hasta que de edia tengan una carga similar:

X = (60 + 80 +20)/3= 160/3 = 50.3 %
P1 -> 50.3 %
P2 -> 50.3 %
P3 -> 50.3 %

Por otro lado, podríamos tener esta media ponderada, con la capacidad de procesamiento de cada procesador ,asumiendo que tienen distintas entre sí , vamos a tomar las operaciones de w/r o e/s como manera de calcular la carga que tiene cada procesador.

P1 Capacidad de cómputo -> 10U
P2 Capacidad de cómputo -> 2 *P1
P3 Capacidad de cómputo = ½ * P1

P1-> 200 Ops/s
P2 -> 300 Ops/s
P3 -> 50 Ops/s
Total Ops/s -> 550 Ops/S

Media 550 / 3 = 183 ops /s

Operaciones tras ponderación :

P1-> 183 Ops/s
P2 -> 366 Ops/s
P3 -> 50 Ops/s

---

10. TOLERANCIA A FALLOS

Se tiene que diseñar el sistema para que sea capaz de realizar la ejecución correcta y completa del procesado y tareas pese a la aparición de fallos.