

Simulación de sistema distribuido

Descripción del Proyecto

El proyecto consiste en desarrollar un **emulador de sistema operativo distribuido** que permita la gestión de recursos y procesos en un entorno distribuido simulado. Los estudiantes crearán un entorno de nodos interconectados, donde cada nodo representará una instancia básica de un sistema operativo capaz de realizar tareas de procesamiento, comunicación y sincronización con otros nodos. Este sistema debe ser capaz de distribuir procesos y recursos entre los nodos, asegurando la sincronización y coordinación entre ellos y la tolerancia a fallos básicos.

Objetivos de Aprendizaje

- Aplicar conceptos de sistemas distribuidos para gestionar recursos y procesos en un entorno distribuido.
- Implementar mecanismos de sincronización y comunicación para lograr la coordinación entre los nodos.
- Evaluar la escalabilidad y tolerancia a fallos, permitiendo probar el sistema en diferentes escenarios de carga y en situaciones de fallo simulado.

Requerimientos Funcionales

Funcionalidad de los Nodos

- **Gestión de Procesos:** Cada nodo debe poder ejecutar procesos y permitir la asignación dinámica de estos a cualquier nodo disponible en la red.
- **Gestión de Recursos Compartidos:** Los nodos deben administrar el acceso a recursos compartidos de manera coordinada, evitando condiciones de carrera y conflictos.
- **Sincronización entre Nodos:** Implementar mecanismos de sincronización para gestionar el acceso a recursos compartidos.
- **Manejo de Fallos:** Cada nodo debe detectar si otro nodo está inactivo o fuera de servicio y redistribuir los procesos asignados a ese nodo.

Funcionalidad de la Red Distribuida

- **Comunicación entre Nodos:** Implementar un sistema de comunicación que permita a los nodos enviar y recibir mensajes sobre el estado de los recursos y la disponibilidad de procesos.
- **Asignación Dinámica de Procesos:** El sistema debe permitir que los procesos puedan ser transferidos de un nodo a otro si un nodo está sobrecargado o falla.
- **Balanceo de Carga:** Implementar un algoritmo básico de balanceo de carga para asegurar que los procesos se distribuyan de manera uniforme entre los nodos.
- **Escalabilidad:** El emulador debe permitir agregar nuevos nodos a la red distribuida sin necesidad de detener el sistema.
- **Tolerancia a Fallos:** Si un nodo falla, el sistema debe poder detectar el fallo, informar a los otros nodos y redistribuir los procesos en los nodos activos restantes.

Casos de Uso

Caso de Uso 1: Asignación de Proceso a un Nodo

- **Actores:** Usuario, Nodo
- **Descripción:** El usuario solicita ejecutar un proceso en la red distribuida. El sistema identifica el nodo más adecuado en función de la carga y asigna el proceso a dicho nodo.
- **Precondición:** Los nodos están activos y conectados a la red distribuida.
- **Postcondición:** El proceso se asigna al nodo seleccionado y comienza su ejecución.
- **Flujo Normal:**
 1. El usuario solicita la ejecución de un proceso.
 2. El sistema verifica el estado de cada nodo y evalúa la carga de trabajo.
 3. El sistema asigna el proceso al nodo con menor carga.
 4. El nodo asignado ejecuta el proceso.
- **Excepciones:**
 - Si todos los nodos están sobrecargados, el proceso se coloca en una cola de espera hasta que un nodo esté disponible.
 - Si ocurre una falla en el nodo asignado antes de que el proceso inicie, el sistema reasigna el proceso a otro nodo disponible.

Caso de Uso 2: Sincronización de Recursos Compartidos

- **Actores:** Nodo
- **Descripción:** Un nodo solicita acceso a un recurso compartido. Si el recurso está disponible, el nodo lo adquiere; si no, espera hasta que esté disponible para asegurar la exclusión mutua.

- **Precondición:** Existen recursos compartidos en la red, y los nodos pueden solicitar acceso a dichos recursos.
- **Postcondición:** El recurso es asignado al nodo solicitante o el nodo entra en estado de espera hasta que el recurso esté disponible.
- **Flujo Normal:**
 1. Un nodo solicita acceso a un recurso compartido.
 2. El sistema verifica la disponibilidad del recurso solicitado.
 3. Si el recurso está disponible, el nodo adquiere el recurso y lo utiliza.
 4. Al finalizar, el nodo libera el recurso para que otros nodos puedan acceder a él.
- **Excepciones:**
 - Si el recurso no está disponible, el nodo espera hasta que sea liberado.
 - Si ocurre una desconexión mientras un nodo está esperando el recurso, el sistema elimina la solicitud de acceso del nodo desconectado.

Caso de Uso 3: Manejo de Fallo de Nodo

- **Actores:** Nodo
- **Descripción:** Un nodo detecta que otro nodo ha fallado. El sistema marca el nodo fallido, redistribuye los procesos que estaban en ejecución en dicho nodo y asegura que los procesos continúen en otros nodos.
- **Precondición:** Todos los nodos están activos y conectados a la red distribuida.
- **Postcondición:** Los procesos asignados al nodo fallido se reasignan y continúan ejecutándose en otros nodos disponibles.
- **Flujo Normal:**
 1. Un nodo detecta la inactividad o desconexión de otro nodo en la red.
 2. El sistema marca al nodo fallido como inactivo y lo remueve de la lista de nodos disponibles.
 3. El sistema identifica los procesos que estaban en ejecución en el nodo fallido.
 4. Los procesos se redistribuyen entre los nodos restantes de la red.
 5. Los nodos restantes comienzan la ejecución de los procesos reasignados.
- **Excepciones:**
 - Si no hay suficientes nodos disponibles para redistribuir todos los procesos, el sistema mantiene algunos procesos en espera hasta que se libere un nodo o se reconecte un nodo previamente fallido.
 - Si el nodo fallido se recupera rápidamente, el sistema puede reasignar procesos de nuevo al nodo recuperado.

Pruebas de Uso

Prueba 1: Asignación de Procesos y Balanceo de Carga

- **Objetivo:** Verificar que el sistema asigna los procesos al nodo adecuado y distribuye la carga entre los nodos.
- **Entradas:** Carga inicial en cada nodo; proceso nuevo que debe ser asignado.
- **Procedimiento:**
 1. Asignar procesos a los nodos hasta alcanzar un nivel de carga variable en cada nodo.
 2. Crear un nuevo proceso y solicitar su asignación.
 3. Verificar que el proceso se asigna al nodo menos cargado.
- **Resultados Esperados:** El proceso debe ser asignado al nodo que tenga menos carga en ese momento.
- **Resultados Obtenidos:** (Llenar tras la ejecución de la prueba)

Prueba 2: Sincronización de Recursos Compartidos

- **Objetivo:** Comprobar que los nodos acceden a los recursos compartidos de forma sincronizada.
- **Entradas:** Solicitudes concurrentes de acceso a un mismo recurso desde diferentes nodos.
- **Procedimiento:**
 1. Configurar dos o más nodos para que soliciten acceso al mismo recurso al mismo tiempo.
 2. Observar el manejo del recurso y verificar que solo un nodo accede al recurso a la vez.
 3. Liberar el recurso y observar si el próximo nodo en espera lo adquiere.
- **Resultados Esperados:** Solo un nodo debe tener acceso al recurso en un momento dado, y los demás deben esperar su turno.
- **Resultados Obtenidos:** (Llenar tras la ejecución de la prueba)

Prueba 3: Manejo de Fallos

- **Objetivo:** Verificar que el sistema redistribuye correctamente los procesos en caso de fallo de un nodo.
- **Entradas:** Nodo en ejecución, procesos asignados al nodo, fallo simulado.
- **Procedimiento:**
 1. Asignar varios procesos a un nodo específico.
 2. Simular el fallo de dicho nodo.
 3. Observar la redistribución de los procesos en los nodos restantes.
- **Resultados Esperados:** Los procesos deben redistribuirse y ejecutarse en los nodos restantes sin interrumpir el sistema.
- **Resultados Obtenidos:** (Llenar tras la ejecución de la prueba)

Prueba 4: Escalabilidad del Sistema

- **Objetivo:** Evaluar la capacidad del sistema para agregar nuevos nodos sin afectar su funcionamiento.
- **Entradas:** Estado inicial de la red distribuida, nuevos nodos a agregar.
- **Procedimiento:**
 1. Ejecutar el sistema con un conjunto de nodos iniciales.
 2. Agregar nuevos nodos a la red distribuida mientras los procesos están en ejecución.
 3. Verificar que el sistema integre los nuevos nodos sin interrupciones.
- **Resultados Esperados:** El sistema debe aceptar los nuevos nodos y redistribuir la carga de procesos sin interrupciones.
- **Resultados Obtenidos:** (Llenar tras la ejecución de la prueba)

Prueba 5: Redistribución Automática de Procesos

- **Objetivo:** Comprobar que el sistema redistribuye los procesos de manera automática si un nodo alcanza su máxima capacidad de carga.
- **Entradas:** Número de procesos y límite de capacidad de carga en los nodos.
- **Procedimiento:**
 1. Asignar múltiples procesos a los nodos hasta que uno de los nodos alcance su límite de carga.
 2. Observar si el sistema redistribuye los procesos excedentes a otros nodos disponibles.
- **Resultados Esperados:** El sistema debe redistribuir automáticamente los procesos al nodo más adecuado.
- **Resultados Obtenidos:** (Llenar tras la ejecución de la prueba)

Entregables

- **Código Fuente:** Código del emulador, documentado adecuadamente.
- **Pruebas y Resultados:** Documentación de pruebas en cada caso de uso y sus resultados.
- **Documentación de Diseño:** Explicación de la arquitectura, mecanismos de comunicación, sincronización, y gestión de fallos.

Rúbrica de Evaluación

Criterio	Excelente (5)	Bueno (3-4)	Insuficiente (1-2)
Implementación de Mecanismos Distribuidos	Implementación completa y eficiente de comunicación y sincronización entre nodos.	Comunicación y sincronización correctas, pero con algunas ineficiencias.	Falla en la implementación de comunicación y/o sincronización.
Eficiencia en la Sincronización de Recursos	Gestión sincronizada y sin conflictos de recursos compartidos entre nodos.	Sincronización de recursos compartidos funcional pero con leves conflictos o fallos.	Fallos significativos en la sincronización de recursos.
Tolerancia a Fallos	El sistema redistribuye procesos sin errores en caso de fallo.	El sistema redistribuye procesos, pero con errores menores o retrasos.	El sistema falla al redistribuir procesos en caso de fallo.
Documentación y Claridad	Documentación completa y clara, detallando arquitectura, diseño y rendimiento.	Documentación adecuada, con detalles básicos de diseño y rendimiento.	Documentación incompleta o confusa, faltan detalles importantes.