

Roteiro de Estudo de Sistemas Distribuídos e de Tempo Real

O que é um processo?

Um processo é um programa em execução. Um processo é um consumidor dos recursos e serviços disponibilizados por um sistema operacional. Do ponto de vista do sistema operacional ele é identificado por um número, conhecido PID.

Como ele usa o processador?

Um sistema operacional faz o revezamento de processos para usar o processador. Geralmente o tempo disponível de processador é dividido em

Quais são estados de um processo?

- Rodando: enquanto está de fato usando o processador.
- Pronto: enquanto aguarda na fila para usar o processador.
- Bloqueado: enquanto aguarda a liberação de algum recurso ou término de alguma tarefa.

Como um processo usa a memória?

Durante a criação de um processo, as informações sobre o processo são colocadas na área de memória do sistema operacional para que ele possa acompanhar a vida do processo.

É delimitada outra área de memória para o próprio processo para a qual o seu código compilado que está no disco é copiado.

Além disso, o processo pode alocar mais memória para armazenar temporariamente os seus dados.

Como um processo lida com arquivos e se comunica em rede?

Um processo lida com arquivos e comunicação em rede através de chamadas de sistema do sistema operacional. Chamadas de sistema são funções específicas do sistema operacional, que ao serem invocadas fornecem recursos ou serviços.

Por exemplo, chamadas de sistema como `open()`, `write()`, `close()`, `seek()`, `getwd()`, `mkdir()` podem ser incluídas no código para que um processo manipule arquivos e diretórios.

No caso da comunicação em rede existe um conjunto de chamadas de sistema e regras de utilização, conhecido como API de soquetes. Como exemplos de chamadas de sistemas de rede, podemos citar `socket()`, `bind()`, `listen()`, `send()`, `recv()`

Quando criamos um socket, podemos escolher entre dois tipos de serviço de transporte de dados na rede, o TCP e o UDP. O TCP presta o serviço com garantia de entrega, o UDP presta o serviço de melhor esforço.

Qual a definição de Sistema Distribuído?

Um sistema distribuído é um sistema no qual os componentes se comunicam de forma coordenada para executar uma tarefa. Esses componentes podem ser processadores, computadores, processos, ou objetos. Eles podem estar interligados através de barramentos, redes ou middlewares. Eles trocam mensagens pedindo que eles executem determinada ação ou aguardem um certo evento.

O que é transparência?

Transparência é um conceito central em sistemas distribuídos. Significa que o usuário de um sistema distribuído deve experimentar a sensação de estar utilizando um único sistema computacional e sua operação deve parecer o mais simples possível. Trata-se de esconder a complexidade dos detalhes internos de um sistema distribuído do usuário.

O que disponibilidade?

Um sistema distribuído deve estar a disposição quando o usuário precisar de seus serviços. Diversos imprevistos podem ocorrer internamente em um sistema distribuído, como mal funcionamento do hardware, ausência de energia, problemas na rede de comunicação. Um projeto de sistema distribuído deve adotar medidas para contornar ou amenizar os efeitos desses imprevistos nos usuários.

O que é escalabilidade?

Sistemas distribuídos devem ser capazes de atender uma quantidade crescente de usuários sem deixar de estar disponível, nem deixar a qualidade do serviço ficar abaixo de certos parâmetros. Um projeto de sistema distribuído deve adotar medidas para manter um serviço de qualidade, bem como consumir o mínimo de recursos na ausência de usuários.

O que é Middleware?

Middleware é o software responsável pela interconexão dos diversos componente do sistema distribuído. Um middleware pode executar tarefas como a invocação de procedimentos ou métodos remotos, a distribuição, roteamento e armazenamento de mensagens. Lembrando que os componentes de um SD geralmente incluem diversos sistemas computacionais, logo o middleware serve como uma cola que segura as diferentes partes.

Qual a definição de Sistema de Tempo Real?

Um sistema em que a exatidão depende não apenas da correção lógica dos cálculos, mas também do instante em que os resultados são produzidos. Para um sistema de tempo real, entregar respostas fora do prazo é o mesmo que entregar respostas incorretas.

Os sistemas de tempo real podem ser classificados em:

- **Hard Real-Time:** prazo rígido, atraso = falha (ex.: sistemas de controle de usinas nucleares).
- **Soft Real-Time:** atrasos tolerados em pequena escala (ex.: streaming).
- **Firm Real-Time:** atrasos ocasionais tornam o resultado inútil, mas não catastrófico (ex.: sensores meteorológicos).

Quais as técnicas utilizadas na construção de sistemas de tempo real?

- Hardware especializado
- Escalonamento de Processos com o algoritmo adequado para o cenário
- Uso equilibrado de Interrupções
- Boas práticas no gerenciamento de recursos compartilhado
- Particionamento Temporal
- Controle de Latência

Com relação a evolução dos sistemas distribuídos, podemos citar:

- Criação do computador

- Maturação dos computadores centralizados com os mainframes
- Maturação do software com o surgimento das primeiras linguagens de programação
- Surgimento da programação orientada a objetos e o nascimento da Internet
- Surgimento dos computadores pessoais e dos primeiros serviços da Internet
- Evolução dos sistemas distribuídos como corporativos, de grade, e finalmente nuvem.

Sistemas Centralizados x Sistemas Distribuídos

Aspecto	Centralizado	Distribuído
Arquitetura	Um servidor central controla tudo	Vários nós autônomos interconectados
Confiabilidade	Baixa (ponto único de falha)	Alta (tolerância a falhas)
Escalabilidade	Limitada	Elevada
Gerenciamento	Simple	Complexo
Desempenho	Limitado ao servidor central	Pode aproveitar paralelismo
Segurança	Mais fácil de aplicar	Mais difícil, devido à dispersão
Consistência	Natural, já que há um único centro	Difícil de manter entre vários nós