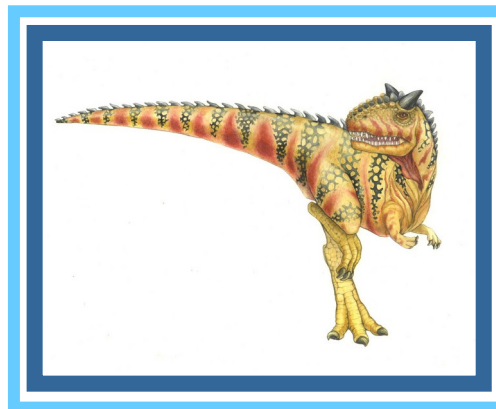
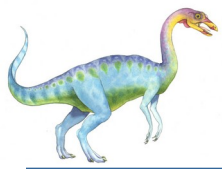


Questionário





Questionário

- 1) Quando um processo cria um novo processo usando a operação `fork()`, qual dos estados a seguir é compartilhado entre o processo pai e o processo filho?
- a. Pilha
 - b. Heap
 - c. Segmentos de memória compartilhada

Resposta:





Questionário

- 1) Quando um processo cria um novo processo usando a operação `fork()`, qual dos estados a seguir é compartilhado entre o processo pai e o processo filho?
- a. Pilha
 - b. Heap
 - c. Segmentos de memória compartilhada

Resposta:

Somente os segmentos de memória compartilhada são compartilhados entre o processo pai e o processo filho recém-criado por `fork`. São feitas cópias da pilha e do heap para o processo recém-criado.





Questionário

2) No que diz respeito ao mecanismo RPC (Chamada de Procedimento Remota), considere a semântica “exatamente um”. O algoritmo de implementação dessa semântica é executado corretamente mesmo quando a mensagem ACK retornada ao cliente é perdida devido a um problema na rede? Descreva a sequência de mensagens e discuta se a semântica “exatamente um” continua sendo preservada.

Resposta:





Questionário

2) No que diz respeito ao mecanismo RPC (Chamada de Procedimento Remota), considere a semântica “exatamente um”. algoritmo de implementação dessa semântica é executado corretamente mesmo quando a mensagem ACK retornada ao cliente é perdida devido a um problema na rede? Descreva a sequência de mensagens e discuta se a semântica “exatamente um” continua sendo preservada.

Resposta:

A semântica “exatamente um” garante que um procedimento remoto seja executado exatamente uma vez e somente uma vez. O algoritmo geral para fornecer essa garantia combina um esquema de conhecimento (ACK – acknowledgment) combinado com marcadores de tempo (ou algum outro contador incremental que permita ao servidor distinguir entre mensagens duplicadas).

A estratégia geral é que o cliente envie uma RPC ao servidor junto com um marcador de tempo. O cliente também dará início a um relógio de expiração de tempo. O cliente então esperará por uma de duas ocorrências:

(1) ele receberá um ACK do servidor indicando que o procedimento remoto foi executado ou

(2) ele expirará por tempo. Se o cliente expirar por tempo, assumirá que o servidor foi incapaz de executar o procedimento remoto, e, assim, o cliente invocará a RPC uma segunda vez, enviando um último marcador de tempo.

O cliente pode não receber o ACK por uma de duas razões:

(1) a RPC original jamais foi recebida pelo servidor ou

(2) a RPC foi corretamente recebida – e executada pelo servidor –, mas o ACK foi perdido.

Na situação (1), o uso de ACKs permitirá que o servidor finalmente receba e execute a RPC. Na situação (2), o servidor receberá uma RPC em duplicata e utilizará o marcador de tempo para identificá-la como uma duplicata e não executar a RPC uma segunda vez. É importante observar que o servidor deverá enviar um segundo ACK de volta ao cliente para informá-lo de que a RPC foi executada.





Questionário

- 3) Forneça dois exemplos de programação em que a criação de vários threads proporcione melhor desempenho do que uma solução com um único thread.





Questionário

3) Forneça dois exemplos de programação em que a criação de vários threads proporcione melhor desempenho do que uma solução com um único thread.

Resposta:

(1) Um servidor Web que sirva cada solicitação em um thread separado.

(2) Uma aplicação paralelizada, tal como uma multiplicação de matrizes, em que diferentes partes da matriz podem ser trabalhadas em paralelo.

(3) Um programa de GUI interativo, tal como um depurador, em que são

utilizados um thread para monitorar a entrada do usuário, outro thread para representar a aplicação em execução e um terceiro thread para monitorar o desempenho.





Questionário

4) Que recursos são usados quando um thread é criado? Em que eles diferem dos usados quando um processo é criado?

Resposta:





Questionário

4) Que recursos são usados quando um thread é criado? Em que eles diferem dos usados quando um processo é criado?

Resposta:

Como um thread é menor do que um processo, a criação do thread normalmente utiliza menos recursos do que a criação do processo. A criação de um processo requer a alocação de um bloco de controle de processo (PCB) que é uma estrutura de dados um tanto grande. O PCB inclui um mapa da memória, uma lista de arquivos abertos e variáveis ambientais. A alocação e o gerenciamento do mapa da memória são normalmente a atividade de maior consumo de tempo. A criação tanto de um thread de usuário quanto de um thread de kernel envolve a alocação de uma pequena estrutura de dados para manter o estado dos registradores, a pilha e as prioridades.





Questionário

5)Qual a vantagem de termos tamanhos diferentes para o quantum de tempo em níveis distintos de um sistema de enfileiramento em vários níveis?

Resposta:





Questionário

5)Qual a vantagem de termos tamanhos diferentes para o quantum de tempo em níveis distintos de um sistema de enfileiramento em vários níveis?

Resposta:

Os processos que precisam de serviço mais frequente, como por exemplo os processos interativos, tais como os editores, podem estar em uma fila com um quantum de tempo pequeno. Os processos que não precisam de serviço frequente podem estar em uma fila com um quantum de tempo maior, requerendo menos mudanças de contexto para completar o processamento e fazendo, assim, uso mais eficiente do computador.

