FIAP

Aluno: Breno de Souza Silva

RM: 88332

Checkpoint 1 – Operating System Tuning And Cognation – Professor Sérgio Ricardo Rota

1 – Sistema Operacional Genérico (3,0 pontos)

Um médico veterinário, proprietário de um pet shop, obteve com um amigo o código executável (já compilado) de uma aplicação para automação de pet shops. Segundo seu amigo, a aplicação em questão havia sido desenvolvida e compilada especificamente para seu computador pessoal (sistema A), no qual funciona corretamente. Para sua surpresa, o médico veterinário constatou que a aplicação também funciona corretamente em seu computador pessoal (sistema B).

Considerando a situação descrita acima, analise as afirmações a seguir, indicando se cada afirmação é VERDADEIRA ou FALSA, justificando sua decisão em cada uma delas:

(a) Os computadores pessoais poderiam ter quantidades diferentes de núcleos de processamento (cores).

**Verdadeiro - Sim, os computadores pessoais podem ter quantidades diferentes de núcleos de processamento (cores). Atualmente, é comum encontrar processadores de dois, quatro, seis ou até mais núcleos em computadores pessoais, dependendo do modelo e das especificações do sistema. A vantagem deste tipo de sistema reside no fato de permitir que vários programas sejam executados ao mesmo tempo ou que um único programa seja subdividido em várias partes para serem executadas simultaneamente em mais de um processador. O número de núcleos em um processador determina quantas tarefas ele pode executar simultaneamente. Um processador com mais núcleos pode executar mais tarefas ao mesmo tempo do que um processador com menos núcleos.**

(b) As chamadas de sistema (system call) do sistema operacional do sistema A devem ser compatíveis com as do sistema B.

**Verdadeiro - Em resumo, a compatibilidade das chamadas de sistema entre diferentes sistemas operacionais depende da implementação específica de cada sistema operacional e das chamadas de sistema que eles oferecem, ou seja, como no enunciado diz que, o médico veterinário conseguiu rodar o programa no sistema B, podemos entender que o sistema operacional A tem uma implementação especifica compatível, pois o programa foi executado normalmente.**

(c) O conjunto de instruções do sistema A poderia ser diferente do conjunto de instruções do sistema B.

Falsa, pois ao compilar o programa, o que se obtêm é justamente um conjunto de instruções a partir do sistema A para que o processo possa ser executado, e tendo em vista que o mesmo código compilado foi passado para o sistema B, não teria como o conjunto de instruções ser diferente.

(d) Se os registradores do sistema A forem de 64 bits, os registradores do sistema B poderiam ser de 32 bits.

**Falsa - Os registradores do sistema A e B devem ter compatibilidade na largura de bits, para que a aplicação possa ser executada em ambos os sistemas. Se os registradores do sistema A forem de 64 bits, então os registradores do sistema B também precisam ser de 64 bits ou mais, para que a aplicação possa funcionar corretamente. Se o sistema B tiver registradores de apenas 32 bits, pode haver problemas de compatibilidade e a aplicação pode não funcionar adequadamente. Portanto, é importante que os registradores dos sistemas sejam compatíveis para garantir a execução correta da aplicação em ambos os sistemas.**

(e) Mesmo considerando-se sistemas operacionais diferentes, tal situação pode ocorrer porque a forma como o código do sistema operacional está organizada e o interrelacionamento entre seus diversos subsistemas é praticamente o mesmo para os diversos tipos de sistemas operacionais.

**Falsa - Embora possa haver semelhanças em alguns aspectos dos diferentes sistemas operacionais, como a organização do código e a interação entre subsistemas, não é possível garantir que uma aplicação compilada em um sistema operacional funcionará em outro sistema operacional, mesmo que os sistemas pareçam semelhantes. A compatibilidade entre diferentes sistemas operacionais depende de muitos fatores, como a arquitetura do processador, as bibliotecas compartilhadas e as chamadas de sistema. Portanto, é importante levar em consideração as diferenças entre os sistemas operacionais e testar a aplicação em cada sistema para garantir sua compatibilidade e funcionamento correto.**

2 – Multiprogramação com Eficiência (3,0 pontos)

Uma empresa está projetando um novo modelo de computador. Na atual fase do projeto, estão sendo analisadas algumas propostas para diminuir a sobrecarga associada à troca de contexto:

(a) Aumentar a quantidade de memória RAM, isto é, o espaço de endereçamento disponível para os processos.

**A solução de aumentar a quantidade de memória RAM não resolve diretamente a sobrecarga da troca de contexto, pois é necessário salvar e restaurar o contexto de hardware dos processos, como registradores e outros componentes, independentemente da quantidade de memória disponível. Embora haja possíveis benefícios indiretos para o desempenho geral do sistema, essa proposta não é uma boa solução para o problema específico da sobrecarga de troca de contexto.**

(b) Reservar uma área na memória cache interna ao processador exclusivamente para armazenar o contexto de hardware dos processos.

**Alocar uma área na memória cache interna ao processador para armazenar o contexto de hardware dos processos, na nossa opinião pode ser uma boa proposta interessante para reduzir a sobrecarga de troca de contexto, pois a memória cache é muito mais rápida do que a memória principal. Isso pode reduzir o tempo necessário para salvar e restaurar o contexto do processo, melhorando o desempenho geral do sistema.**

(c) Implementar múltiplos conjuntos de registradores, de tal forma que uma troca de contexto simplesmente significaria a modificação do ponteiro para o conjunto corrente de registradores.

**A implementação de múltiplos conjuntos de registradores pode ser uma solução viável para diminuir a sobrecarga associada à troca de contexto, uma vez que é possível atribuir um conjunto de registradores exclusivo para cada processo, eliminando assim a necessidade de salvar e restaurar os valores dos registradores durante a troca de contexto e tornando a operação mais rápida.**

3 – Análise de Testes de Aplicações (4,0 pontos)

Imagine que você tenha sido contratado como analista de testes de um site especializado em tecnologia. Sua primeira tarefa é realizar um benchmarking – processo de comparação de produtos, serviços e práticas corporativas – de 6 aplicações geradoras de HASH: App1, App2, App3, App4, App5 e App6.

O ambiente de teste é um servidor equipado com um processador de um núcleo (single core) que serializa os acessos ao disco rígido. Além do sistema operacional multitarefa e dos processos do sistema, este servidor executa apenas as aplicações que você solicitar.

Durante a avaliação, você percebeu que algumas aplicações passam, praticamente, 100% do tempo de execução realizando operações lógicas e aritméticas (denominadas aplicações CPU-bound) e outras aplicações passam cerca de 70% do tempo de execução realizando acesso ao disco rígido (denominadas aplicações I/O-bound) e 30% do tempo de execução realizando operações lógicas e aritméticas.

Como parte da avaliação, inicialmente, você executou as aplicações isoladamente e mediu o tempo de execução de cada uma delas (Tabela I). Em seguida, você combinou as aplicações em grupos e as executou simultaneamente (Tabela II).

Tabela I

|  |  |  |
| --- | --- | --- |
| aplicação | tipo | tempo de execução [ms] |
| App1 | CPU-bound | 20 |
| App2 | CPU-bound | 40 |
| App3 | CPU-bound | 60 |
| App4 | I/O-bound | 50 |
| App5 | I/O-bound | 100 |
| App6 | I/O-bound | 150 |

Tabela II

|  |  |
| --- | --- |
| GRUPO | tempo de execução [ms] |
| 1: App1 + App2 + App3 | 123 |
| 2: App4 + App5 + App6 | 210 |
| 3: App3 + App5 + App6 | 160 |
| 4: TODAS | 306 |

Alguns dias depois, quando você foi elaborar o relatório final da análise, revendo as tabelas, você ficou com dúvida em relação aos tempos anotados.

Com base nos dados das Tabela I e II, avalie as afirmações a seguir, indicando se cada afirmação é VERDADEIRA ou FALSA, justificando sua decisão em cada uma delas:

(a) É possível que o tempo total do GRUPO 4 seja 306 ms (milissegundo, 1×10−3 s).

VERDADEIRA: O tempo mínimo de execução do GRUPO 4 é de 220 ms enquanto o máximo ser8ia 370 ms.

(b) O tempo total do GRUPO 1 deveria ser, no mínimo, de 100 ms.

FALSA. A soma dos tempos de execução individual das aplicações do GRUPO 1 é de 120ms (20ms + 40ms + 60ms), portanto, o tempo total do GRUPO 1 deveria ser de, no mínimo, 120ms e não 100ms.

(c) O tempo total do GRUPO 2 deveria ser, no mínimo, de 300 ms.

FALSA. Considerando os três cenários abaixo:

O tempo total do GRUPO 2 deveria ser, no mínimo, de 225 ms.

Uma imagem contendo Gráfico

Descrição gerada automaticamente

(d) O tempo total do GRUPO 3 deveria estar acima de 175 ms e abaixo de 310 ms.

VERDADEIRA. Considerando os três cenários abaixo:

Cenário 1 (Executando na seguinte ordem: App6 + App 5 + App3):

**Carta

Descrição gerada automaticamente com confiança baixa**

**Cenário 2** (Executando na seguinte ordem: App5 + App 6 + App3)**:Uma imagem contendo Texto

Descrição gerada automaticamente**

**Cenário 3** (Executando na seguinte ordem: App3 + App6 + App5)**:**

**Carta

Descrição gerada automaticamente com confiança baixa**

Podemos entender que o tempo máximo de execução do GRUPO 3 é de 280ms, enquanto o mínimo de execução do GRUPO 3 possivelmente poderia ser de 205ms.