



Campus Campina Grande

Disciplina: Sistemas Operacionais

Data: 06 de Agosto de 2021

Professor: David Candeia Medeiros Maia

Aluno: Luiz Medeiros Neto

Respostas da prova sobre Histórico Conceitos de S.O, Processos e Threads.

1-

Mediante ao contexto de Sistema Operacional e suas aplicações, é possível evidenciar que um S.O é um conjunto de softwares que permitem administrar recursos de um Sistema Computacional. Em suas funções, ressalta-se o papel de fazer a abstração do hardware, alocar ordenadamente e controladamente processadores, memórias e dispositivos de entrada e saída. Ainda é possível falar sobre o Kernel, que é o coração do Sistema Operacional, seu papel principal é fazer a comunicação de hardware com o software de forma altamente segura sem quebrar protocolos ou lógicas de gerenciamento, com o auxílio dos leitores de hardware e interpretadores de comandos para interpretar os comandos inseridos pelos usuários.

2-

Basicamente, a diferença entre os dois modos de operação, está na característica de que em modo usuário o acesso a recursos do sistema e ao hardware fica limitado, esse modo é característico dos softwares instalados, onde eles podem enviar informações de saída e entrada ou solicitar algum recurso que será analisado a permissão por exemplo. Em modo Kernel(núcleo) não há limitação do que pode ser feito na máquina; tendo em vista que é possível trabalhar a nível de hardware modificando como cada instrução é armazenada ou endereçada, ou até mesmo como a CPU e a ULA são controlados e alocados de acordo com a atividade de processos, haja vista que quem deve fazer isso é o S.O e por isso ele possui acesso não restrito aos recursos.

Manter dois modos distintos ajuda no gerenciamento de recursos do Sistema Computacional, uma vez que se tivéssemos apenas modo Kernel o uso do computador seria

inviável por questões de segurança do Kernel e do usuário. Além do fato de ser possível o uso errôneo de recursos por parte do usuário, modificando toda a lógica de trabalho do S.O no endereçamento, alocação e desalocação de memória; permitir reiniciar ou desligar o sistema a qualquer instante, etc. É possível, também, desorganizar a forma com que se utilizaria os dados de leitura dos interpretadores de comando ou as lógicas de comunicação interprocessos. Em modo usuário o acesso a recursos é restrito, logo, é mais difícil fazer chamadas que danifiquem ou atrapalhem o funcionamento do S.O ou de outros softwares em execução, ou que até mesmo façam invasões à privacidade do usuário.

3-

Chamadas ao sistemas, são requisições que softwares instalados ou programas elaborados pelo usuário, por exemplo, podem fazer ao Sistema Operacional. Alguns comandos POSIX são:

- shutdown -h now / systemctl poweroff: desliga a máquina.
- fork(): cria um processo idêntico ao que o chamou e retorna o PID do processo criado.
- comando kill(): usado para encerrar um processo, por exemplo.

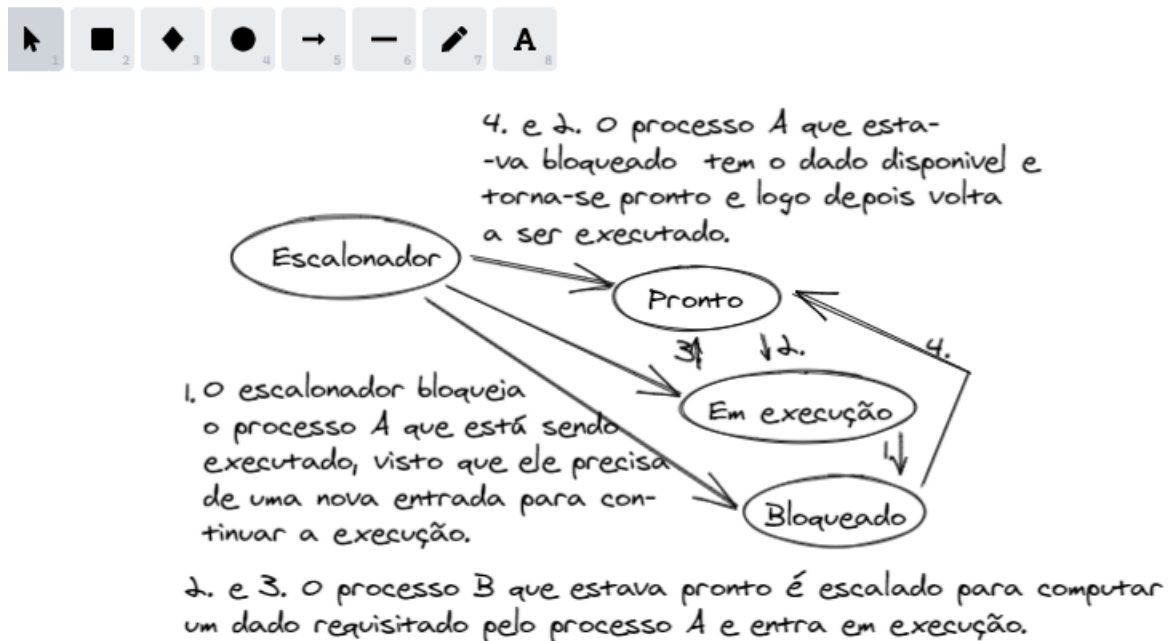
4-

Os processos possuem 3 estados, o estado indica o status de um processo naquele momento, ou seja, qual a situação dele ou o que ele está fazendo em determinado intervalo de tempo. São os estados de um processo:

- Em execução: o processo está requisitando recursos de memória ao Sistema Operacional, fazendo operações matemáticas, inserindo dados de saída, computando dados etc, ou seja, está usando a unidade central de processamento(CPU) em determinado instante ou intervalo de tempo.
- pronto: o processo está pronto para ser executado, porém está temporariamente parado para dar lugar a outro processo que está alocando os recursos do sistema.
- bloqueado: incapaz de ser executado enquanto algum fator externo relacionado não ocorrer, como por exemplo a obtenção de determinado dado de entrada ou algum dado computado por outro processo, tratamentos de erros realizados, chamadas do usuário, sinalizações externas de arquivos, etc.

É importante frisar, que o escalonador é o órgão responsável pela gerência dos processos com o auxílio da tabela de processos; o escalonador coordena a CPU, determina o tempo de execução de um processo e o processo que será executado em sequência por meio de sinalizações de entrada externas ou arquivos.

Figura 1.



5-

Processos são programas em execução, que realizam alguma ação dentro do Sistema Computacional e que podem ser controlados de alguma forma, seja pelo usuário, por softwares externos ou pelo Kernel. Vale lembrar, que um processo ocupa espaço de memória para criar variáveis globais, organizar instruções e possuem estados, por exemplo.

Thread é um fluxo de execução sequencial dentro de um processo, normalmente um processo possui uma thread nomeada thread de execução, podendo também, ter mais de uma thread que podem cooperar ou não. Em convergência com os processos, uma thread possui estados, variáveis, etc.

6-

Nas vantagens de uso de aplicações multithreads, incluem-se:

- Paralelismo dentro de um mesmo processo, ou seja, várias atividades sendo executadas ou não ao mesmo tempo;
- Uma vez que a utilização de vários processos for inviável, a aplicação de multithread pode ser útil;
- Compartilhamento entre threads de recursos já alocados na memória;
- Criar e destruir threads é mais fácil;
- Ganhos em desempenho de execução do processo;

Como um exemplo do uso de threads pode ser citado um programa de navegador, que ao mesmo tempo em que processa dados de entrada recebidos, pode formatá-los buscando organizá-los com o objetivo de impressão ou envio, e ainda procurar por arquivos ou dados em outros servidores ou dispositivos da rede.

7-

a-) Falsa, Justificativa válida:

- Trecho: "possuía milhares de linhas de código".
- Correção: possuía milhões de linhas de código.

b-) Falsa: Justificativa: O espaço de endereçamento de um processo, é o local onde ele aloca suas variáveis, sua pilha, texto e descritores de E/S por exemplo.

c-) Verdadeiro, entretanto, do jeito em que se encontra, a sentença diz mais respeito ao conceito de multiprocessos, portanto uma sentença mais completa seria: time-sharing se refere ao tempo compartilhado por processos numa fila, ou seja, o tempo determinado pelo escalonador para coordenar vários processos, que permite a alternância de processos, isso se deve ao fato de termos diferentes processos numa fila de processos, que ocupam simultaneamente distintos endereços de memória num sistema computacional.

d-) Verdadeiro.

e-) Falso. Justificativa: Os processos são mais seguros que as threads , uma vez que são executados no seu próprio endereço de memória. Threads é mais perigoso, visto que uma thread pode fazer qualquer coisa dentro de um espaço de endereçamento, podendo atualizar valores de variáveis, apagar a pilha de memória de outra thread, etc.