Nome: Daniel Taiki Ukita ,Gustavo Veronica Santos e Helena Carneiro dos Santos

Curso: Eng. De Computação

Disciplina: Sistemas Operacionais **Semestre:** 4°

Exercícios

1 - Explique o que está associado a cada processo que informa onde o processo pode ler e escrever e a relação coma memória.

R: A forma para controlar o acesso à memória, geralmente são utilizados mecanismos de gerenciamento de memória que definem as permissões de leitura e escrita para cada processo. Essas permissões são associadas a diferentes partes da memória e ajudam a garantir que os processos não acessem ou modifiquem inadvertidamente a memória uns dos outros.

2 - Explique por que o processo pode ser considerado como um contêiner que armazena informações para um programa.

R: O processo pode ser considerado um contêiner que armazena todas as informações necessárias para executar um programa de maneira isolada e segura, garantindo que cada programa tenha seu próprio ambiente de execução e recursos sem interferir nos outros. Esse conceito de encapsulamento e isolamento é fundamental para a segurança e estabilidade dos sistemas operacionais.

3 – Qual a função do espaço de endereçamento?

R: Sua função principal é permitir que o sistema operacional gerencie e organize a memória de forma eficiente, fornecendo um método para identificar e acessar dados armazenados na memória do sistema. Dentre outras funções como, desempenhar um papel fundamental na organização, acesso, isolamento e gerenciamento da memória em um sistema de computador.

4 - Explique a função do ponteiro do ponteiro de pilha.

R: O ponteiro de pilha é um registrador utilizado para gerenciar a pilha de execução em programas. A função do ponteiro de pilha é essencial no contexto de execução de programas e sub-rotinas. De modo que mantém o controle do topo da pilha de execução. A pilha é uma estrutura de dados em que os dados são empilhados e desempilhados em ordem last-in, first-out (LIFO). Isso significa que o último item empilhado é o primeiro a ser removido.

5 - Quais são as principais chamadas de sistema de gerenciamento de processos ?

R: As chamadas de sistema relacionadas ao gerenciamento de processos são funções que um programa pode chamar para interagir com o sistema operacional e solicitar a criação, controle e manipulação de processos. Essas chamadas de sistema são essenciais para a execução de programas e a coordenação de tarefas em um sistema operacional. Dentre os comandos mais comuns estão:

- fork(): Essa chamada de sistema é usada para criar um novo processo (processo filho) a partir de um processo existente (processo pai).

- exec(): As chamadas de sistema da família exec (por exemplo, execve(), execl(), execvp()) são usadas para substituir o código do processo atual por um novo programa.

- wait() e waitpid(): Essas chamadas de sistema são usadas para esperar que um processo filho termine a execução antes que o processo pai prossiga.

- exit(): A chamada de sistema exit() é usada para terminar a execução de um processo. Ela pode retornar um código de status que indica se o processo terminou com sucesso ou com erro.

- getpid(): Essa chamada de sistema retorna o ID do processo (PID) do processo atual. É útil para identificar processos e interagir com eles.

- kill(): A chamada de sistema kill() é usada para enviar sinais a processos específicos. Os sinais podem ser usados para controlar ou interromper a execução de um processo.

- nice() e renice(): Essas chamadas de sistema permitem ajustar a prioridade de execução de um processo

- sched_yield(): A chamada de sistema sched_yield() permite que um processo voluntariamente desista de seu tempo de CPU para dar a outros processos a oportunidade de executar.

- chdir(): Usada para alterar o diretório de trabalho atual de um processo.

- umask(): A chamada de sistema umask() define as permissões padrão para a criação de arquivos por um processo.