

1º. Projeto de pesquisa

1. Título: Gerenciamento de memória em sistemas operacionais

2. Objetivos do projeto e descrição Geral

Suponha um sistema operacional, no qual os processos (programas em execução) ocupam a memória principal conforme descrito no exemplo da Figura 01 a seguir:

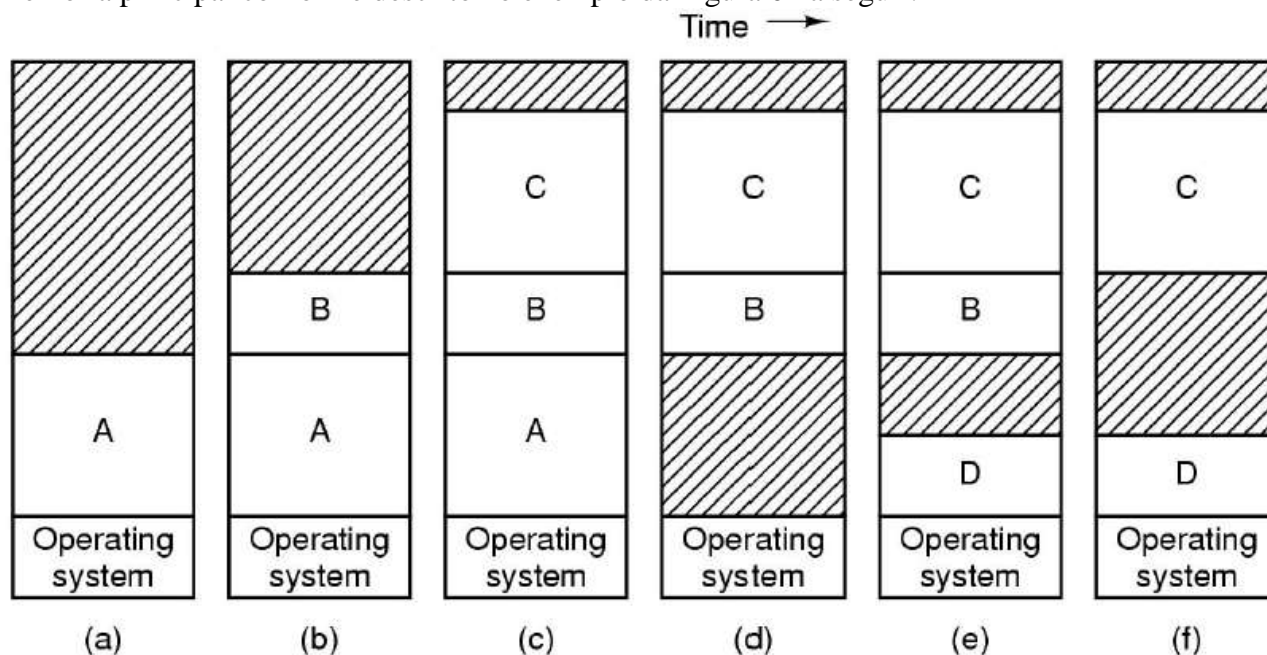


Figura 01 – Alocação de processos na memória principal

Nesse exemplo, inicialmente apenas o processo A está na memória. Em seguida, os processos B e C são criados e ocupam a memória. Logo depois, o processo A termina e, em seguida entra o processo D na memória e, um pouco depois, o processo B termina. Percebe-se que, nesse esquema, processos criados ocupam a memória e, ao encerrarem, deixam espaços, que podem ser ocupados por outros processos.

O objetivo do projeto é que o aluno construa um programa que simule a ocupação de memória por processos conforme os requisitos listados a seguir.

3. Dicas/Requisitos

- Os processos, quando iniciados, devem ter como parâmetros o seu tamanho (em Kbytes) e o tempo de execução.
- Processos devem ser alocados na memória de acordo com o seu tamanho e devem ter um label identificador (por exemplo, o nome A ou B como está apresentado na Figura 01).
- O processo de alocação de memória para um processo deve seguir a seguinte regra: o processo deve ser colocado na primeira região de memória (a contar do início para o final da memória) onde esse processo couber.

- Na hipótese de um processo não caber nos buracos de memória disponíveis, o programa deve tentar encontrar uma forma de reorganização dos processos na memória de modo a acomodar o novo processo. Se depois dessa tentativa não houver espaço disponível, deve-se imprimir uma mensagem de que não foi possível alocar memória para aquele processo.
- O tempo de execução de um processo é aleatório (não previsível). Portanto, caberá ao aluno definir como ele vai simular esse tempo de execução e o término de um processo. Sugere-se ainda que o programa imprima o registro de entrada e saída de processos na memória.
- O programa deve ser capaz de imprimir, a qualquer momento, quais posições de memória estão ocupadas (P) e quais estão livres (H). Imagine que a memória é um grande vetor onde cada posição armazena um Kbyte.
- Além das opções para inclusão de processos na memória e impressão, o programa deve ter uma opção de sair (deixar o programa). Nesse caso, o programa deve oferecer a opção de gravar os dados em arquivo. Da mesma forma, quando o programa for iniciado, deve ser possível ler as informações desse arquivo para continuar a execução da simulação da memória.
- A gerência de espaços de processo (P) e buracos (H) na memória deve ser feito por meio de uma lista circular duplamente encadeada, com cabeçalho. Nesse caso, o aluno deve desenhar o nó considerando pelo menos: tipo de nó (P ou H), posição de memória inicial, tamanho da área de memória livre ou ocupada.
- No processo de entrada e saída de processos na memória, as seguintes possibilidades podem acontecer com os nós da lista circular:

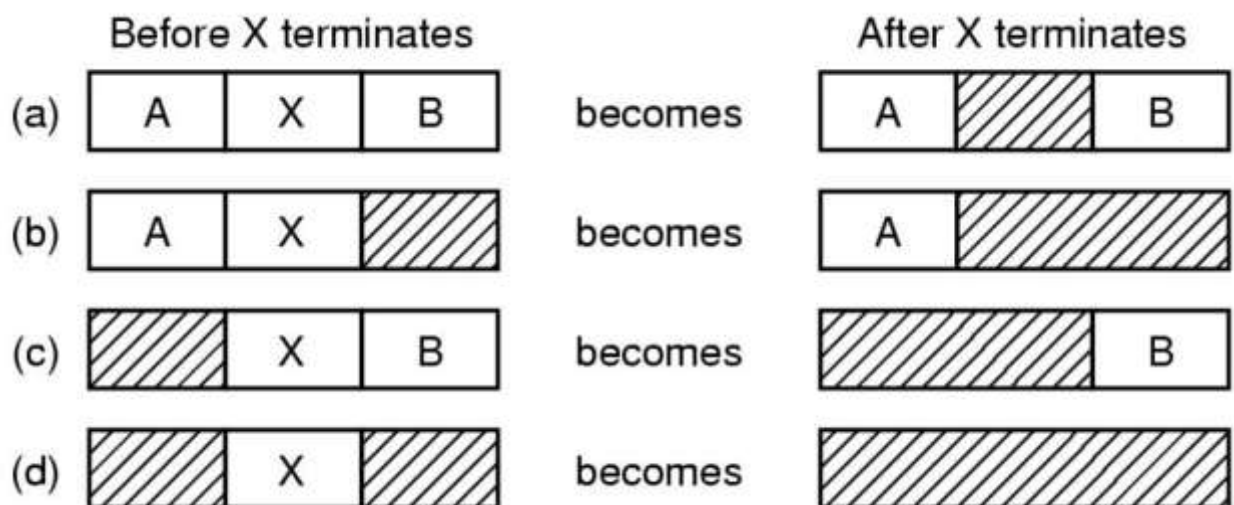


Figura 02 – Situações possíveis para um processo X em relação aos seus vizinhos

Nessa figura, a letra (a) indica a situação do processo X antes de terminar e após terminar considerando que os dois vizinhos mais próximos da lista são processos. Nas letras (b) e (c), considera-se uma situação de X ao terminar, tendo como vizinhos um nó de processo e um buraco. Na letra (d), percebe-se que X ao terminar, sendo vizinho de dois buracos, fará com que a lista passe a ter um único nó do tipo buraco com tamanho igual a soma dos buracos anteriores mais o tamanho do processo que finalizou.

- Caberá ao aluno definir estratégias para inserir processos na memória e realocar processos na memória.

4. Requisitos e regras para o trabalho

Este trabalho deve ser feito obedecendo as seguintes regras:

- O trabalho pode ser feito por grupos de até 3 alunos.
- O grupo deve apresentar a aplicação funcionando no laboratório (e entregá-la em mídia digital junto com relatório via Moodle).
- O grupo de alunos receberá notas relativas à implementação de código, além da nota de relatório e apresentação oral.
- Cada aluno do grupo deve entregar um relatório INDIVIDUAL, contendo:
 - i) Os objetivos do trabalho (incluindo uma declaração sobre quais requisitos de implementação citados foram resolvidos e se foram resolvidos integralmente).
 - ii) Um diário de atividades, relatando o que foi resolvido ou estudado em que dia até chegar a versão final do trabalho. Nesse diário, relatar uma descrição resumida sobre os problemas encontrados e as soluções adotadas ao longo dos estudos.
 - iii) Todo o código deve estar identado e devidamente documentado, inclusive com uma descrição das funções e dos parâmetros usados.
 - iv) O relatório deve prover uma análise do grau de falhas e limitação da solução apresentada e as melhorias que poderiam ser providas, se for o caso.
 - v) Sua opinião pessoal sobre o projeto.
- A nota é individual e o valor máximo será dado ao aluno que demonstrar conhecimento e aprendizado satisfatório com o projeto. Outros requisitos como cumprimento das datas, trabalho coordenado em grupo e inserção de novas funcionalidades no projeto também serão consideradas para emissão da nota final.
- Datas de entrega: (i) relatório escrito e códigos – devem ser postados no Moodle até às 23h do dia 13/10/2017, (ii) apresentação oral – em 17/10/2017. No dia, haverá sorteio para as apresentações. Alunos que não estiverem prontos, perderão a nota de apresentação. Tempo de apresentação: 20 minutos p/ grupo.

5. Referências

[Tanenbaum, 2003] Tanenbaum, A. Sistemas Operacionais Modernos. 2a. Ed (livro texto do curso) *(essa referência é importante para compreensão dos conceitos relativos ao uso de memória em sistemas operacionais).*

Demais referências são relativas ao tema de estruturas de dados e podem ser coletadas do plano de ensino da disciplina.