Universidade de Brasília — Faculdade UnB Gama

Estruturas de Dados e Algoritmos - Turma E — Período: 1o./2015

Prof.: Fernando W Cruz — *Data entrega: 03/05/2015; apresentação em 05/05/2015*

1°. Projeto de pesquisa

A) Descrição do Projeto

Um sistema operacional faz gerenciamento de memória como se ela fosse um vetor dividido em partes menores de 4KBytes denominadas de páginas. Por sua vez, os processos (programas em execução) também são divididos em blocos de 4Kbytes. Portanto, quando são executados, os processos ocupam algumas páginas de modo que a memória em um dado momento é um vetor contendo páginas ocupadas e livres, como está ilustrado na letra (a) da Figura 1, com os processos A, B, C, D e E.

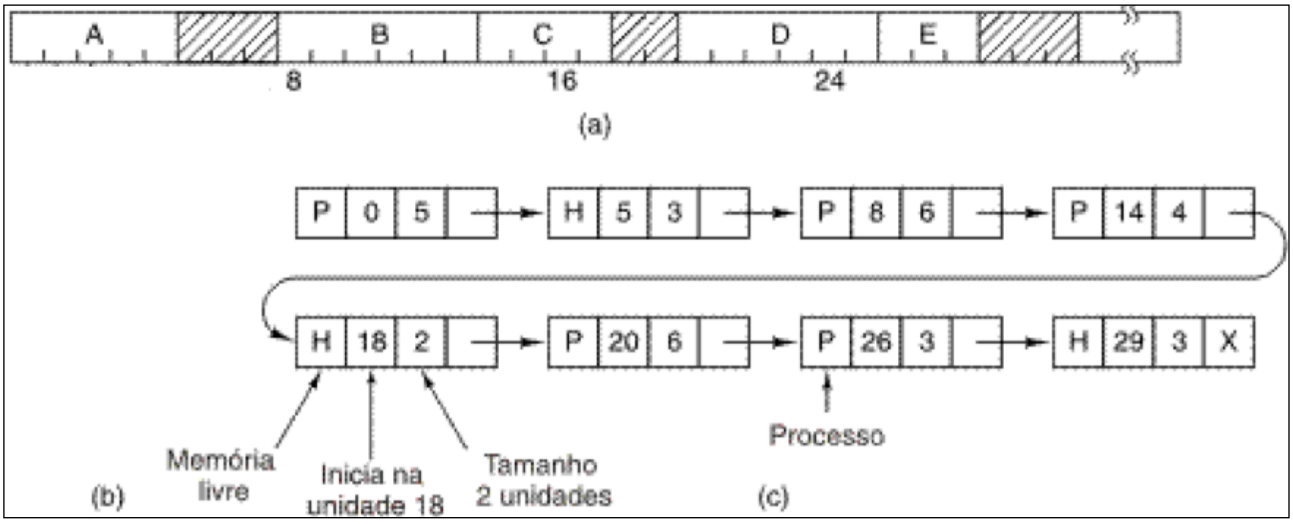

Figura 1 – Exemplo de memória preenchida e estrutura de dados de controle

De acordo com o item (a) dessa figura, o processo A ocupa as cinco primeiras páginas da memória e a área hachurada em seguida representa três páginas não usadas no momento. Os demais processos ficam espalhados em posições variadas e podem ser intercalados por espaços vazios. Na parte (b) da Figura 1 está uma lista encadeada que armazena informações sobre o uso da memória. Nesse caso, cada nó da figura contém as seguintes informações:

- Tipo de nó: P - nó que se refere a um processo qualquer; H - nó que se refere a um buraco que é, na prática, uma área de memória não usada (*hole*)
- Número da página de referência onde se inicia o processo (P) ou espaço em branco (H).
- Tamanho do espaço ocupado pelo processo (P) ou buraco (H)

O objetivo desse projeto é constuir programa que seja capaz de gerenciar a memória de modo a acomodar as demandas dos processos. Ou seja, deve ser capaz de incluir e excluir processos na memória, procurando manter a lista encadeada da letra (b) da Figura 1 com o *status* corrente da memória. De um modo geral, o programa deve ser capaz de tratar as situações descritas para os processos A, X e B na Figura 2 a seguir:

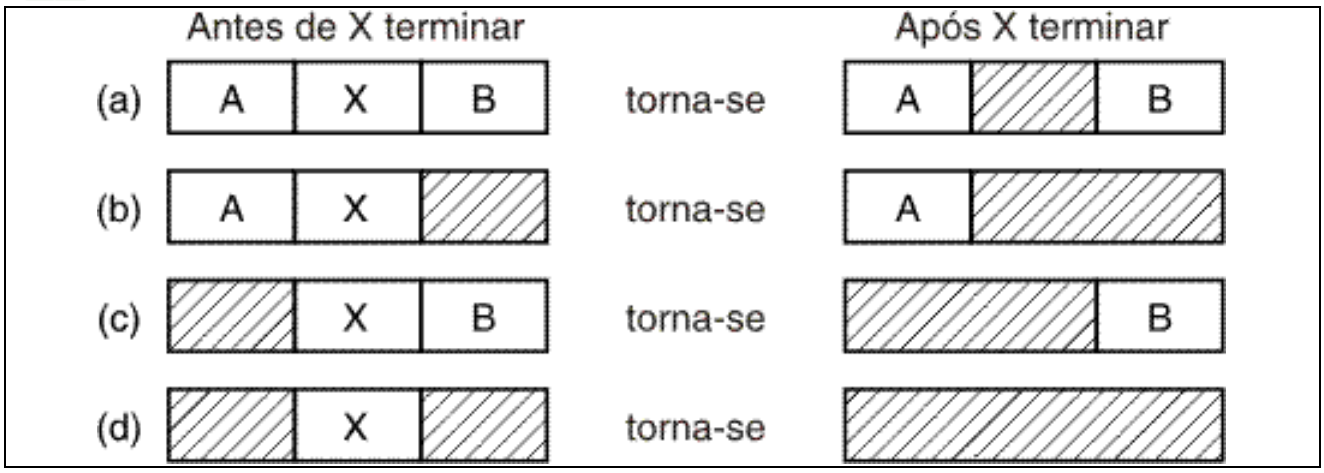

Figura 2 – Situação da memória com os processos A, X e B que devem ser tratados

B) Requisitos do projeto

A solução computacional para este problema deve considerar os seguintes requisitos:

1. O controle da memória deve ser feito com estruturas de dados dinâmicas e uso de cabeçalhos
2. O código deve ser escrito em linguagem C em ambiente Linux/gcc
3. O código deve ser modularizado na forma de funções devidamente documentadas para facilitar a compreensão do texto (a organização do código - na forma de módulos ou funções bem definidas - será levada em consideração para a menção dada ao projeto)
4. O programa deve permitir que o usuário informe o tamanho total da memória e o tamanho das páginas (por exemplo, 4 Kilobytes)
5. O programa deve permitir as funções de inclusão e exclusão de processos a qualquer instante, observando-se os limites da memória (sugere-se uso de arquivos para facilitar a simulação de várias entradas de dados).
6. A inclusão de um processo deve ser feita sempre encaixando-o no primeiro espaço livre encontrado
7. O programa deve ser capaz de atender a uma demanda de exclusão de processos da memória solicitada pelo usuário (isso em geral ocorre quando um processo termina). Cabe ao aluno decidir o que mudar nas estruturas para atender a essa demanda
8. O programa deve ser capaz de atender à demanda do usuário de reorganização dos processos na memória de modo a garantir que as áreas livres espalhadas na memória sejam colocadas em conjunto
9. O programa deve apresentar o *status* da memória e da lista encadeada relacionada a qualquer momento. Nesse caso, os alunos devem usar alguma biblioteca gráfica que facilite a visualização sobre o status da memória/lista encadeada
10. O programa deve simular a inclusão/exclusão de vários processos e caberá ao aluno definir a forma adequada para encerramento do programa.

C) Questões de Ordem

- A data para entrega do projeto é 03/05/2015 via Moodle. A data de apresentação do projeto é 05/05/2015.
- O grupo pode ser formado por até 3 alunos e todos devem estar presentes no dia da apresentação
- O grupo deve apresentar a aplicação funcionando no laboratório (e entregá-la em mídia digital) até a data de entrega do projeto. Obs.: O código-fonte gerado em linguagem C, deve estar documentado (trazer em arquivo .c para testes no dia da apresentação). Obs.: Códigos com erro de compilação não serão corrigidos.
- Cada grupo deve entregar um relatório, contendo:
  i) Uma descrição do problema
  ii) A metodologia utilizada (como cada grupo se organizou para realizar o atividade, incluindo um roteiro sobre os encontros realizados e o que ficou resolvido em cada encontro)
  iii) Uma descrição da solução, incluindo algoritmos, estruturas de dados envolvidas bem como os passos necessários para uso do código
  iv) Opiniões pessoais sobre o projeto de cada um dos participantes (inclua sugestões, principais dificuldades, maiores lições etc)
  v) Um anexo contendo uma descrição sobre as funções e estruturas de dados criadas.

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.

D) Referências

[Tanenbaum, 2003] Tanenbaum, A. Sistemas Operacionais Modernos. 2a. Ed (livro texto do curso). Essa referência é importante para compreensão dos conceitos relativos à gerenciamento de memória.

Obs.: Sobre estruturas de dados, utilizar as referências bibliográficas listadas no plano de ensino da disciplina EDA.