



## Políticas de Alocação de Memória

Universidade Federal de Uberlândia  
Faculdade de Computação  
Prof. Dr. rer. nat. Daniel D. Abdala

## Na Aula Anterior...

- Gerenciamento sem abstração de memória;
  - Multiprogramação sem abstração de memória;
  - Realocação estática;
  - Hardware de proteção;
- Espaço de endereçamento;
  - Realocação Dinâmica (registros base|limite);
  - Troca de Memória;
  - Memória Virtual;

2

## Nesta Aula

- Conceitos Gerais;
- Mapas de Bits;
- Listas Livres;
- Algoritmos de alocação de memória.

3

## Conceitos Gerais

- Gerenciamento de Memória Livre;
- Necessário quando a memória é alocada dinamicamente;
- Responsabilidade do SO gerenciar;
- Duas formas principais:
  - Mapa de Bits;
  - Listas Livres

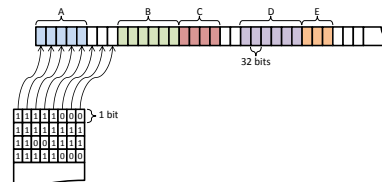
4

## Mapa de Bits

- Memória é dividida em **unidades de alocação – UA**;
- Tamanho em bytes da UA é um parâmetro do SO;
- Um bit é usado para representar cada UA;
- UAs pequenas requerem mais espaço para o mapa de bits e grandes menos bits;
- Problemas:
  - Pequena:
    - Muita memória gasta com o Mapa de bits;
    - Percorrer o mapa de bits se torna mas custoso;
  - Grande:
    - Desperdício de memória quando a quantidade de memória a ser alocada não é múltipla da UA;

5

## Ex: Mapa de Bits



6

### Ex: Mapas de Bits

- Considere o seguinte problema:
- Um SO utiliza mapas de bits para gerenciar a memória livre;
- Cada UA equivale a uma word, ou seja, 32 bits;
- Considerando que a memória disponível no sistema seja de 4GB, quando % dela será utilizada para armazenar o mapa de bits?

7

### Ex: Mapa de Bits

- Para cada 32 bits, é necessário 1 bit no mapa de memória;
- $4GB = 32N$  bits
- Neste contexto o mapa de bits ocupará N bits;
- $4GB = 4.294.967.296$  bytes
- $34.359.738.368$  bits = 32N bits
- $N = 1.073.741.824$  bits =  $134.217.728$  bytes  $\approx$  134 MB usado pelo mapa de bits

8

### Ex: Mapas de Bits

- Considere agora que a UA seja igual a 2KB;
- Quanta memória seria necessária para gerar um mapa de bits para 4GB de memória?

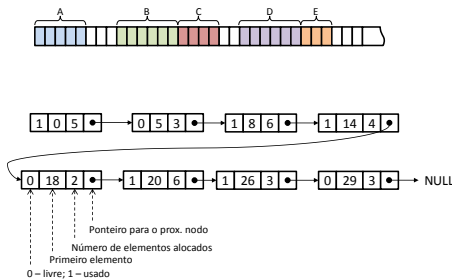
9

### Listas Livres

- Baseado na ideia de listas esparsas;
- Armazenar apenas o estado do bloco de memória, seu ponto de início e tamanho;
- Quais vantagens são evidentes em relação ao mapa de bits;
- Note que o mesmo conceito de UA se aplica as listas livres.

10

### Listas Livres



11

### Implementação de Listas Livres

- Embora a implementação da listas livres como uma lista encadeada seja possível, ela é necessariamente a melhor estrutura de dados para o serviço?
  - Considere o problema de alocar e liberar um bloco de memória!

12

## Alocação/Liberação de Memória

- Alocação é feita via uma chamada do sistema;
- O mesmo se aplica a liberação de memória;
- O gerenciador de memória se encarrega de executar os algoritmos de alocação/liberação de memória;

```
#include <stdlib.h>
void *malloc(size_t size);
void free(void *ptr_to_memory);
```

13

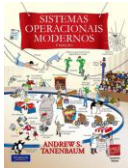
## Algoritmos de Alocação

- Basicamente procurar uma região na memória que acomode a requisição;
- Diversas variantes:
  - First fit;
  - Next fit;
  - Best fit;
  - Worst fit;
  - Quick fit.

14

## Bibliografia - Básica

- 3ª Edição
- Páginas 112-114



15