

Sistemas Operacionais

Estrutura de Dados em Sistemas Operacionais

Felipe Augusto Lima Reis
felipe.reis@ifmg.edu.br



**INSTITUTO
FEDERAL**
Minas Gerais

Sumário



- 1 Introdução
- 2 Estruturas Básicas
- 3 Árvores
- 4 Outras estruturas

INTRODUÇÃO

Introdução

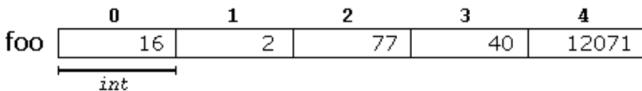
- Para um bom entendimento dos Sistemas Operacionais, é necessário o entendimento das estruturas usadas para armazenamento de dados e controle de processos/atividades;
- Serão detalhadas as seguintes estruturas:
 - Arrays, listas, filas e pilhas;
 - Árvores;
 - *Hashtables* e mapas.

ARRAYS, LISTAS, FILAS E PILHAS

Arrays

- **Array** é uma estrutura dados básica, onde cada elemento pode ser acessado diretamente.
- A memória principal é construída como um array
 - Se o item a ser armazenado é maior que um byte, então múltiplos bytes podem ser alocados para o item;
 - O item é endereçado como índice do item \times tamanho do item [Silberschatz et al., 2012].

```
int foo [5] = { 16, 2, 77, 40, 12071 };
```



Representação de Array em C++.
Fonte: Adaptado de [cplusplus.com, 2021]

Listas

- **Listas** são estruturas utilizadas para coleção de dados em uma sequência, onde os dados são acessados em uma ordem particular.
- Possui como vantagem possibilidade de armazenamento de dados de forma não contínua;
- É considerada um tipo de estrutura de dados linear e dinâmica, onde espaços de memória podem ser adicionados à medida em que são necessários.

Listas

- A forma mais comum de implementação de listas são as **listas encadeadas** (*linked lists*)
 - Cada item (ou célula) possui um espaço destinado ao armazenamento de informações e um ponteiro, que armazena o endereço de memória do próximo item;
 - Caso não exista um próximo item, a última célula aponta para NULL.



Lista encadeada.

Fonte: [Silberschatz et al., 2012]

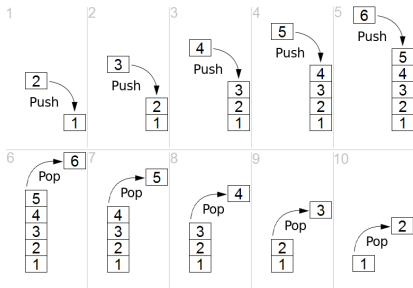
Nota: Existem diferentes tipos de implementação de listas, como listas duplamente encadeadas e listas circulares.

Listas

- Listas possibilitam fácil inserção e exclusão de dados;
- Uma desvantagem é a necessidade de percorrer todos os elementos, no pior caso, para encontrar uma determinada informação;
- As listas, muitas vezes, são usadas diretamente pelos algoritmos do *kernel*
 - No entanto, elas são mais utilizadas para construção de estruturas de dados mais robustas, como pilhas e filas [Silberschatz et al., 2012].

Pilhas

- **Pilha (*stack*)** é uma estrutura de dados ordenada onde o último item a entrar (na pilha) será o primeiro a ser removido.
 - Esse tipo de estrutura é denominada **LIFO** (*Last In, First Out*).



Estrutura de dados Pilha.

Fonte: [Wikipedia Contributors, 2021b]

Nota: *Last In, First Out* = Último a entrar, primeiro a sair.

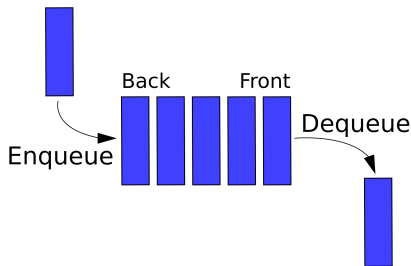
Pilhas



- Pilhas são utilizadas pelo Sistema Operacional durante a chamadas de uma função
 - Parâmetros, variáveis locais e o endereço de retorno são colocados na pilha quando uma função é chamada;
 - Ao retornar da chamada de função, o SO retira esses itens da pilha [Silberschatz et al., 2012].
- A operação de adição de um item na pilha é denominado **push**, enquanto a ação de retirar um item da pilha é denominada **pop** [Silberschatz et al., 2012].

Filas

- **Fila** (*queue*) é uma estrutura de dados ordenada onde o primeiro item a entrar (na fila) será o primeiro a ser removido.
 - Esse tipo de estrutura é denominada **FIFO** (*First In, First Out*).



Estrutura de dados Fila.

Fonte: [Wikipedia Contributors, 2021a]

Nota: *First In, First Out* = Primeiro a entrar, primeiro a sair.

Filas



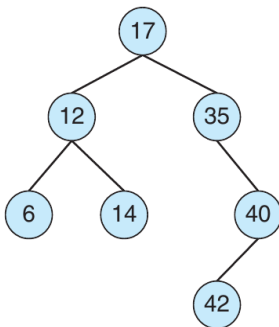
- Filas são utilizadas pelo Sistema Operacional em tarefas cuja ordem de execução deve seguir a ordem de requisição
 - Ao imprimir documentos, a ordem de impressão é a mesma da ordem de solicitação;
 - Tarefas a serem processadas também são organizadas em filas¹ [Silberschatz et al., 2012].
- Operações de adição e retirada de itens da fila são denominadas, respectivamente, **push** (enfileirar ou *enqueue*) e **pop** (desenfileirar ou *dequeue*).

¹Veremos ao longo do curso, que algumas filas podem ter prioridade.

ÁRVORES DE DADOS

Árvores de Dados

- **Árvores** são estrutura de dados que podem representar informações hierárquicas [Silberschatz et al., 2012].



Árvores de dados.

Fonte: [Silberschatz et al., 2012]

Árvores de Dados



- As informações são armazenadas nos nós das árvores
 - O **nó raiz** é o nó inicial, frequentemente representado no alto da estrutura;
 - O nó raiz se liga a outros nós (denominados **nós filhos**), por meio de arestas (ou ramos);
 - Estes, por sua vez podem ser ligados a outros nós filhos, gerando uma relação pai-filho(s);
 - Um nó que não possui filhos é conhecido como **nó folha** ou nó terminal.

Árvores de Dados



- Em uma estrutura de árvore genérica, um dado nó pode ter filhos ilimitados
 - Algumas estruturas restringem o número de filhos, por questões de desempenho e organização dos dados;
 - Uma **árvore binária** é um tipo de árvore que limita o número de filhos a apenas dois;
 - Em árvores de busca binária, os dados são organizados de modo que o nó do lado esquerdo seja inferior ou igual ao nó do lado direito
 - Essa estrutura reduz o custo de busca de um item na árvore.

Árvores de Dados

- A forma mais comum do uso de árvores é encontrada na organização de diretórios.



Árvores de diretórios.

Fonte: [Wikipedia Contributors, 2019]

HASH TABLES E BITMAPS

Hash tables

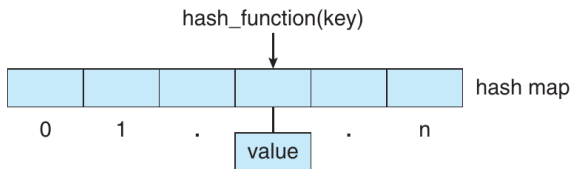
- Um **hash table**² é uma estrutura de dados capaz de mapear chaves em valores.
 - Uma **função hash** recebe um valor como entrada, executa uma operação matemática e retorna um valor numérico (**hash code**);
 - O **hash code** pode ser usado como índice em uma array de dados (ou uma tabela de dados);
 - Tal operação faz com que o acesso a uma determinada informação seja extremamente rápido
 - Idealmente, não é necessário procurar o registro na estrutura de dados³ [Silberschatz et al., 2012]

²Também denominado **tabela hash**, **hash map** ou **map**.

³No entanto, dependendo da função de hash, do tamanho do vetor e do uso da tabela, colisões podem ocorrer.

Hash tables

- Devido ao seu desempenho, as hash tables são amplamente usadas em Sistemas Operacionais [Silberschatz et al., 2012].
 - Hash tables podem ser utilizados para login no sistema;
 - O nome do usuário pode ser utilizado como chave para acesso à senha, que é comparada à senha digitada [Silberschatz et al., 2012].



Hash tables.

Fonte: [Silberschatz et al., 2012]

Bitmaps

- Um **bitmap** é uma string de n dígitos binários que podem ser usados para representar o status de n itens.
- Um recurso, por exemplo, pode ser referenciado como disponível ou indisponível para alocação;
- Possuem como vantagem sua eficiência do uso de espaço;
- São comumente usados quando há necessidade de representar a disponibilidade de um grande número de recursos [Silberschatz et al., 2012].

010101110001001

Bitmap.

Fonte: Adaptado de [Coppin, 2004]

Referências I



Coppin, B. (2004).

Artificial intelligence illuminated.

Jones and Bartlett illuminated series. Jones and Bartlett Publishers, 1 edition.



cplusplus.com (2021).

Arrays.

[Online]; acessado em 09 de Junho de 2021. Disponível em:

<https://www.cplusplus.com/doc/tutorial/arrays/>.



Silberschatz, A., Galvin, P. B., and Gagne, G. (2012).

Operating System Concepts.

Wiley Publishing, 9th edition.



Wikipedia Contributors (2019).

Sistema de ficheiros.

[Online]; acessado em 14 de Abril de 2020. Disponível em:

https://pt.wikipedia.org/wiki/Sistema_de_ficheiros.



Wikipedia Contributors (2021a).

Queue (abstract data type).

[Online]; acessado em 09 de Junho de 2021. Disponível em:

[https://en.wikipedia.org/wiki/Queue_\(abstract_data_type\)](https://en.wikipedia.org/wiki/Queue_(abstract_data_type)).



Wikipedia Contributors (2021b).

Stack (abstract data type).

[Online]; acessado em 09 de Junho de 2021. Disponível em:

[https://en.wikipedia.org/wiki/Stack_\(abstract_data_type\)](https://en.wikipedia.org/wiki/Stack_(abstract_data_type)).

Referências II