

Universidade Cruzeiro do Sul

Tema: Memória virtual

Sistemas Operacionais

Nome: Gabriel Gustavo M. da Silva

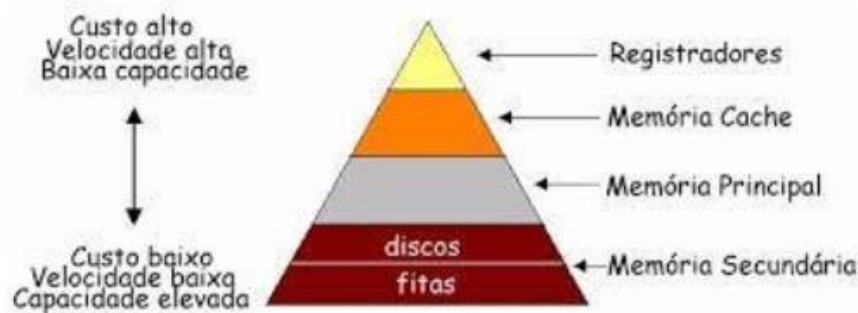
## Sumário

1. ----- Introdução
  - 1.1 Objetivo
  - 1.2 Justificativa
  - 1.3 Metodologia
  - 1.4 Contribuições
2. ----- Tema
3. ----- Trabalhos  
Relacionados
4. ----- Conclusões
5. ----- Referências

## 1 - Introdução

Memórias são componentes fundamentais em um computador, uma vez que são responsáveis pelo armazenamento de dados, informações, arquivos, processos e programas, e existem em diversas formas podendo ser organizadas em uma hierarquia. Tais memórias possuem características e finalidades específicas, mas em geral podem ser caracterizadas como voláteis (perdem o conteúdo armazenado na ausência de energia elétrica) ou não voláteis (permanentes).

Pode-se descrever a hierarquia de memória da seguinte forma:



**Registradores:** memória volátil responsável pelo armazenamento de dados utilizados pela unidade lógica aritmética e está presente dentro do processador.

**Cache:** memória volátil responsável pelo armazenamento de dados frequentemente utilizados pela C.P.U.

**Memória Principal:** memória volátil responsável pelo armazenamento de processos a serem executados pela C.P.U.

**Memória Secundária:** memória não volátil responsável pelo armazenamento de dados, arquivos, programa e informações permanentes.

Sem dúvidas, cada tipo de memória possui seu devido valor, mas a memória principal, em essencial, é extremamente importante para a eficiência da máquina, uma vez que é responsável por reduzir tempo de espera do processador por novas instruções. No entanto, durante grande parte da história da computação até os dias atuais, em comparação com memórias secundárias, esse recurso é limitado, o que se deve principalmente ao alto custo monetário em relação a capacidade da memória.

Um dos principais problemas enfrentados no desenvolvimento dos primeiros sistemas operacionais, no que se refere a alocação de memória, era que o processo não cabia por completo na memória principal, sendo necessário sua divisão em blocos menores de

instruções que pudessem ser processados separadamente. Essa técnica ficou conhecida como overlay.

A técnica de Overlay ainda é utilizada em alguns sistemas monoprogramáveis que utilizam alocação de memória contígua simples, mas com o tempo e o crescimento do número de processos, houve a necessidade de se implementar sistemas multitarefas e multiprogramáveis e assim, a alocação contígua simples tornou-se inviável nesse cenário. Além disso, a técnica de Overlay era totalmente dependente da arquitetura, a divisão do programa em módulos era um trabalho lento e a interação dos módulos entre si era de total responsabilidade do programador e o sistema operacional ficava responsável apenas pela permutação entre módulos.

Com isso, viu-se necessário criar maneiras para um melhor gerenciamento desse recurso valioso.

Com o tempo foram propostas outras formas de se alocar e particionar a memória, estática e dinamicamente. Isso possibilitou que mais processos fossem alocados na memória principal, mas não resolveu por completo o problema de capacidade: “Mais processos requerem mais recursos”. Dessa forma, surgiu a técnica de memória virtual.

### **1.1 - Objetivo**

Explicar o que é memória virtual, o motivo de sua criação, suas principais características e formas de implementação, reunindo o máximo de informações relevantes de maneira concisa, buscando “desmistificar” como a memória virtual contribui para os principais sistemas operacionais.

### **1.2 - Justificativa**

O tema, memória virtual, apesar de ser extremamente importante, acaba por passar eventualmente despercebido entre os estudantes da área de tecnologia da informação, mesmo esses estando em contato direto com dispositivos que implementam essa técnica. Você já parou para pensar como seria o mundo atual se não existissem sistemas operacionais multiprogramados?

Sem a memória virtual, a implementação desses sistemas seria praticamente impossível... e esse trabalho possui o intuito de explicar o motivo.

### **1.3 - Metodologia**

A metodologia utilizada foi: Pesquisas em artigos científicos, vídeo aulas, apresentações e anotações feitas em sala de aula.

### **1.4 - Contribuições**

Esse projeto busca facilitar o entendimento de memória virtual e reunir concisamente informações sobre o tema de modo que outros estudantes possam utiliza-las, poupando grande parte do seu tempo com pesquisas.

## **2 - Tema**

### **O que é memória virtual?**

A técnica de memória virtual consiste na utilização de parte da memória secundária como memória principal, dessa forma, as duas memórias são combinadas possibilitando um “aumento lógico” na memória principal. Em outras palavras, a memória virtual refere-se a um segmento da memória secundária que atua como uma “extensão” da memória principal.

### **Espaços de endereçamento físico e virtual**

O conjunto de endereços discretos em uma memória recebe o nome de espaço de endereçamento e pode ser associado a registradores reais (ou físicos) e lógicos (ou virtuais).

A memória principal é dividida em células e cada uma possui seu endereço único, ou seja, célula é a menor unidade endereçável da memória principal, esses endereços são formados por grupos de bits e na memória principal são endereços físicos. No entanto, memórias secundárias possuem formas de endereçamento diferentes da memória principal, devido a sua arquitetura e organização.

Como exemplo, pode-se citar os HDs, que fisicamente são constituídos por segmentos denominados setores, que em conjuntos concêntricos formam trilhas, que por sua vez formam cilindros. Neste caso, os endereços físicos estão associados a posição física em que o dado se encontra e normalmente as trilhas e cilindros recebem uma numeração.

Endereços virtuais são utilizados pelos processos em técnicas de memória virtual, permitindo que os processos sejam alocados virtualmente e utilizem memória principal concorrentemente. Eles não possuem relação direta com endereços reais no que se refere as posições de memória ao se alocar um processo e assim, partes de um mesmo processo não serão necessariamente alocadas em posições adjacentes na memória principal, sendo necessário o mapeamento de memória.

### **Mapeamento de memória**

Denomina-se mapeamento de memória o processo de associar endereços da virtuais com endereços físicos. Tal “conversão” é realizada por um circuito integrado, a unidade de gerenciamento de memória a (MMU), uma solução a nível de hardware que pode estar ou não localizada dentro da CPU.

## Principais técnicas de memória virtual

### Swapping:

É uma técnica de gerenciamento de memória que consiste em trocar processos, de modo que, os processos que precisem utilizar a memória principal aguardem e depois de um tempo são trazidos do disco (memória virtual) para a memória primária (swapped in) e posteriormente retirados (swapped out).

### Paginação:

Técnica de memória virtual que consiste na divisão da memória e dos processos em partes de mesmo tamanho fixo, respectivamente denominadas blocos ou frames e páginas virtuais. Essa técnica não separa informações do mesmo tipo em páginas diferentes, podendo haver códigos dados e textos em uma mesma página.

Para que seja possível mapear uma página virtual, que se encontra na memória secundária a um frame correspondente, na memória principal, utiliza-se uma estrutura, a tabela de páginas, que é única para cada processo. Essas tabelas podem ser armazenadas em arrays de registradores, na memória cache na MMU ou na própria RAM.

Quando as tabelas são armazenadas na RAM, dois registradores são utilizados. O 1º é o registrador base (ou PTBR), que atua apontando para o início da tabela e é responsável por indicar o endereço físico de memória onde a tabela está alocada. O 2º é o registrador de tamanho da tabela de página (ou PTLR) e atua indicando tamanho da tabela de página (número de entradas da tabela ou número de páginas).

Entretanto, com número de acessos pode ocorrer um overhead (custo alto de recursos), já que se acessará a memória para buscar instruções na tabela e outro para o dado em si. Como solução, propõe-se uma cache da tabela das páginas mais usadas, ou TLB Translation Lookaside Buffer. Exemplo representado a seguir.

Valid	Virtual page	Modified	Protection	Page frame
1	140	1	RW	31
1	20	0	R X	38
1	130	1	RW	29
1	129	1	RW	62
1	19	0	R X	50
1	21	0	R X	45
1	860	1	RW	14
1	861	1	RW	75

Dessa forma, antes de uma página ser procurada na tabela de página será procurada na TLB. Caso não seja encontrada na TLB, mas estiver na RAM ocorrerá uma falha chamada Soft miss e a TLB será atualizada para solucionar. Caso página não esteja na TLB e nem RAM ocorrerá uma falha chamada Hard miss e ela será trazida do disco à RAM e depois para TLB.

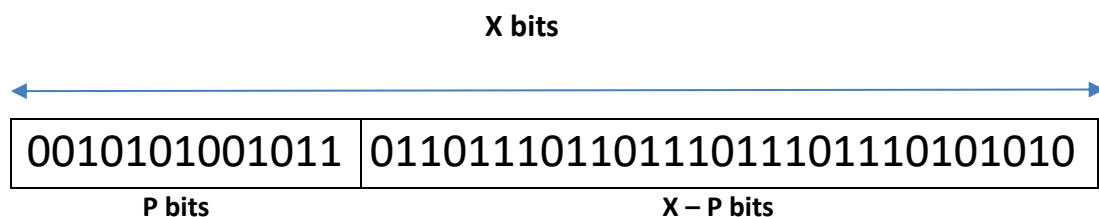
### Componentes da tabela de paginação

#### Endereços:

O mapeamento é feito a partir do cálculo relativo ao endereço lógico acessado pelo processo. Esse endereço de x bits é composto por:

- Número da página (p bits) – usado como índice de uma página na tabela de páginas. O número de páginas que se pode endereçar é calculado por  $2^p$ .
- Deslocamento de página (d bits), também conhecido como offset – é o número de endereços possíveis dentro de uma página dado por x-p bits, ou seja,  $2^{x-p}$  endereços que também corresponde ao tamanho de uma página.

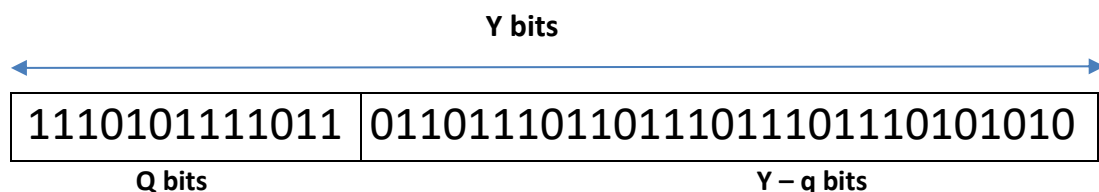
Exemplo:



De forma semelhante, o endereço físico é composto por:

- Número de frames (q bits) – usado como índice para identificar o quadro na memória principal. O número de frames que se pode endereçar é calculado por:  $2^q$ .
- Deslocamento de frame (y-q bits) é o número de endereços possíveis que se pode endereçar dentro de um frame.

Exemplo:



### **Calculando o tamanho de endereços lógicos e reais a partir do tamanho das memórias.**

Tamanho do endereço (T) =  $\log_2 C$  bits; em que C é a capacidade da memória em bytes.

Exemplo:

Memória virtual de 64 KB.

$$X = \log_2 65536 \text{ bits} = 16 \text{ bits.}$$

Memória Física de 32 KB.

$$Y = \log_2 32768 \text{ bits} = 15 \text{ bits}$$

Considerando que se tem páginas e frames de 4KB, pode-se calcular o número de páginas e de quadros, da seguinte forma:

$$64 \text{ KB de memória virtual} / 4\text{KB por página} = 16 \text{ páginas virtuais.}$$

$$32 \text{ KB de memória real} / 4\text{KB por quadro} = 8 \text{ quadros.}$$

E se pode calcular o número de bits necessários para endereçar as páginas e quadros da seguinte forma:

$$P = \log_2 16 = 4$$

$$Q = \log_2 8 = 3$$

Eventualmente  $P > Q$ .

### **Outros componentes de uma tabela de páginas.**

**Bit de residência (presente/ausente)** – Informa se a página se encontra ou não na memória principal.

**Bits de proteção** – 0 (leitura/escrita) 1 (leitura) 2 (execução).

**Bit de modificação** – Informa se a página foi alterada (1) ou não (0).

**Bit de referência** – 1 (foi referenciada recentemente).

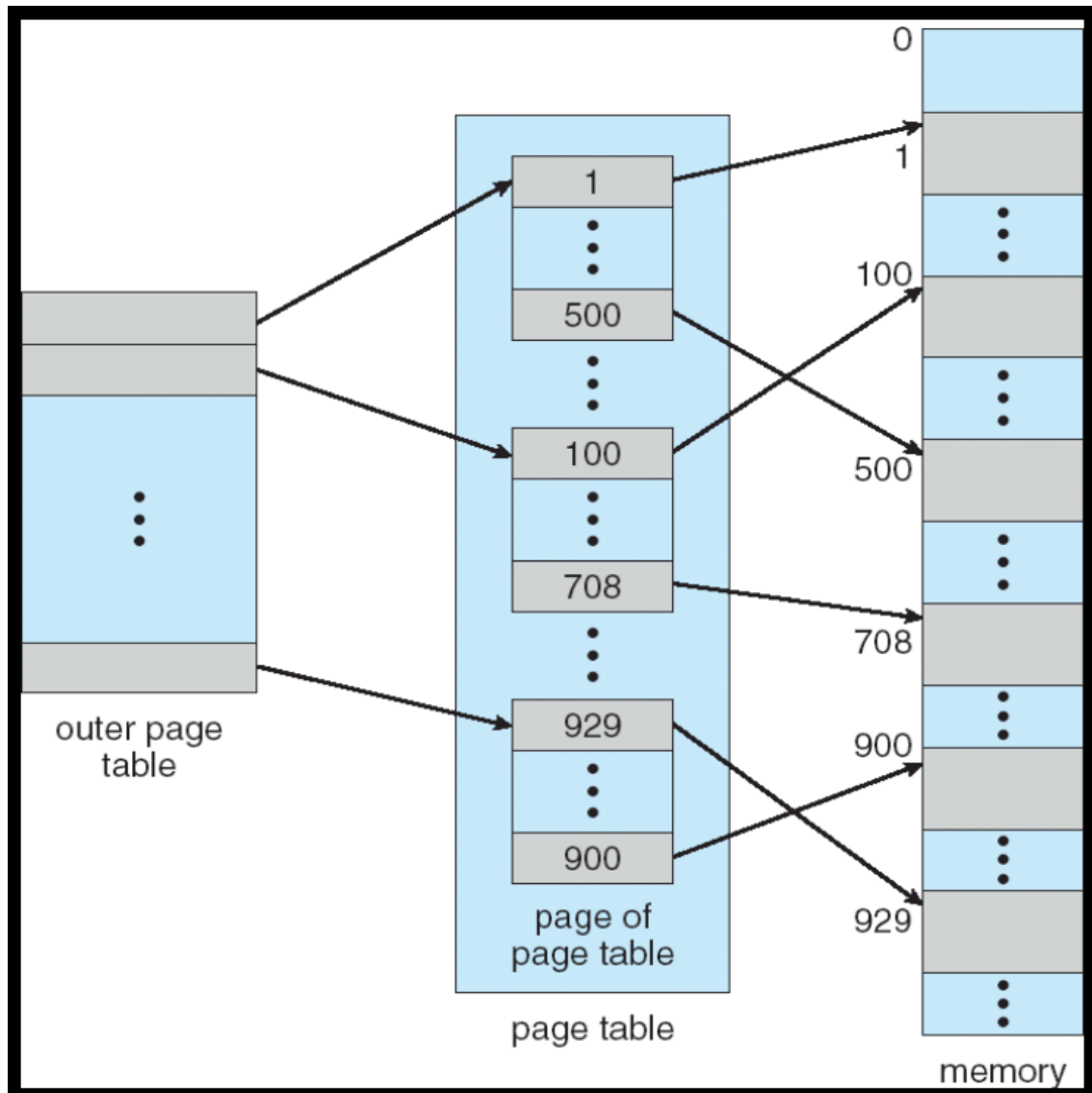
**Bit de cache** – Permite desabilitar o caching da páginas.



### Problemas tabelas de páginas grandes.

Um dos grandes problemas da paginação é o tamanho da tabela de páginas. Tabelas de páginas extensas ocupam uma região significativa e preciosa da memória e para contornar esse problema, as tabelas de páginas podem ser estruturadas das seguintes formas:

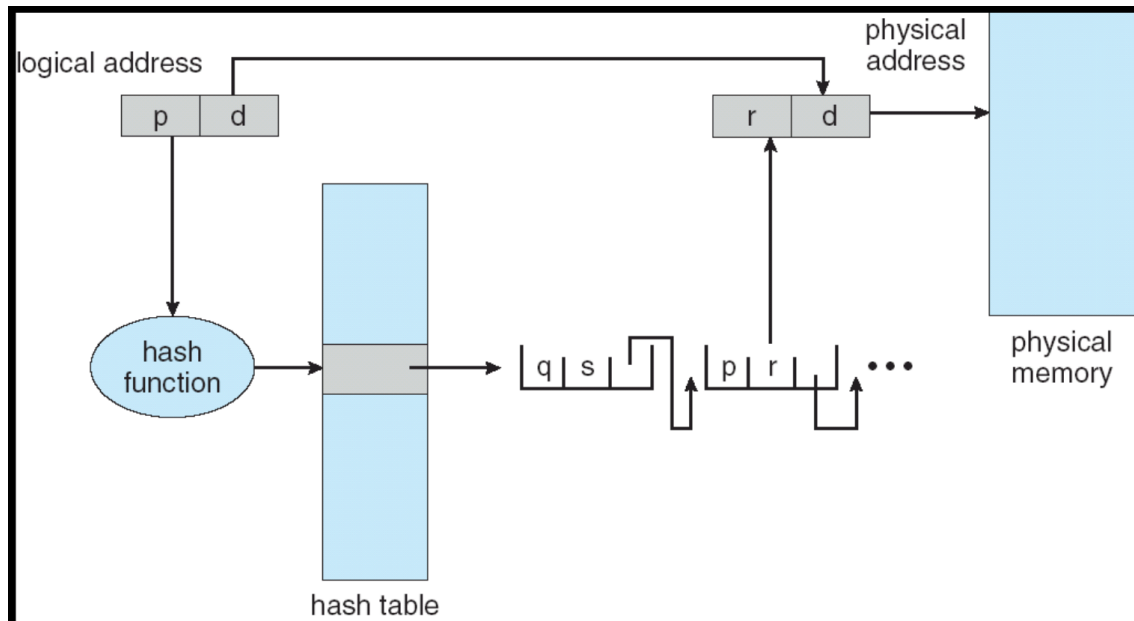
#### Paginação Hierárquica:



Consiste em “paginar” uma tabela de páginas, ou seja, uma tabela de páginas será quebrada em múltiplas tabelas de páginas e consequentemente o endereço lógico será subdividido, como exemplo acima que representa uma tabela de páginas de dois níveis. Dessa forma, é possível carregar na memória principal apenas as páginas mais utilizadas

### Tabelas de páginas em Hash:

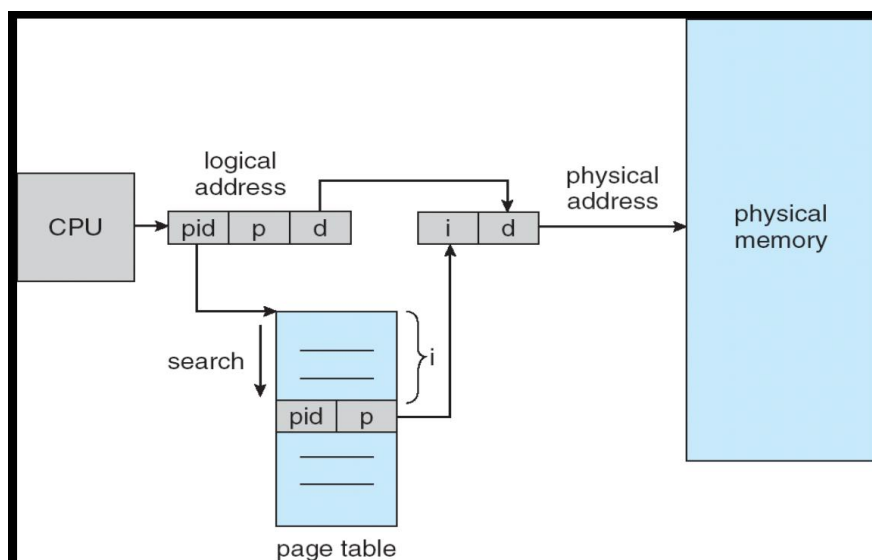
Nessa solução, o número da página virtual em usado em uma função hash. Nessa tabela serão armazenados em cada “registro” o número da página virtual  $q$ , o número do frame  $s$  e, caso ocorra colisões, um ponteiro para um próximo item, como o exemplo abaixo:



Todavia, apesar dessa solução funcionar bem com 32 bits torna-se inviável com 64 bits já que o tamanho da tabela passará a ocupar aproximadamente 30 Peta bytes = 30.000.000 GB e como solução propõe-se:

### Tabela de página invertida:

Nessa solução, o mapeamento é realizado tendo como base os endereços físicos e uma única tabela será utilizada para mapear todo o sistema. A entrada será composta por PID (Process ID.) e o endereço lógico que serão procurados na tabela e sendo encontrados um índice será informado e junto com deslocamento fornecerá o endereço físico. Como no exemplo a seguir:



### **Políticas de alocação de páginas:**

**Fixa:** Cada processo possui um número máximo de páginas reais, definido quando o processo é criado e o limite é igual para todos os processos.

- Vantagem: simples de ser implementado.
- Desvantagens:
  1. Número muito pequeno de páginas reais causa muita paginação.
  2. Número muito grande de páginas reais pode causar desperdício de memória principal.

**Dinâmica:** O número máximo de páginas reais por cada processo varia em tempo de execução.

- Vantagem: processos com elevada taxa de paginação podem ter seu limite de páginas reais ampliado. Processos com baixa taxa de paginação podem ter seu limite de páginas reais reduzido
- Desvantagem: monitoramento constante para saber das suas necessidades quanto ao número de páginas.

### **Busca de páginas:**

Paginação simples – Todas as páginas são carregadas na memória principal.

Paginação por demanda – As páginas são carregadas na memória principal assim que o processador executar uma instrução que gere um page fault (evento que ocorre quando uma página é requisitada na memória e não é encontrada).

### **Segmentação:**

Outra técnica de memória virtual é a segmentação e consiste na segmentação dos espaços de endereçamentos em blocos de tamanho arbitrário chamados de segmentos. Esses segmentos possuem apenas informações do mesmo tipo, diferentemente da paginação.

Uma das vantagens da segmentação é que cada tipo de segmento pode ser tratado de maneira diferente, por exemplo: códigos podem ser apenas lidos não havendo a necessidade de serem alterados.

### **3 - Trabalhos relacionados:**

Mapeamento e endereçamento em disco secundário, disponível em:

[http://web.mit.edu/rhel-doc/4/RH-DOCS/rhel-isa-pt\\_br-4/s1-storage-data-addr.html](http://web.mit.edu/rhel-doc/4/RH-DOCS/rhel-isa-pt_br-4/s1-storage-data-addr.html)

obs. Apesar de citar memória virtual no seu início, não possui tantos detalhes sobre paginação e em contrapartida é bem detalhado quanto a arquitetura e organização de armazenamentos secundários.

#### **Dissertação sobre memória virtual:**

Disponível em: <http://www.inf.ufpr.br/roberto/dissLauri.pdf>

Obs. Dissertação bem detalhada, porém pouco concisa.

### **4 - Conclusões.**

Memórias virtuais são fundamentais para os sistemas operacionais, principalmente no cenário atual para sistemas multiprogramados, uma vez que auxiliam na solução do problema do tamanho limitado da memória principal que está associado ao grande número de processos concorrentes a seu custo monetário, utilizando soluções “mais baratas”, como o disco secundário e as técnicas mais utilizadas para isso são: a segmentação e a paginação.

## 5 - Referências:

<https://www.gsigma.ufsc.br/~popov/aulas/so1/cap9so.html>

<https://www.ime.usp.br/~song/mac344/slides07-virtual-memory.pdf>

<http://www.ic.uff.br/~viviane.silva/2009.1/ocl/util/aula06.pdf>

[http://web.mit.edu/rhel-doc/4/RH-DOCS/rhel-isa-pt\\_br-4/s1-storage-data-addr.html](http://web.mit.edu/rhel-doc/4/RH-DOCS/rhel-isa-pt_br-4/s1-storage-data-addr.html)

[https://edisciplinas.usp.br/pluginfile.php/343992/mod\\_resource/content/1/Disco%20R%C3%ADgido\\_2013.pdf](https://edisciplinas.usp.br/pluginfile.php/343992/mod_resource/content/1/Disco%20R%C3%ADgido_2013.pdf)

[https://pt.wikipedia.org/wiki/Espa%C3%A7o\\_de\\_endere%C3%A7amento](https://pt.wikipedia.org/wiki/Espa%C3%A7o_de_endere%C3%A7amento)

<https://www.ime.usp.br/~song/mac412/oc/node19.html>

<http://www.inf.ufrgs.br/~johann/sisop1/aula14b.paginas.pdf>

<http://www.inf.ufpr.br/roberto/dissLauri.pdf>

<http://escalonamentoprocessos.blogspot.com/2010/12/swapping-x-paginacao-x-segmentacao-x.html>

<http://www.inf.ufrgs.br/~johann/sisop1/aula15.tlb.pdf>

<https://www.youtube.com/watch?v=Q8ZqjEafmNc&list=PLxI8Can9yAHeK7GUEGxMsqoPRmJKwI9Jw&index=18>

<https://www.youtube.com/watch?v=QjieJPckuDI&list=PLxI8Can9yAHeK7GUEGxMsqoPRmJKwI9Jw&index=19>

<https://www.youtube.com/watch?v=4EaBN98dk40&list=PLxI8Can9yAHeK7GUEGxMsqoPRmJKwI9Jw&index=20>

<http://www.dsc.ufcg.edu.br/~pet/jornal/marco2014/materias/recapitulando.html>

<http://wiki.icmc.usp.br/images/5/5a/Aula11.pdf>

Arquitetura e Organização de Computadores - 8ª Ed. Stallings, William.