

# Memória Virtual

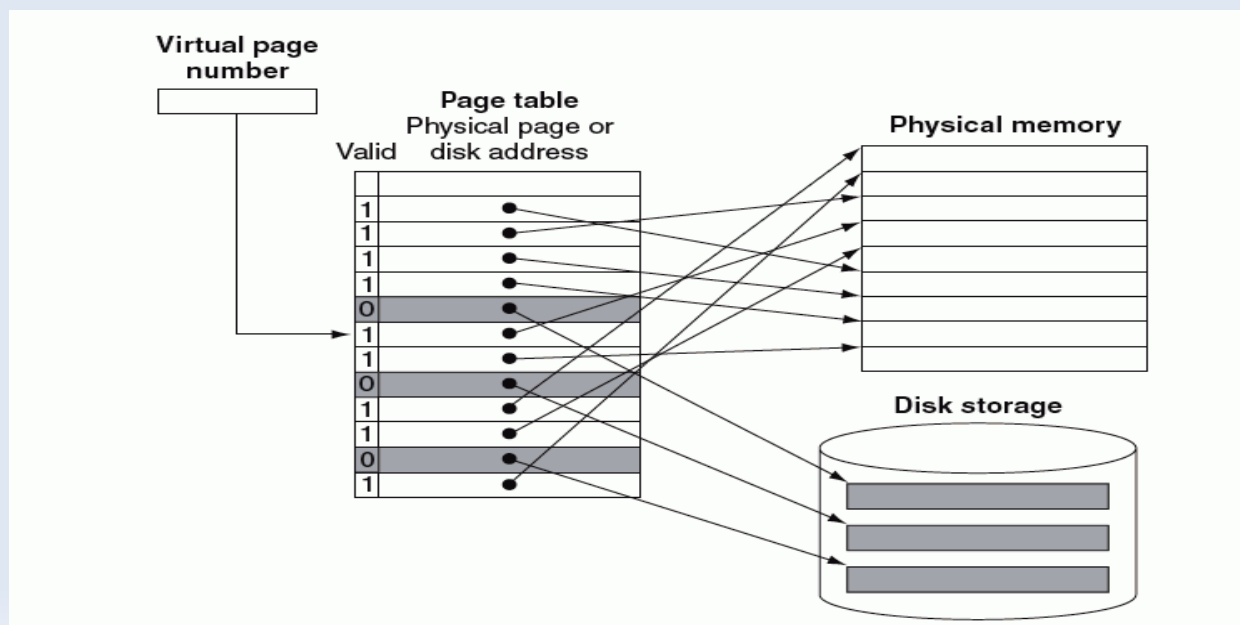
- Prof. M.Sc. Bruno R. Silva
- CEFET-MG Campus VII

# Memória Virtual

- Permitir o compartilhamento seguro e eficiente da memória entre vários programas
- Remover os transtornos de programação de uma quantidade pequena e limitada de memória principal.
- Quatro décadas após a sua invenção, o primeiro motivo é ainda o que predomina.
- A memória virtual nos permite compartilhar eficientemente o processador e a memória principal.
  - Precisamos proteger os programas uns dos outros
  - Um programa pode ler e escrever as partes da memória principal atribuídas a ele.

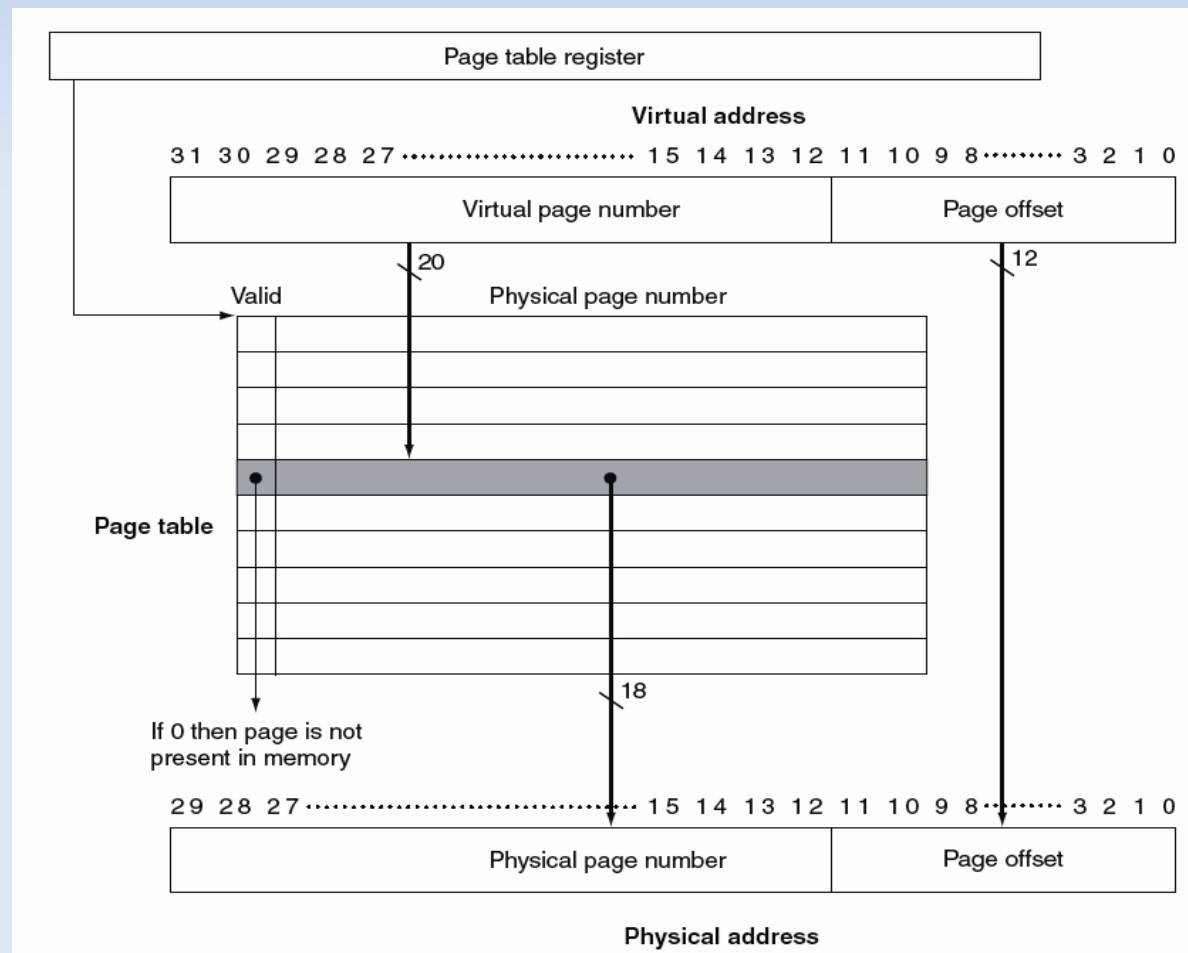
# Memória Virtual

- A memória virtual implementa a tradução do espaço de endereçamento de um programa para os endereços físicos.
- Um bloco de memória virtual é chamado de PÁGINA.
- Uma falha de memória virtual é chamada de FALTA DE PÁGINA
- O processador produz um endereço virtual que é traduzido para um endereço físico por uma combinação de hardware e software.



# Memória Virtual

- Na memória virtual, um endereço é desmembrado em um número de página virtual e um offset de página.



# Memória Virtual

- Uma falta de página tem um altíssimo custo = milhões de ciclos de clock para ser processada.
  - A memória principal é cerca de 100.000 vezes mais rápida que o disco
- O alto custo leva a várias decisões importantes:
  - Páginas devem ser grandes o suficiente para tentar amortizar o longo tempo de acesso.
    - Tamanhos de 4KB a 16KB são comuns
  - Em geral, usa-se o posicionamento totalmente associativo das páginas na memória.
  - As faltas podem ser tratadas no nível de software, pois o overhead será pequeno quando comparado ao tempo de acesso ao disco.
    - O software pode utilizar algoritmos inteligentes para escolher como posicionar as páginas.
  - Sistema de memória virtual utiliza esquema write-back.

# Memória Virtual

- Posicionando uma página e a encontrando novamente
  - Quando houver necessidade de substituir uma página, o S.O. Pode utilizar sofisticados algoritmos e estruturas de dados para monitorar o uso das páginas
    - E assim escolher substituir uma página que não será necessária por longo tempo.
  - A dificuldade de usar um esquema totalmente associativo está em localizar uma entrada.
    - Uma pesquisa sequência é impraticável.
  - Nos sistemas de memória virtual, usamos uma TABELA DE PÁGINAS para indexar a memória. Ela reside na própria memória.
  - Cada programa possui sua própria tabela de páginas que mapeia o espaço de endereçamento virtual deste programa para a memória principal
  - Um registrador de tabela de páginas é incluído no hardware para apontar para o início da tabela de páginas que está em uso atualmente.
  - Por enquanto considere que a tabela de páginas esteja em uma área fixa e contígua da memória
  - Cada entrada na tabela de páginas contém o número da página física para a respectiva página virtual, SE a páginas estiver na memória.

# Memória Virtual

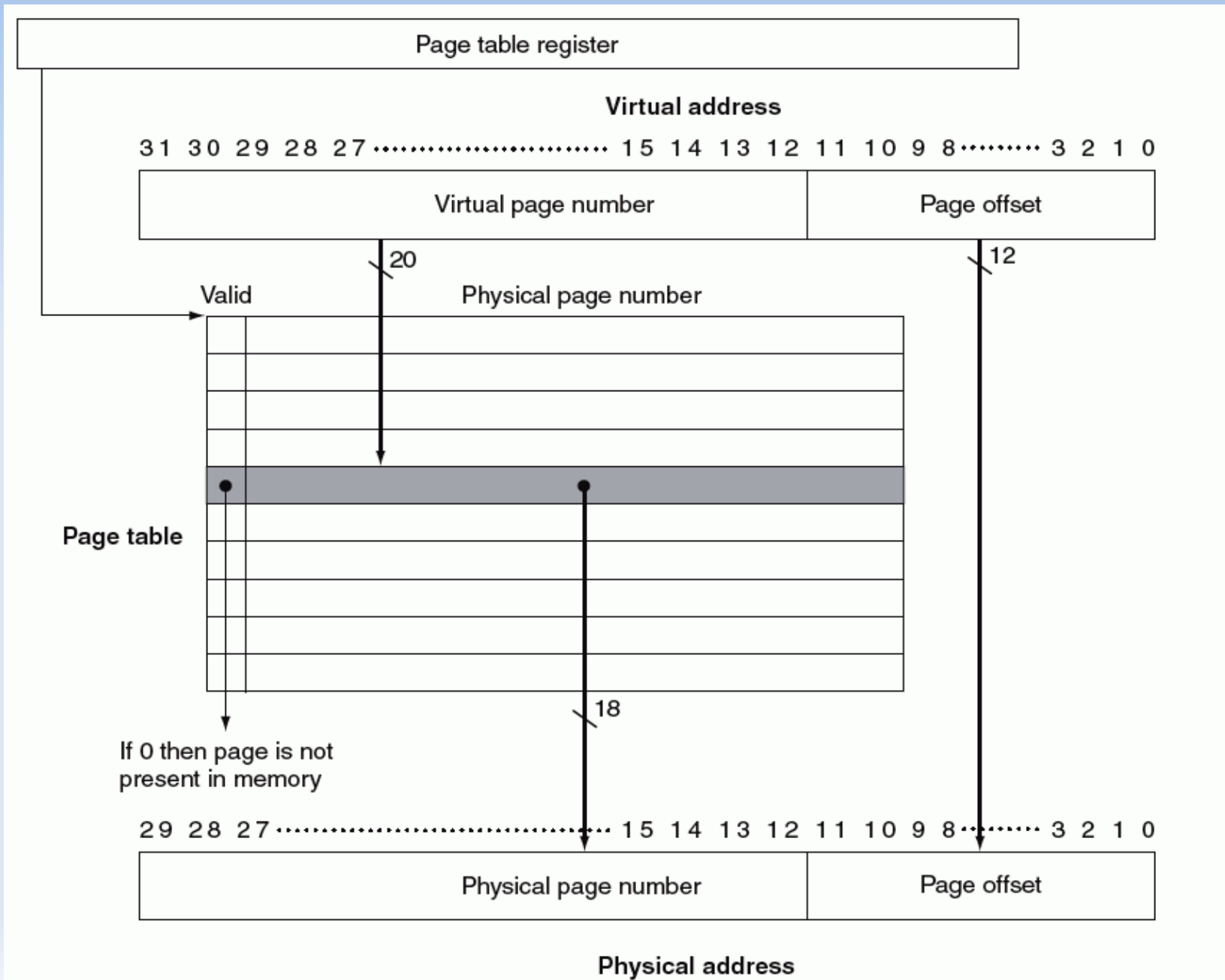
- Estado do processado:
  - Tabela de páginas
  - Contador de programas
  - Registradores
- Se quisermos permitir que outro programa utilize o processador, precisamos salvar o estado atual.
  - Ao restaurar este estado, é possível reiniciar o programa interrompido do ponto onde ele parou.
- Frequentemente nos referimos a esse estado como um processo.
  - O processo é considerado ativo quando está de posse do processador, caso contrário ele é considerado inativo
- O Sistema operacional é o responsável pelo escalonamento dos processos, efetuando a troca de contexto (mudança de estado) quando necessária
- Quando há uma troca de contexto, o S.O. simplesmente carrega o registrador de tabela de páginas, fazendo-o apontar para a tabela de páginas que ele quer tornar ativa.

# Memória Virtual

- Cada processo possui sua própria tabela de páginas
- Diferentes processos usam o mesmo endereço virtual
- O S.O. É o responsável por alocar a memória física e atualizar a tabela de páginas, de modo a evitar colisões no espaço de endereçamento virtual dos processos.
- Cada entrada da tabela de páginas possui um bit de validade que indica se a respectiva página física está ou não presente na memória.
- A tabela de página contém um mapeamento para toda página virtual possível e portanto nenhuma tag é necessária.



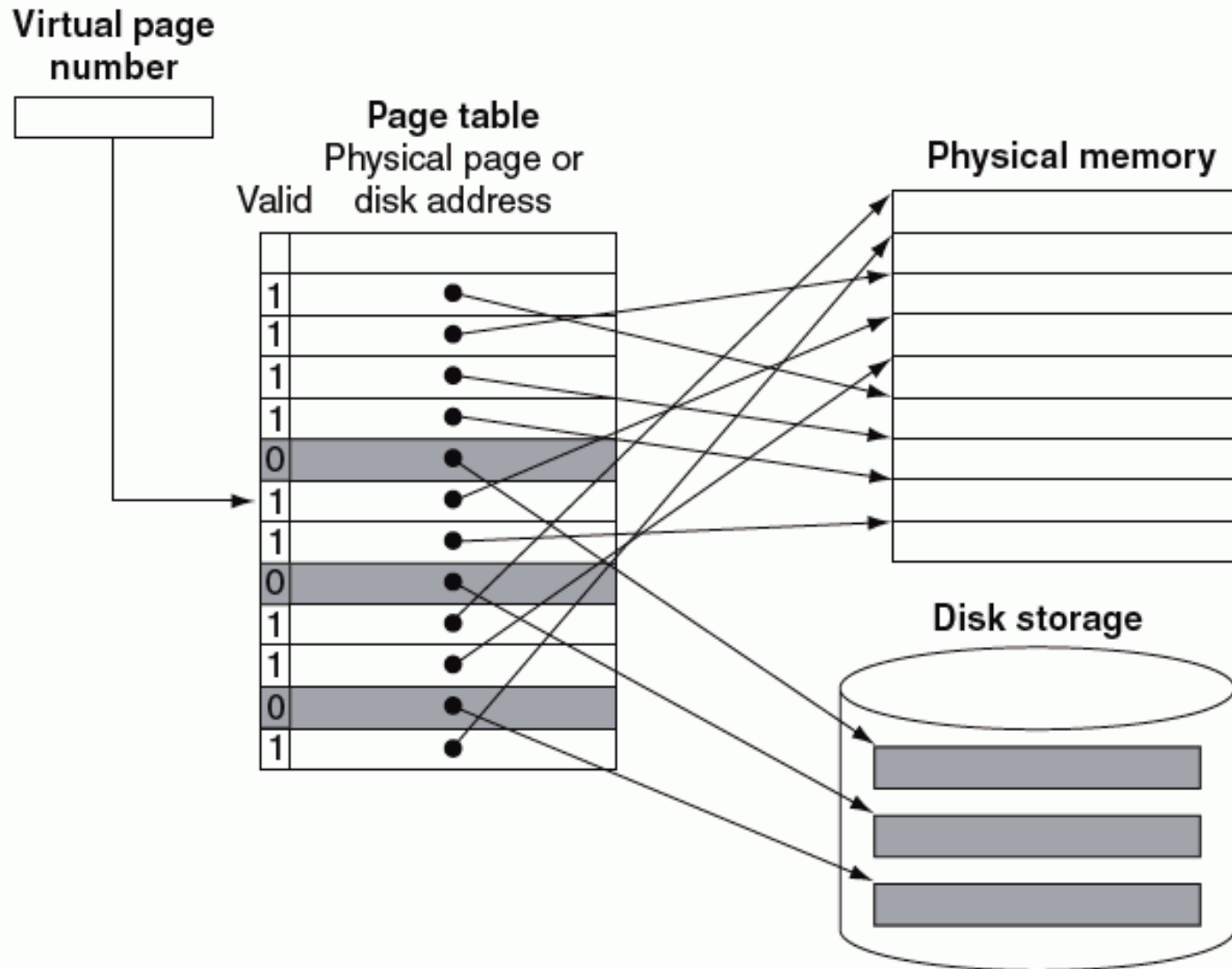
# Memória Virtual



# Memória Virtual

- Falta de página
- Se o bit de validade para uma página virtual estiver desligado, ocorre uma falta de página.
- O S.O. Precisa receber o controle, que é feito pelo mecanismo de exceção.
- Então o S.O. Precisa encontrar a página no próximo nível da hierarquia e decidir onde colocar a página requisitada na memória principal
- É necessário monitorar o local no disco de cada página em um espaço de endereçamento virtual.
- O S.O. Normalmente cria o espaço no disco para todas as páginas de um processo no momento em que ele cria o processo.
  - Este espaço é chamado de área de swap.
- O S.O. Também cria uma estrutura para armazenar onde cada página virtual está armazenada no disco.
  - Essa estrutura pode ser parte da tabela de páginas ou pode ser uma estrutura de dados auxiliar

# Memória Virtual



# Memória Virtual

- O S.O. Também cria uma estrutura para controlar quais processos e quais endereços virtuais usam cada página física.
  - Quando ocorre uma falta de página, se todos os quadros da memória principal estiverem em uso, o SO precisa escolher uma página para substituir.
  - A maioria dos S.Os tenta escolher uma página que supostamente não seja necessária no futuro próximo.
    - Os S.O podem seguir a política LRU (Last Recently Used).
- As páginas substituídas são armazenadas na área de swap do armazenamento secundário.
- Implementar um esquema LRU completamente preciso é muito caro, pois requer atualizar uma estrutura de dados à cada referência de memória.
  - A maioria dos SOs aproxima a LRU monitorando que páginas foram e que páginas não foram usadas recentemente.
  - Alguns computadores fornecem o bit de referência ou bit de uso.
    - É ligado sempre que uma página é acessada.
    - O SO limpa periodicamente os bits de referência e, depois, os registra para que ele possa determinar que páginas foram tocadas durante um determinado período.

# Memória Virtual

- Com um endereço virtual de 32 bits, páginas de 4KB e 4 bytes por entrada da tabela de páginas, podemos calcular o tamanho total da tabela de páginas:

$$\text{Number of page table entries} = \frac{2^{32}}{2^{12}} = 2^{20}$$

$$\text{Size of page table} = 2^{20} \text{ page table entries} \times 2^2 \frac{\text{bytes}}{\text{page table entry}} = 4 \text{ MB}$$

- Precisaríamos usar 4MB de memória para cada programa em execução em um dado momento.
  - Imagine agora um computador com dezenas ou centenas de programas ativos

# Memória Virtual

- Diversas técnicas são usadas para reduzir a quantidade de armazenamento necessária para a tabela de páginas
  - Manter um registrador de limite que restrinja o tamanho da tabela de páginas para um determinado processo.
    - O espaço de endereçamento se expande apenas em uma direção
  - Manter duas tabelas de páginas com e dois registradores limite, dividindo o espaço de endereçamento em dois seguimentos.
    - Isso permite expandir o espaço de endereçamento nas duas direções
  - Os dois esquemas acima, têm a desvantagem de não funcionar bem quando o espaço de endereçamento é usado de uma maneira esparsa.
  - Aplicar uma função de hashing no endereço virtual de modo que a estrutura de dados da tabela de páginas precise ser apenas do tamanho do número de páginas físicas na memória principal.
  - Utilizar múltiplos níveis de tabelas de páginas
    - São úteis com espaços de endereçamento muito grandes e em sistemas que exigem alocação não contígua.

# Memória Virtual

- Por fim, para reduzir a memória principal real consumida pelas tabelas de páginas:
  - A maioria dos sistemas modernos permite que as tabelas de páginas sejam paginadas.
    - Permite que as tabelas de páginas residam no espaço de endereçamento virtual
    - Uma série interminável de faltas de páginas precisa ser evitada,
      - o que pode ser evitado colocando todas as tabelas de páginas no espaço de endereçamento do SO
      - e colocando pelo menos algumas das tabelas de páginas para o sistema em uma parte da memória que é fisicamente endereçada.

# Memória Virtual

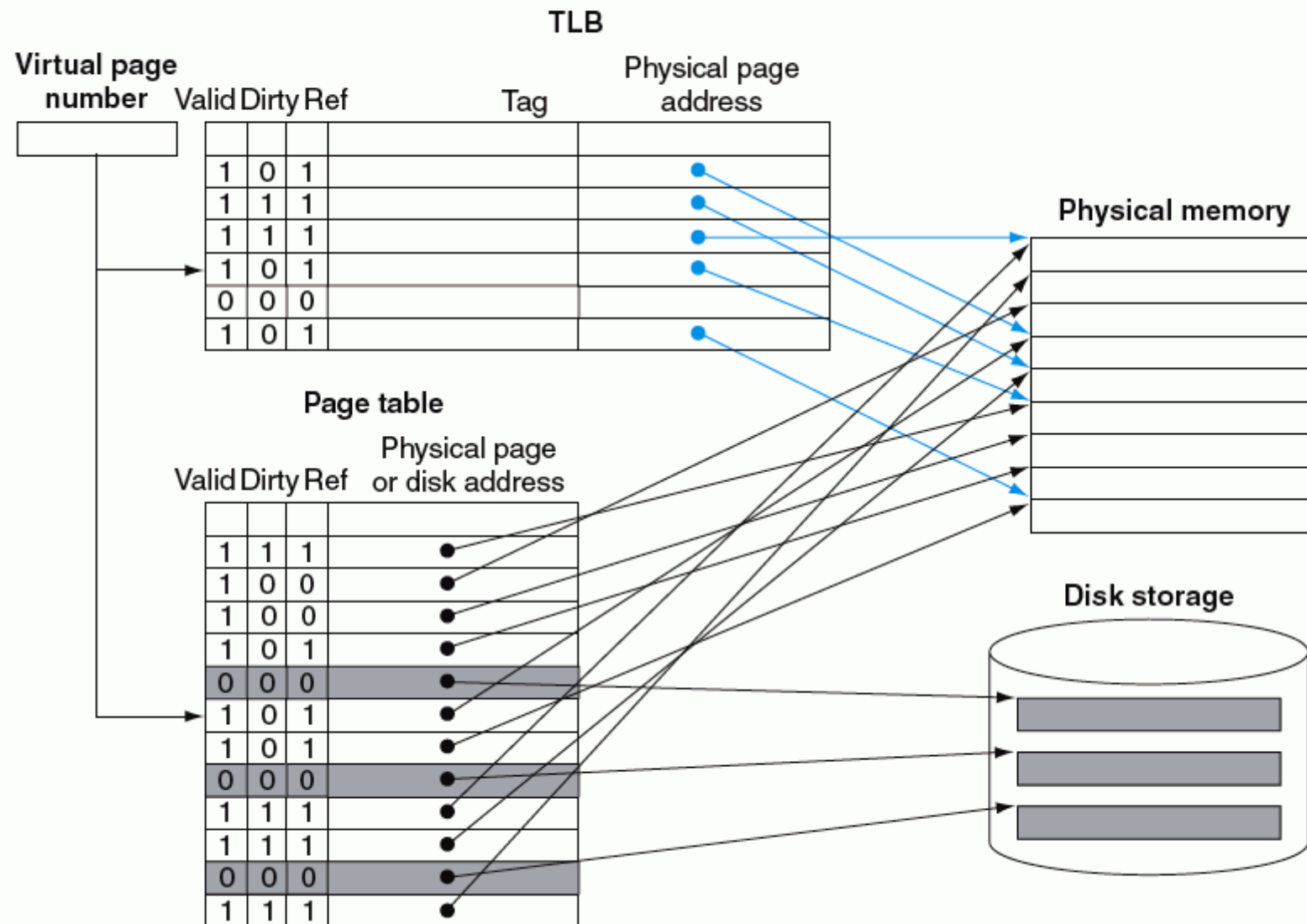
- Escritas
  - Utiliza esquemas write-back
  - O tempo de transferência de disco é pequeno quando comparado ao seu tempo de acesso.
    - Copiar de volta uma página inteira é muito mais eficiente do que escrever words individuais.
  - Um bit de modificação é ligado quando qualquer word de uma página é modificada.
    - Se a página necessitar ser substituída, o SO verifica o bit para decidir se deve escrevê-la no disco antes de substituí-la.
    -



# Memória Virtual

- TLB – Translation-lookaside buffer
  - Funciona como uma Cache para tabelas de páginas.
  - Sem a TLB, cada acesso à memória pode levar, no mínimo o dobro do tempo, devido ao acesso à tabela de páginas.
  - A TLB se baseia na localidade de referência à tabela de páginas
    - Quando uma tradução para um número de página virtual é usada, ela provavelmente será necessária novamente no futuro próximo
      - Pois as referências às words nessa página possuem localidade temporal e espacial.
  -

# Memória Virtual



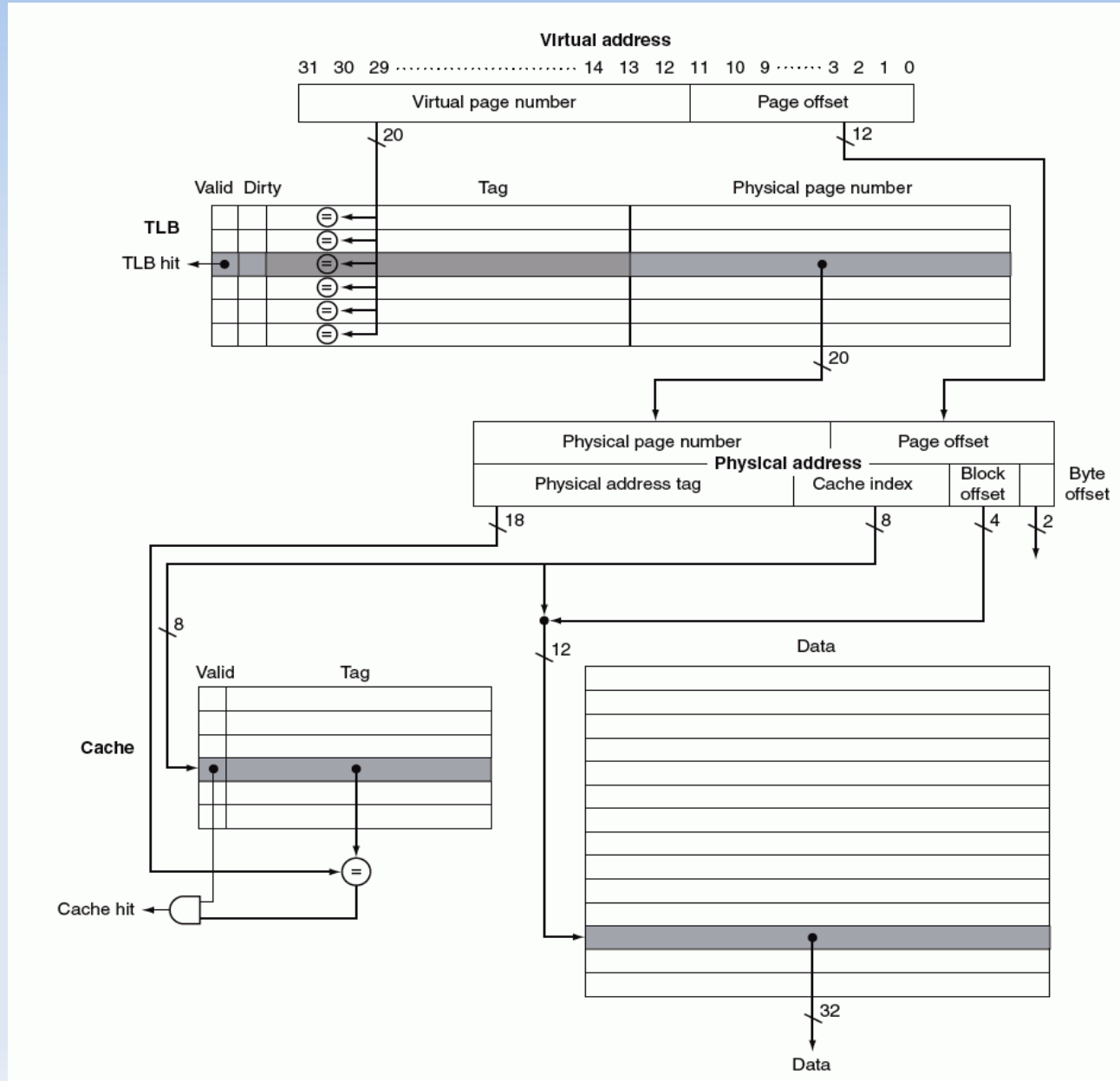
# Memória Virtual

- Em cada referência consultamos o número da página virtual na TLB. Se tivermos um acerto, o número de página física é usado e o bit de referência é ligado.
  - Se o processador estiver realizando uma escrita, o bit de modificação também é ligado.
  - Se ocorrer uma falha na TLB, precisamos determinar se ela é uma falta de páginas ou simplesmente uma falha de TLB.
  - Se a página existir na memória então a falha indica apenas que a tradução está faltando na TLB.
  - Se a página não estiver na memória, então houve uma falta de página e neste caso o processador chama o SO usando uma exceção
  - Depois que em uma falha, a tradução tenha sido recuperada da tabela de páginas, precisamos selecionar uma entrada da TLB para substituir e copiar os bits de referência e de modificação de volta para a entrada da tabela de páginas.

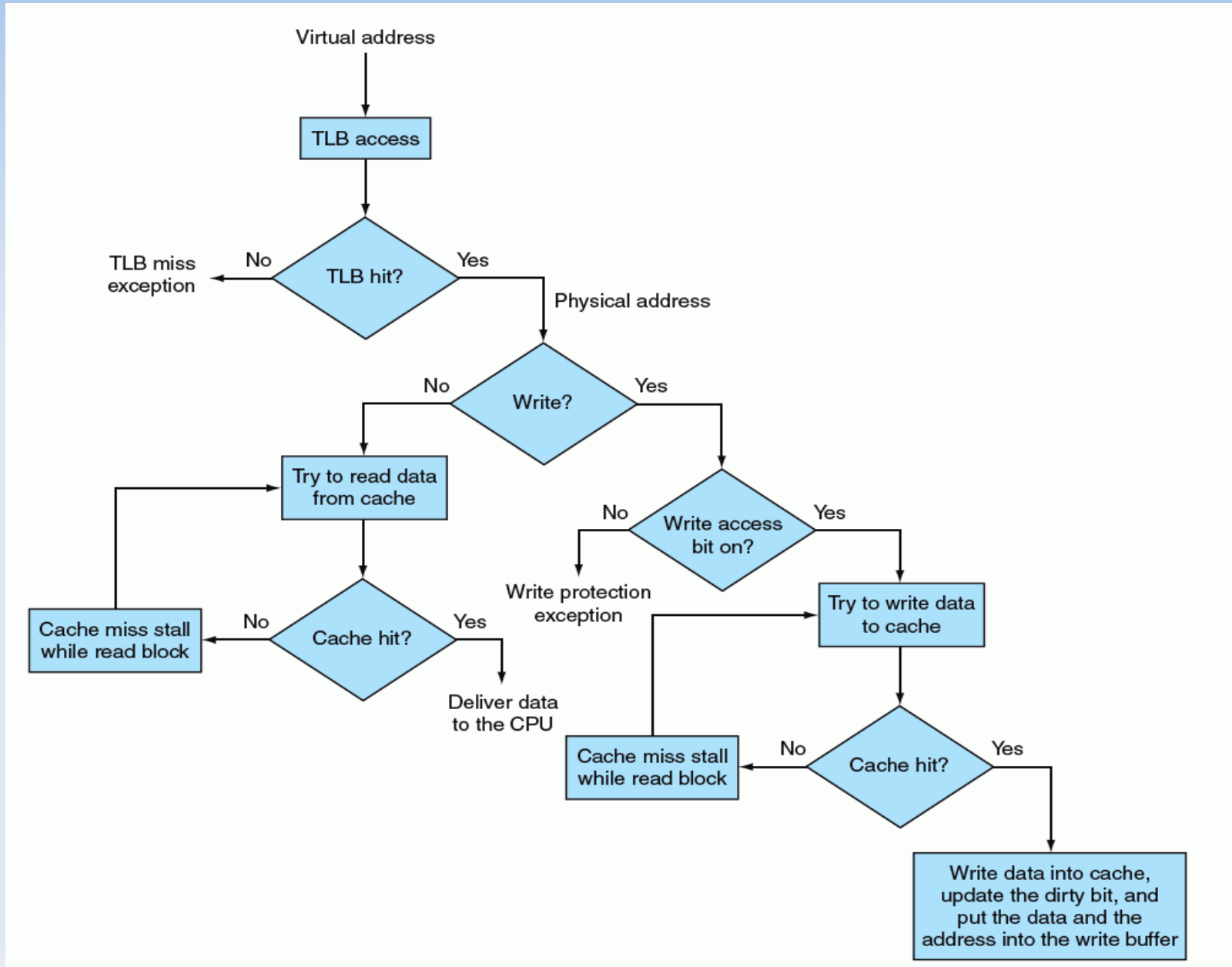
# Memória Virtual

- Alguns valores comuns para uma TLB poderiam ser:
  - Tamanho da TLB: 16 a 512 entradas
  - Tamanho do bloco: 1 a 2 entradas
  - Tempo de acerto: 0,5 a 1 ciclo de clock
  - Penalidade de falha: 10 a 100 ciclos de clock
  - Taxa de falhas: 0.01% a 1%
  -
- Alguns sistemas usam TLBs pequenas e totalmente associativas
- Outros usam TLBs grandes com baixa associatividade.

# A TLB do Intrinsity FastMATH



# A TLB do Intrinsity FastMATH



# Memória virtual

- Integrando memória virtual, TLBs e caches
  - Os dados não podem estar na cache a menos que estejam presentes na memória principal.
  - O SO cuida de remover o conteúdo de qualquer página da cache quando decide migrar esta página para o disco. Ao mesmo tempo, ele modifica as tabelas de páginas e a TLB.
  - Sob circunstâncias ideais um endereço virtual é traduzido pela TLB e enviado para a cache onde os dados apropriados são encontrados, recuperados e desvolvidos ao processador.
  - No pior caso, uma referência pode falhar em todos os três componentes: TLB, tabela de páginas e a cache.
  -

# Memória virtual

TLB	Page table	Cache	Possible? If so, under what circumstance?
hit	hit	miss	Possible, although the page table is never really checked if TLB hits.
miss	hit	hit	TLB misses, but entry found in page table; after retry, data is found in cache.
miss	hit	miss	TLB misses, but entry found in page table; after retry, data misses in cache.
miss	miss	miss	TLB misses and is followed by a page fault; after retry, data must miss in cache.
hit	miss	miss	Impossible: cannot have a translation in TLB if page is not present in memory.
hit	miss	hit	Impossible: cannot have a translation in TLB if page is not present in memory.
miss	miss	hit	Impossible: data cannot be allowed in cache if the page is not in memory.

**FIGURE 7.26 The possible combinations of events in the TLB, virtual memory system, and cache.** Three of these combinations are impossible, and one is possible (TLB hit, virtual memory hit, cache miss) but never detected.



# Memória virtual

- Cache fisicamente indexada e fisicamente rotulada
  - Tanto o índice quanto a tag de cache são endereços físicos em vez de virtuais.
- Cache virtualmente indexada e virtualmente rotulada.
  - A cache é acessada com endereço virtual que não foi traduzido para um endereço físico.
  - Isso tira a TLB do caminho crítico, reduzindo a latência da cache.
- Cache virtualmente indexada, mas fisicamente rotulada.
  - Utilizam apenas a parte do offset da página do endereço, que é físico para indexação, mas usam tags físicas.