

# Ch. 08 – Virtual Memory

## 8.1 – Hardware and Control Structure

### 8.1.1 Locality and Virtual Memory

### 8.1.2 Simple Paging

### 8.1.3 Simple Segmentation

### 8.1.4 Combined Paging and Segmentation

### 8.1.5 Protection and Sharing

## 8.2 – Operating System Software

### 8.2.1 Fetch Policy

### 8.2.2 Placement Policy

### 8.2.3 Replacement Set Management

### 8.2.4 Cleaning Policy ... .. (continue)

# Ch. 08 – Virtual Memory

## 8.2 – Operating System Software

### 8.2.3 ... (continue)

### 8.2.4 Cleaning Policy

### 8.2.5 Load Control

## 8.3 UNIX and Solaris Memory Management (LEITURA COMPLEMENTAR)

### 8.3.1 Paging System

### 8.3.2 Kernel Memory Allocator

## 8.4 LINUX Memory Management (LEITURA COMPLEMENTAR)

### 8.4.1 LINUX Virtual Memory

### 8.4.2 Kernel Memory Allocation

# Referências Bibliográficas

- Operating Systems – Internals and Design Principles. William Stallings. 7<sup>th</sup>, Prentice-Hall 2012.
- Instructor Resources – Operating Systems - 7th  
<http://williamstallings.com/OperatingSystems/OS7e-Instructor/>

You're gonna need a bigger boat. – Steven Spielberg, J AWS , 1975

## 8 – Virtual Memory / 8 – Virtual Memory (overview)

### 8 – Virtual Memory (overview)

- **“paging and segmentation”** .. capítulo anterior introduziu os conceitos de paginação e segmentação e analisou suas deficiências.
- **“virtual memory”** .. requer um pouco mais de análise pelo fato de que o gerenciamento de memória contempla aspectos da relação complexa entre o “hardware” do processador e o “software” do sist. oper.
- .. inicialmente vamos nos ater ao aspecto de “hardware” da memória virtual, observando o uso de paginação, segmentação e combinação de paginação e segmentação.
- .. na sequência, examina-se os problemas envolvidos no projeto de uma instalação de memória virtual em sistemas operacionais.

## 8 – Virtual Memory / 8 – Virtual Memory (overview)

### ... 8 – Virtual Memory (overview)

- “**key terms**” .. relacionados a memória virtual.
- “**virtual memory**” .. alocação de memória no qual a memória secundária pode ser endereçada como se fizesse parte da memória principal.
- “**virtual address**” .. endereço atribuído a um local na memória virtual para permitir que esse local seja acessado como se fizesse parte da memória principal, ou seja, uma extensão da memória principal.

<b>Virtual memory</b>	A storage allocation scheme in which secondary memory can be addressed as though it were part of main memory.
<b>Virtual address</b>	The address assigned to a location in virtual memory to allow that location to be accessed as though it were part of main memory.
<b>Virtual address space</b>	The virtual storage assigned to a process.
<b>Address space</b>	The range of memory addresses available to a process.
<b>Real address</b>	The address of a storage location in main memory.

## 8 – Virtual Memory / 8 – Virtual Memory (overview) ... 8 – Virtual Memory (overview)

- “**key terms**” .. relacionados a memória virtual.
- “**virtual address space**” .. memória virtual atribuído a um processo.
- “**address space**” .. faixa de endereços de memória disponíveis para um processo, ou seja, para código, dados e pilha.
- “**real address**” .. endereço de uma posição na memória principal, ou seja, endereço físico considerando o espaço real disponível na memória física instalada no sistema computacional.

<b>Virtual memory</b>	A storage allocation scheme in which secondary memory can be addressed as though it were part of main memory.
<b>Virtual address</b>	The address assigned to a location in virtual memory to allow that location to be accessed as though it were part of main memory.
<b>Virtual address space</b>	The virtual storage assigned to a process.
<b>Address space</b>	The range of memory addresses available to a process.
<b>Real address</b>	The address of a storage location in main memory.

## 8 – Virtual Memory / 8.1 – Hardware and Control Structure

### 8.1 – Hardware and Control Structure

- “**comparing**” .. processos de paginação e segmentação assim como o particionamento fixo e dinâmico, estabelecem os princípios básicos para o gerenciamento de memória:
- .. todas as referências dentro de um processo são endereços lógicos que são dinamicamente traduzidos para endereços físicos em tempo de execução - “**process at run time**”.
- .. um processo pode ser dividido em várias partes, sejam páginas ou segmentos e, essas partes não estão necessariamente localizadas de forma contígua na memória principal durante a execução.
- “**premissa**” .. com os 02 princípios presentes, então não é necessário que todas as páginas ou todos os segmentos de um processo estejam na memória durante a execução, entretanto:
- ... como todo este procedimento pode ser posto em prática ?!

## 8 – Virtual Memory / 8.1 – Hardware and Control Structure

### ... 8.1 – Hardware and Control Structure

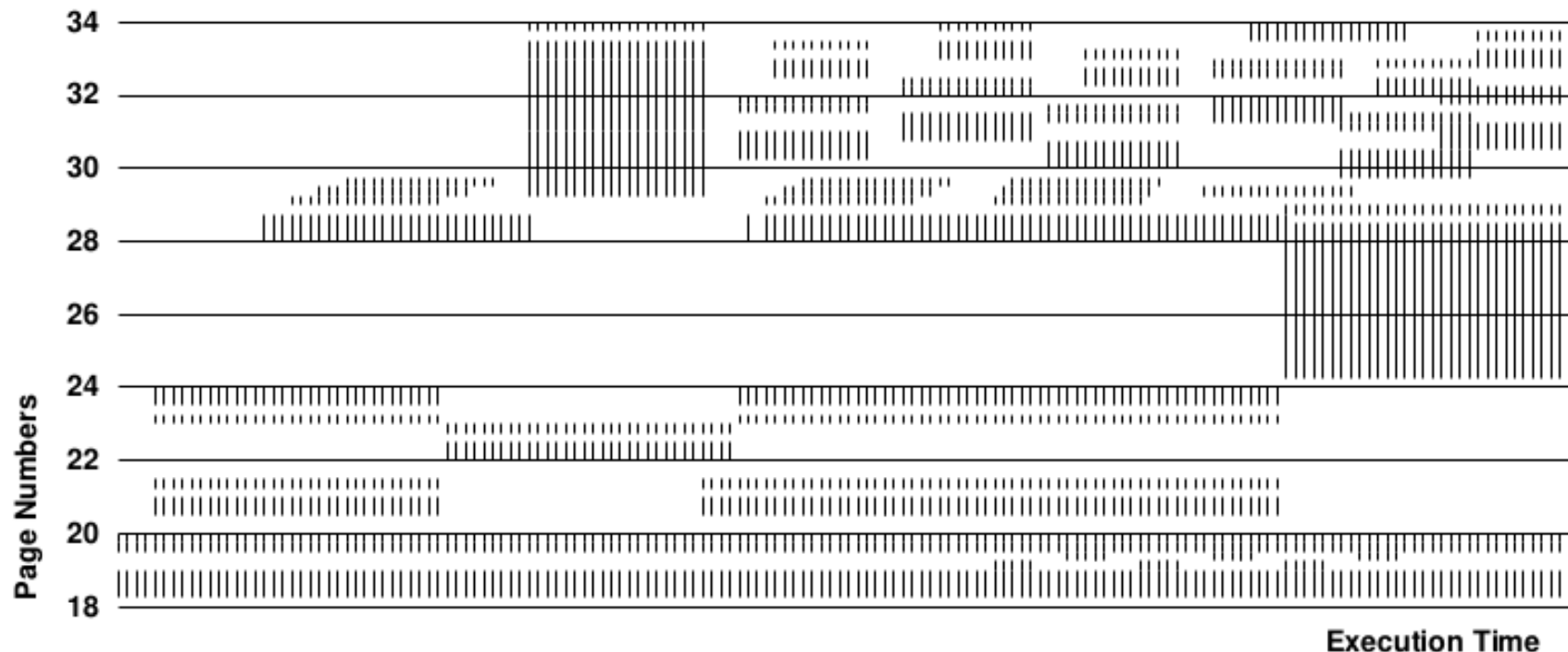
- “**implications**” .. que conduzem a melhora de performance ..
- “**more processes may be maintained in main memory**” .. mais processos podem ser mantidos na memória principal, posto que apenas algumas partes de cada processo são carregados.
- “**process may be larger than all of main memory**” .. um processo pode ser maior que toda a memória principal, não obstante, o mesmo poderá ser carregado para a memória a fim de ser executado.
- .. pelo fato do processo executar na memória principal, esta memória é referenciada como “**real memory**”, enquanto que os programadores ou usuários trabalham com uma memória maior – “**virtual memory**”.
- “**vantagem**” .. proporciona uma multiprogramação efetiva e libera os usuários das preocupações acerca da memória principal.



## 8 – Virtual Memory / 8.1 – Hardware and Control Structure

### 8.1.1 – Locality and Virtual Memory

- Quanto a pergunta se esta abordagem é prática, experiências com sist. oper. demonstram sem dúvida que memória virtual funciona.
- “**principle of locality**” .. muitas das justificativas são baseada na crença do princípio da localidade que estabelece que as referências as instruções e dados tendem a ser agrupadas.



## 8 – Virtual Memory / 8.1 – Hardware and Control Structure

### ... 8.1.1 – Locality and Virtual Memory

- **“practical scheme”** .. exige suporte de “hardware” e “software” do sist. oper. para o esquema de paginação e/ou segmentação.
- .. tabela resume as características de paginação e segmentação, com e sem o uso de memória virtual.

Simple Paging	Virtual Memory Paging
Main memory partitioned into small fixed-size chunks called frames.	Main memory partitioned into small fixed-size chunks called frames.
Program broken into pages by the compiler or memory management system.	Program broken into pages by the compiler or memory management system.
Internal fragmentation within frames.	Internal fragmentation within frames.
No external fragmentation.	No external fragmentation.
Operating system must maintain a page table for each process showing which frame each page is in.	Operating system must maintain a page table for each process showing which frame each page is in.
Operating system must maintain a free frame list.	Operating system must maintain a free frame list.
Processor uses page number, offset to calculate absolute address.	Processor uses page number, offset to calculate absolute address.
All the pages of a process must be in main memory for process to run, unless overlays are used.	Not all pages of a process need be in main memory frames for the process to run. Pages may be read in as need.
	Reading a page into main memory may require writing a page out to disk.

## 8 – Virtual Memory / 8.1 – Hardware and Control Structure

### ... 8.1.1 – Locality and Virtual Memory

- “**practical scheme**” .. exige suporte de “hardware” e “software” do sist. oper. para o esquema de paginação e/ou segmentação.
- .. tabela resume as características de paginação e segmentação, com e sem o uso de memória virtual.

Simple Segmentation	Virtual Memory Segmentation
Main memory not partitioned.	Main memory not partitioned.
Program segments specified by the programmer to the compiler (i.e., the decision is made by the programmer)	Program segments specified by the programmer to the compiler (i.e., the decision is made by the programmer).
No internal fragmentation.	No internal fragmentation.
External fragmentation.	External fragmentation.
Operating system must maintain a segment table for each process showing the load address and length of each segment.	Operating system must maintain a segment table for each process showing the load address and length of each segment.
Operating system must maintain a list of free holes in main memory.	Operating system must maintain a list of free holes in main memory.
Processor uses segment number, offset to calculate absolute address.	Processor uses segment number, offset to calculate absolute address.
All the segments of a process must be in main memory for process to run, unless overlays are used.	Not all segments of a process need be in main memory frames for the process to run. Segments may be read in as need.
	Reading a segment into main memory may require writing one or more segments out to disk.

# 8 – Virtual Memory / 8.1 – Hardware and Control Structure

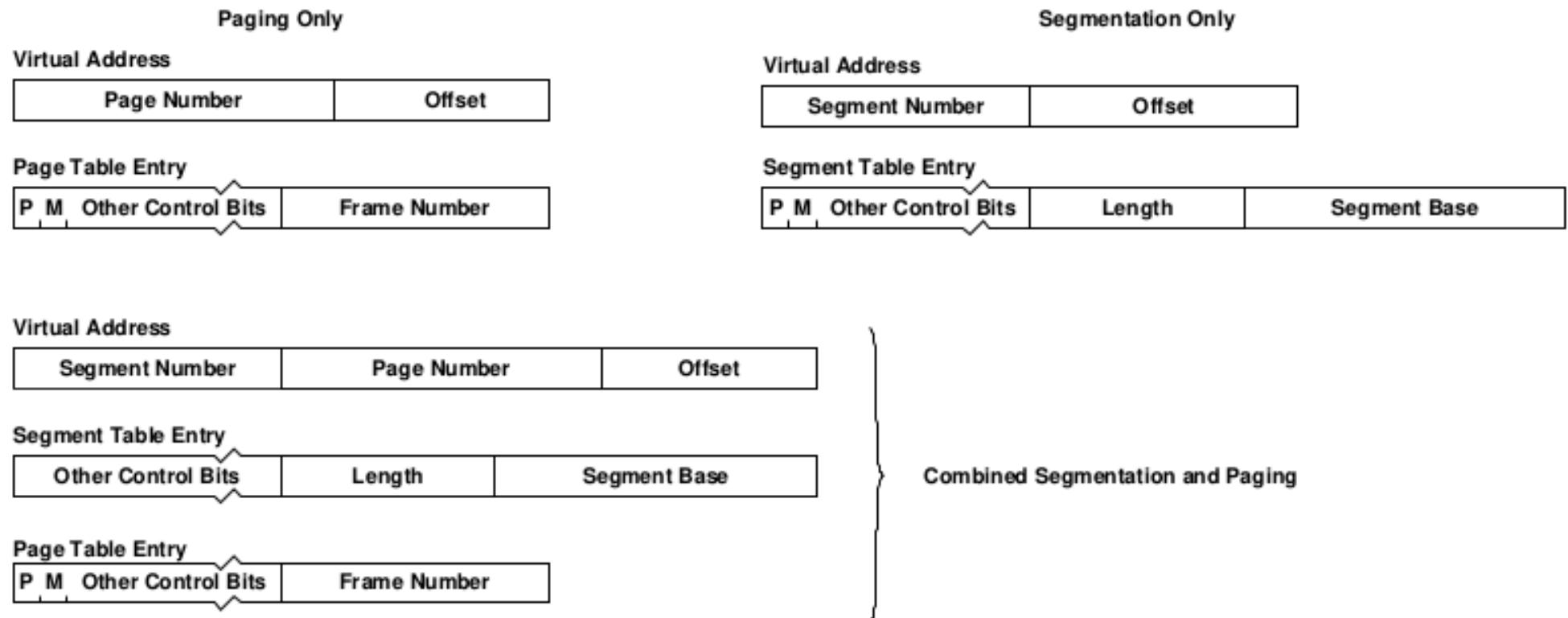
## ... 8.1.1 – Locality and Virtual Memory

Simple Paging	Virtual Memory Paging	Simple Segmentation	Virtual Memory Segmentation
Main memory partitioned into small fixed-size chunks called frames.	Main memory partitioned into small fixed-size chunks called frames.	Main memory not partitioned.	Main memory not partitioned.
Program broken into pages by the compiler or memory management system.	Program broken into pages by the compiler or memory management system.	Program segments specified by the programmer to the compiler (i.e., the decision is made by the programmer)	Program segments specified by the programmer to the compiler (i.e., the decision is made by the programmer).
Internal fragmentation within frames.	Internal fragmentation within frames.	No internal fragmentation.	No internal fragmentation.
No external fragmentation.	No external fragmentation.	External fragmentation.	External fragmentation.
Operating system must maintain a page table for each process showing which frame each page is in.	Operating system must maintain a page table for each process showing which frame each page is in.	Operating system must maintain a segment table for each process showing the load address and length of each segment.	Operating system must maintain a segment table for each process showing the load address and length of each segment.
Operating system must maintain a free frame list.	Operating system must maintain a free frame list.	Operating system must maintain a list of free holes in main memory.	Operating system must maintain a list of free holes in main memory.
Processor uses page number, offset to calculate absolute address.	Processor uses page number, offset to calculate absolute address.	Processor uses segment number, offset to calculate absolute address.	Processor uses segment number, offset to calculate absolute address.
All the pages of a process must be in main memory for process to run, unless overlays are used.	Not all pages of a process need be in main memory frames for the process to run. Pages may be read in as need.	All the segments of a process must be in main memory for process to run, unless overlays are used.	Not all segments of a process need be in main memory frames for the process to run. Segments may be read in as need.
	Reading a page into main memory may require writing a page out to disk.		Reading a segment into main memory may require writing one or more segments out to disk.

## 8 – Virtual Memory / 8.1 – Hardware and Control Structure

### 8.1.2 – Simple Paging

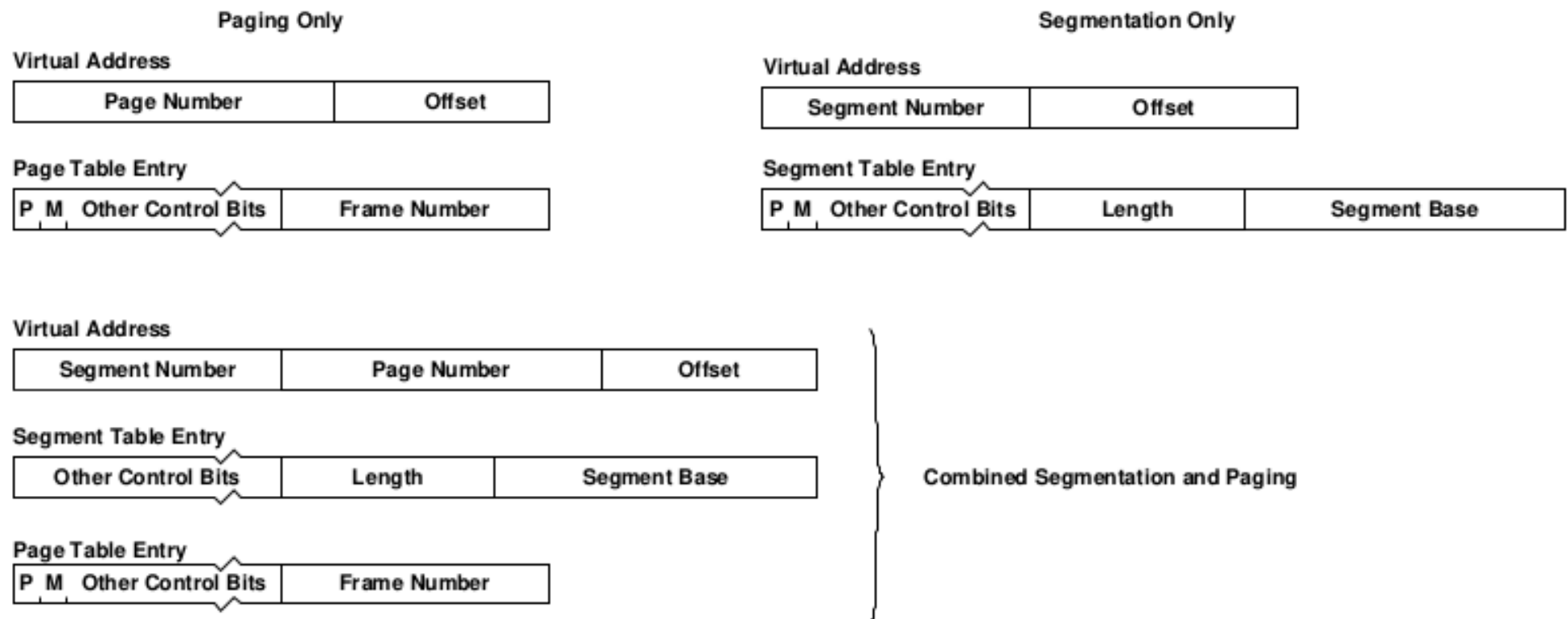
- “**virtual memory**” .. termo geralmente associado a sistemas que empregam a paginação, embora a memória virtual baseada na segmentação também seja usada .. discutida a seguir.



## 8 – Virtual Memory / 8.1 – Hardware and Control Structure

### ... 8.1.2 – Simple Paging

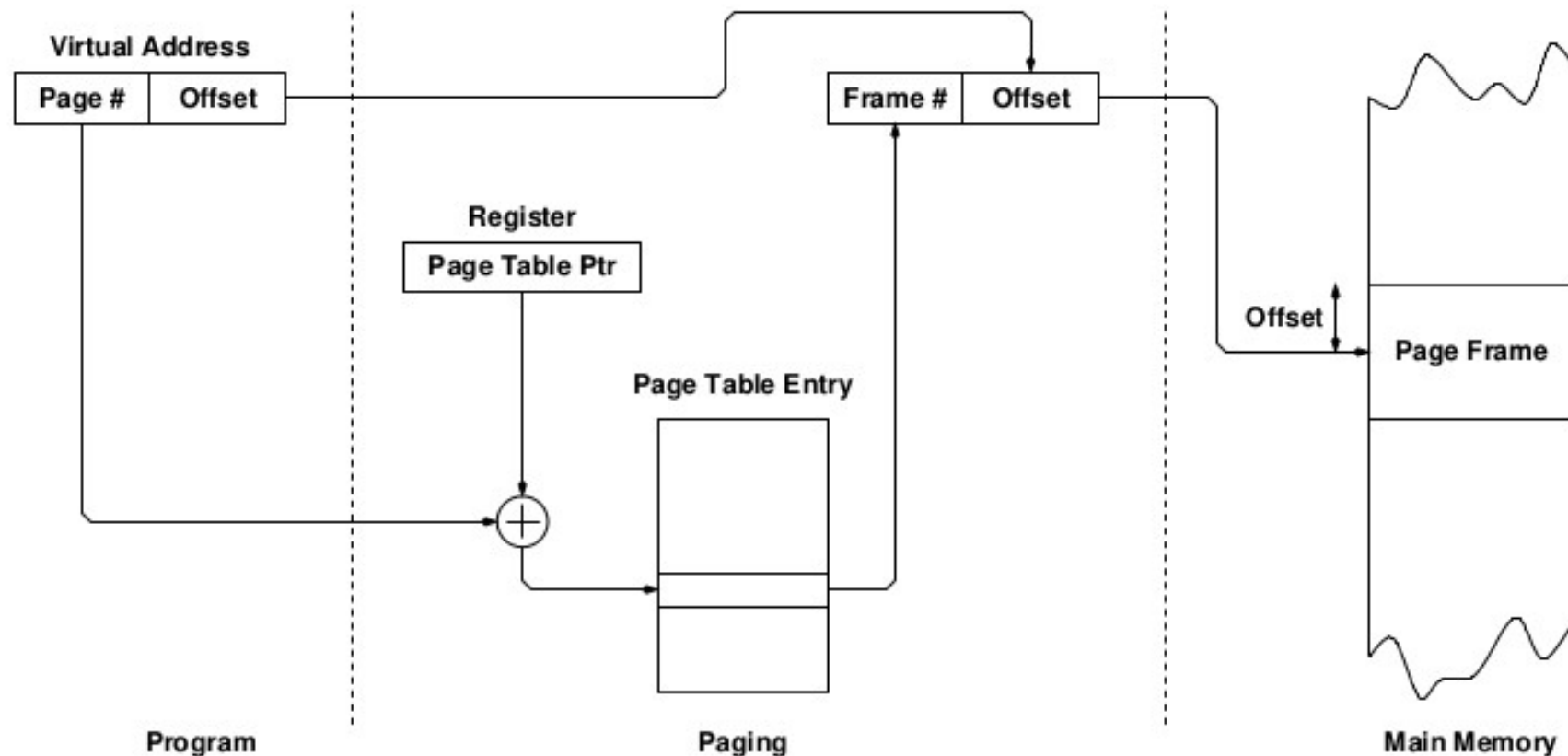
- .. na discussão da paginação simples, cada processo tem sua própria tabela de páginas, ou seja, tabela de página individualizada.
- .. quando todas as suas páginas são carregadas na memória principal, a tabela de páginas é criada e carregada na memória principal.



## 8 – Virtual Memory / 8.1 – Hardware and Control Structure

### ... 8.1.2 – Simple Paging

- .. processo de leitura de uma palavra da memória principal envolve a tradução de endereço virtual, consistindo do nro. da página e offset para endereço físico, consistindo do número do “frame” e offset.



## 8 – Virtual Memory / 8.1 – Hardware and Control Structure

### ... 8.1.2 – Simple Paging

- **“problema”** .. na maioria dos sistemas, há uma tabela de página por processo e, claramente, a quantidade de memória devotada para tabela de páginas pode ser inaceitável.
- e.g., na arquitetura VAX, cada processo pode ter até  $2^{31} = 2$  GB de memória virtual e, caso, utilize páginas de  $2^9$  bytes = 512 bytes » serão necessários  $2^{22}$  entradas por tabela de página por processo.
- .. para exemplificar alguns cálculos assuma uma memória principal de 4 GB ou  $2^{32}$  bytes endereçáveis, com “frames” de 512 bytes ( $2^9$ ), então tem-se  $2^{32} / 2^9 = 2^{23}$  “frames” na memória principal.
- .. como cada entrada da tabela de página de cada processo precisa ter o campo referente ao “frame”, ou seja, 23 bits, tem-se para cada processo uma tabela com  $2^{22}$  entradas, cada uma com 23 bits.



## 8 – Virtual Memory / 8.1 – Hardware and Control Structure

### ... 8.1.2 – Simple Paging

- e.g., na arquitetura VAX, cada processo pode ter até  $2^{31} = 2$  GB de memória virtual e, caso, utilize páginas de  $2^9$  bytes = 512 bytes » serão necessários  $2^{22}$  entradas por tabela de página por processo.
- ..  $2^{22}$  (entradas) \* 23 (bits) = 96468992 bits que convertido em bytes » 12058624 bytes ou 94,98 Mbytes (Mega = 1024).
- “**conclusão**” .. quantidade de memória devotada para tabela de páginas por processo para todos os processos pode ser inaceitável.
- “**solução**” .. armazenar a tabela de páginas na memória secundária em vez de armazená-la na memória principal, o que por outro lado, significa que estão sujeitas às mesmas regras de paginação.

## 8 – Virtual Memory / 8.1 – Hardware and Control Structure

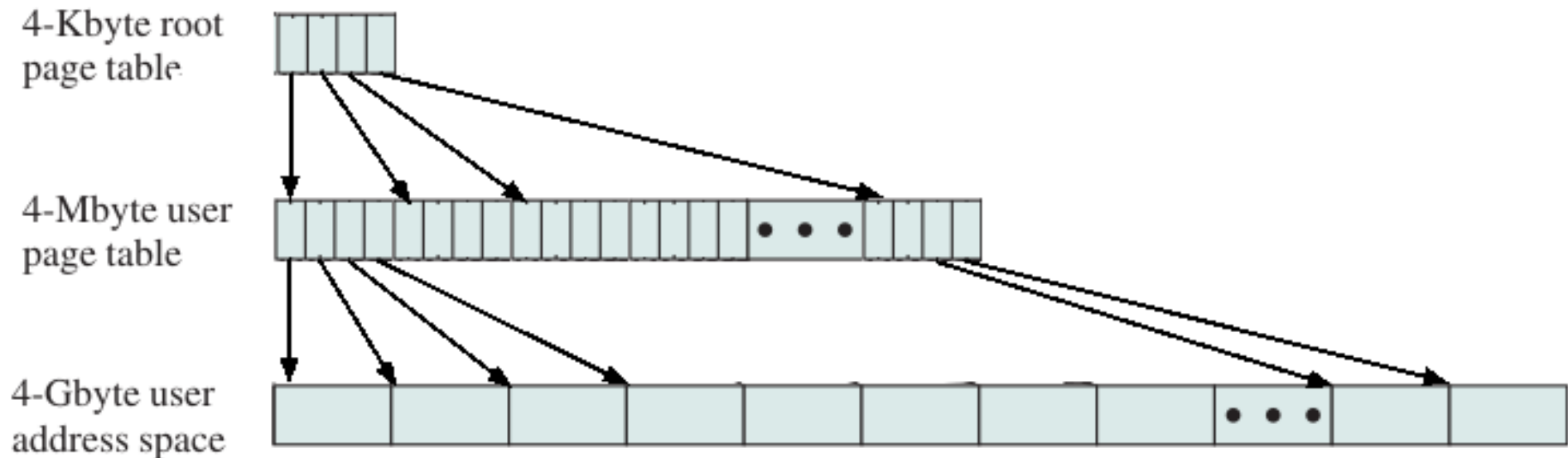
### ... 8.1.2 – Simple Paging

- **“alternativa”** .. uso de 02 níveis na organização da tabela de páginas por processo, neste caso, há um diretório de páginas no qual cada entrada aponta para uma tabela de páginas.
- .. nesse esquema, há um diretório de páginas, no qual cada entrada aponta para uma tabela de páginas.
- .. se o comprimento do diretório de páginas for “x” e se o comprimento máximo de uma tabela de páginas for “y”, um processo pode consistir em até “x \* y” páginas.
- .. normalmente, o comprimento máximo de uma tabela de página é restrito para ser igual ao tamanho de uma página.
- **“observação”** .. esta asserção acima é extremamente importante e, de certo modo, limita ou restringe o tamanho de uma tabela de páginas do ponto de vista prático de implementação.

## 8 – Virtual Memory / 8.1 – Hardware and Control Structure

### ... 8.1.2 – Simple Paging

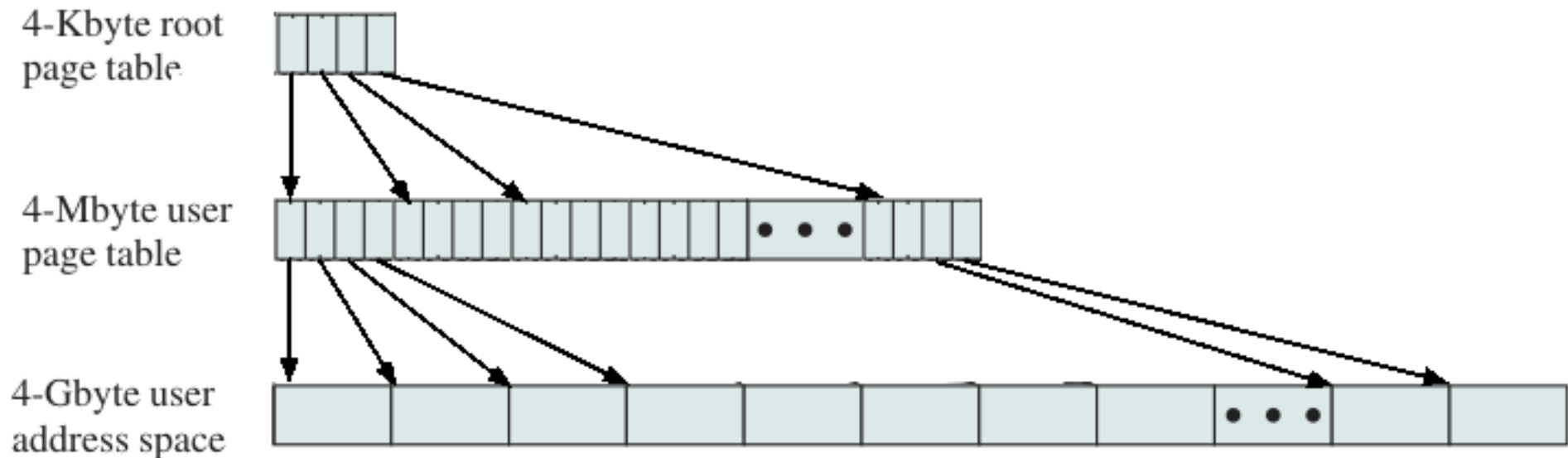
- e.g. seja um esquema de 02 níveis para uso de um endereço de 32 bits, onde cada página tem 4 KB ( $2^{12}$  bytes) e, considerando o espaço de endereçamento de  $2^{32}$  bytes ou 4 GB, tem-se  $2^{20}$  páginas.
- .. se cada uma dessas páginas for mapeada por uma entrada de tabela de página de 4 bytes (32 bits), pode-se criar uma tabela de página composta de  $2^{20}$  PTEs que requerem 4 MB ( $2^{22}$  bytes).



## 8 – Virtual Memory / 8.1 – Hardware and Control Structure

### ... 8.1.2 – Simple Paging

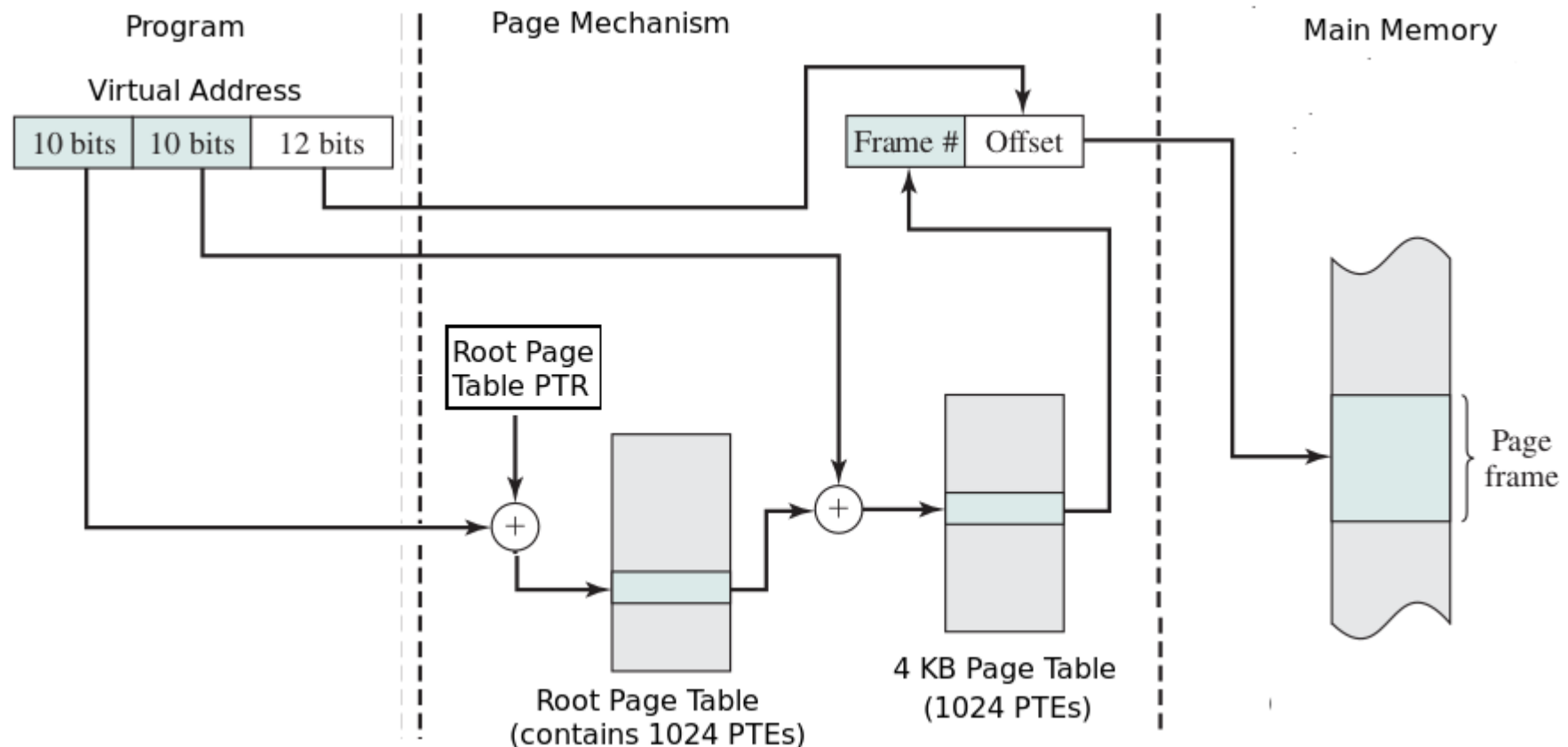
- e.g. seja um esquema de 02 níveis para uso de um endereço de 32 bits, onde cada página tem 4 KB ( $2^{12}$  bytes) e, considerando o espaço de endereçamento de  $2^{32}$  bytes ou 4 GB, tem-se  $2^{20}$  páginas.
- .. trata-se de uma tabela de páginas ocupando  $2^{10}$  páginas que pode ser mantida na memória virtual e mapeada por uma tabela de página raiz com  $2^{10}$  PTEs ocupando 4 KB ( $2^{12}$ ) da memória principal.



## 8 – Virtual Memory / 8.1 – Hardware and Control Structure

### ... 8.1.2 – Simple Paging

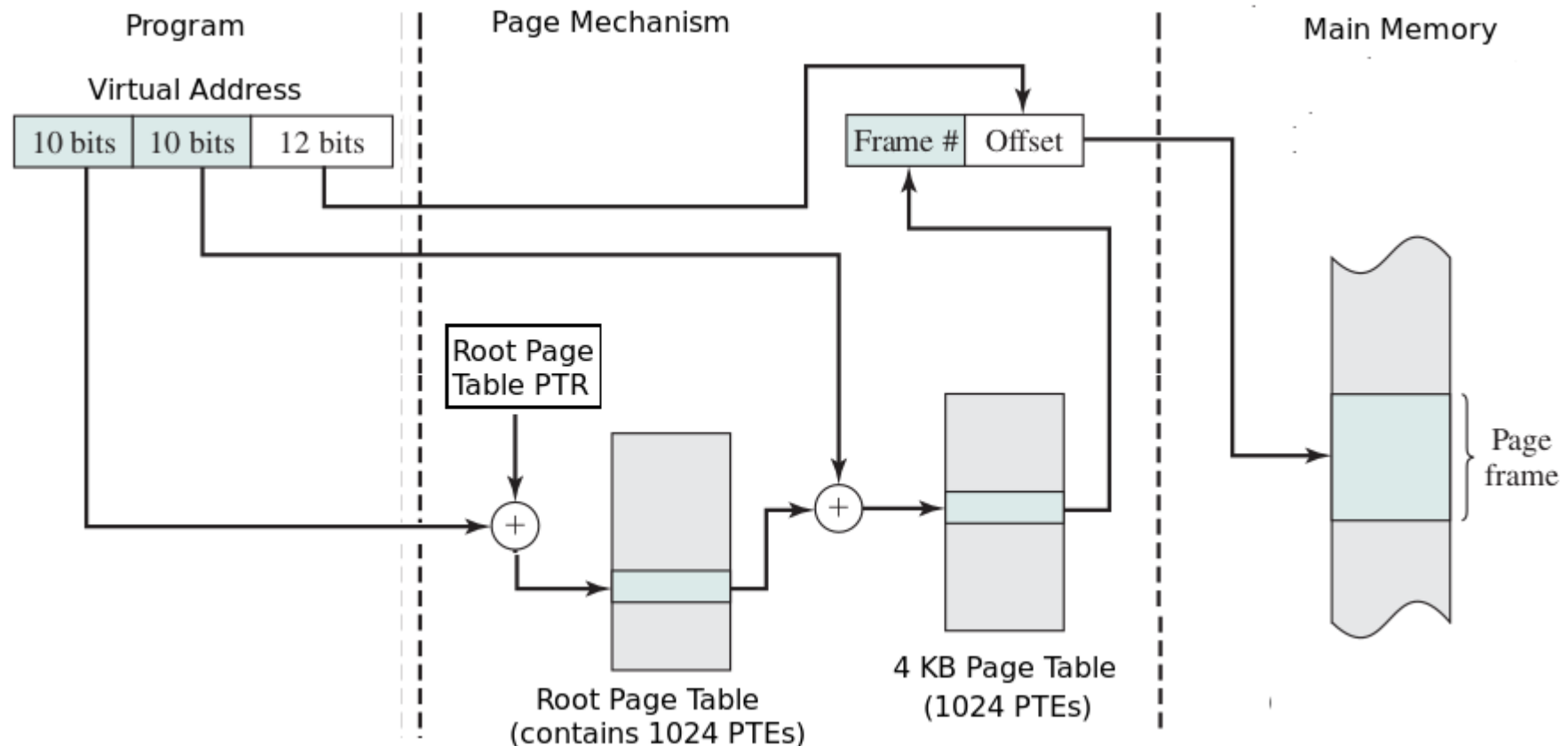
- e.g., seja a representação das etapas envolvidas na tradução de endereços para este exemplo anteriormente descrito, onde a página raiz sempre permanece na memória principal.



## 8 – Virtual Memory / 8.1 – Hardware and Control Structure

### ... 8.1.2 – Simple Paging

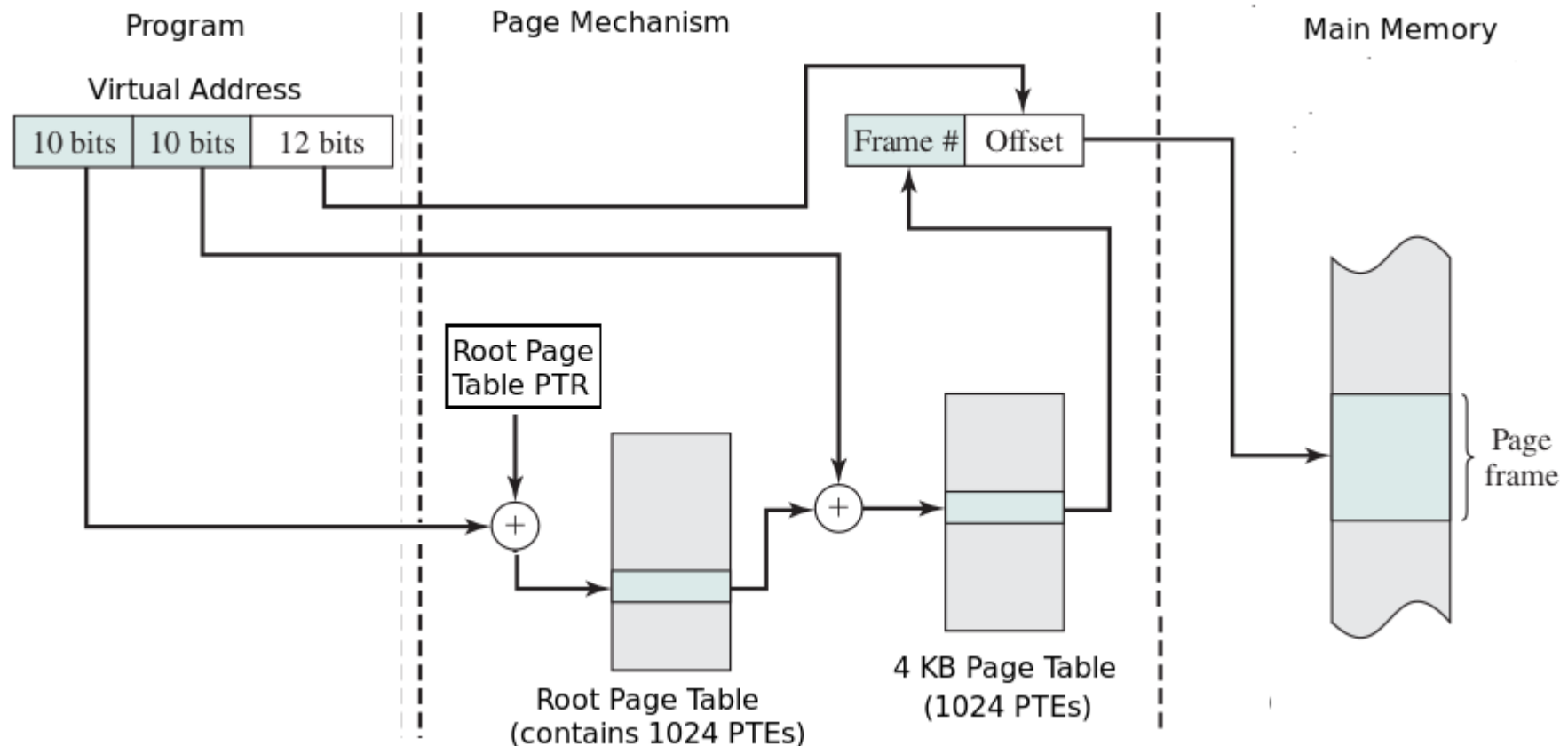
- .. os primeiros 10 bits de um endereço virtual são usados como índice na página raiz de modo que contém um PTE (Page Table Entry) para uma página da tabela de páginas do usuário.



## 8 – Virtual Memory / 8.1 – Hardware and Control Structure

### ... 8.1.2 – Simple Paging

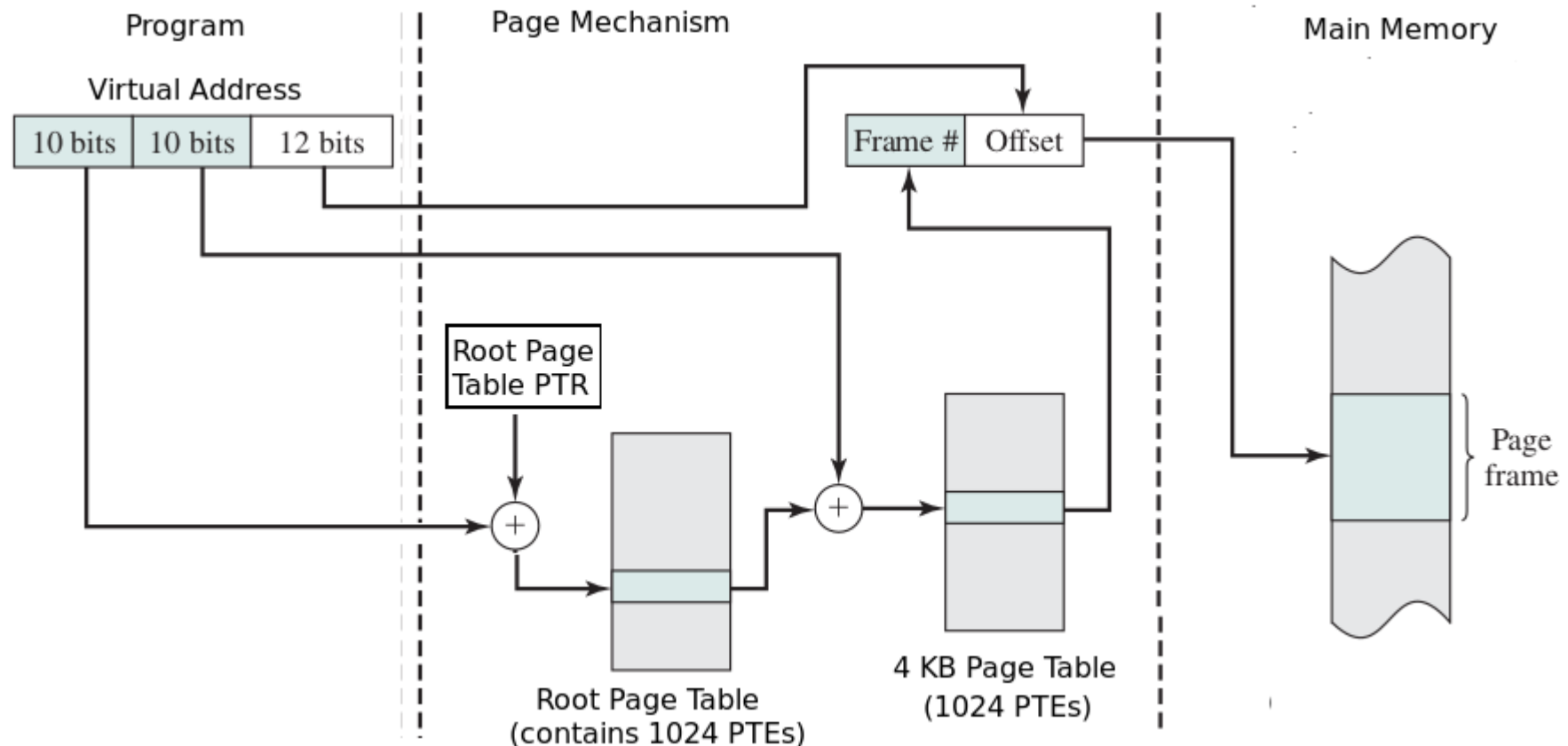
- .. se essa página estiver na memória principal, os próximos 10 bits do endereço virtual indexam na página PTE do usuário para encontrar o PTE para a página que é referenciada pelo endereço virtual.



## 8 – Virtual Memory / 8.1 – Hardware and Control Structure

### ... 8.1.2 – Simple Paging

- .. se essa página não estiver na memória principal, ocorre uma falha de página, gatilhando a busca da página na memória secundária.





## 8 – Virtual Memory / 8.1 – Hardware and Control Structure

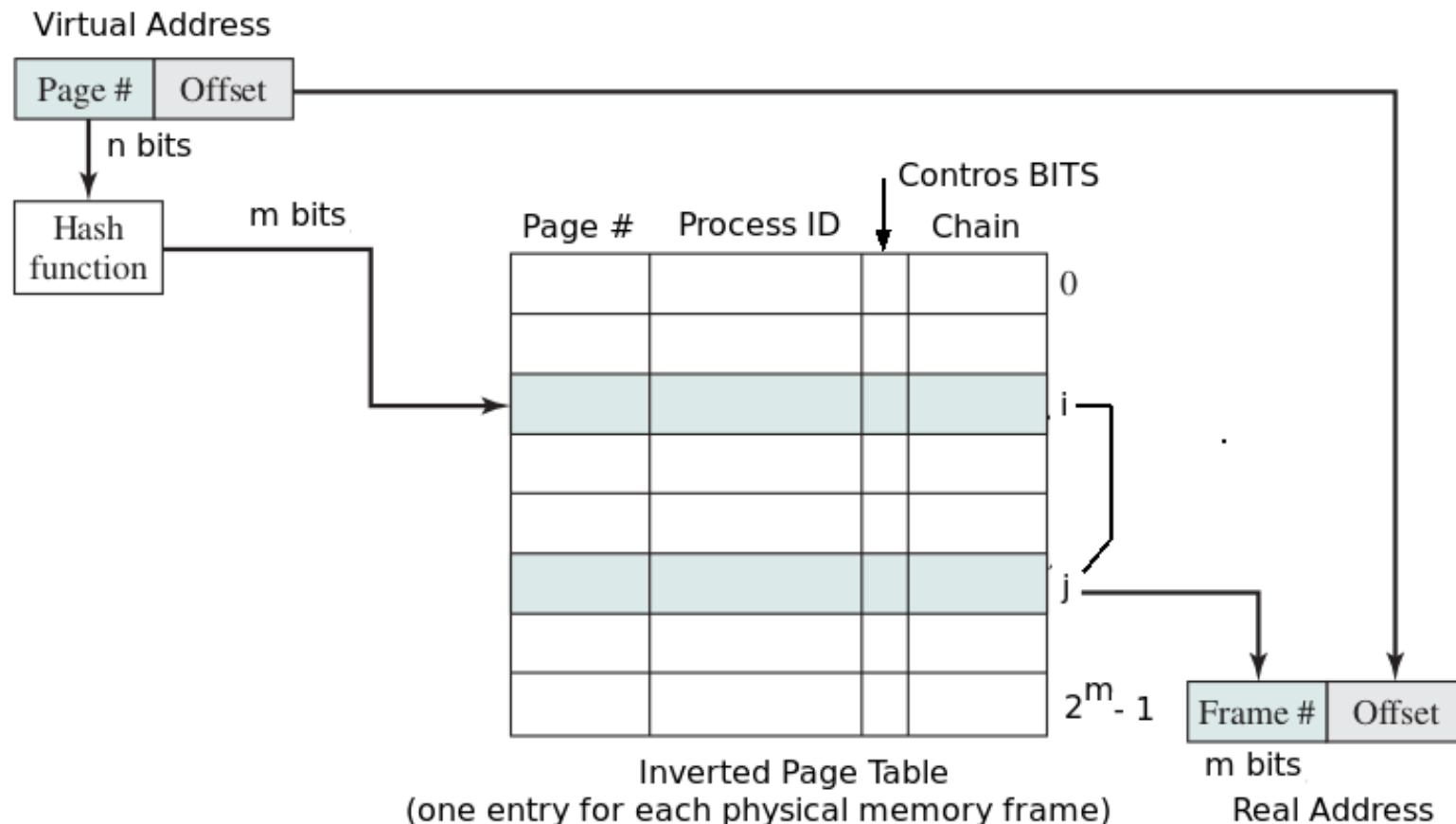
### ... 8.1.2 – Simple Paging

- “**desvantagem**” .. do tipo de tabelas de página já discutido é que seu tamanho é proporcional ao do espaço de endereço virtual.
- “**inverted page table**” .. abordagem alternativa para o uso de tabelas de página de um ou vários níveis é o uso de uma estrutura da “**tabela de página invertida**”.
- .. variações nesta abordagem são usadas na arquitetura PowerPC, UltraSPARC, IA-64 e em implementações do sistema operacional Mach no RT-PC utilizam-se desta técnica.
- “**idéia**” .. parte do número da página de um endereço virtual é mapeado em um valor “hash” que é um ponteiro para a “tabela de página invertida”, que contém as entradas da tabela de página.
- .. há uma entrada na tabela de página invertida para cada “page frame” da memória real, em vez de uma por página virtual.

## 8 – Virtual Memory / 8.1 – Hardware and Control Structure

### ... 8.1.2 – Simple Paging

- e.g., para uma memória real com  $2^m$  “frames”, a tabela de página invertida contém  $2^m$  entradas, ou seja, de 0 a  $2^m - 1$ , de modo que a  $i$ -ésima entrada se refere ao “frame” “ $i$ ”.



## 8 – Virtual Memory / 8.1 – Hardware and Control Structure

### ... 8.1.2 – Simple Paging

- **“page table entry”** .. cada entrada na tabela de página contém:
- **“page number”** .. parte do número da página do endereço virtual.
- **“process identifier”** .. processo que possui esta página, onde a combinação do nro. da página e do ID do processo identifica uma página no espaço de endereço virtual de um determinado processo.
- **“control bits”** .. este campo inclui sinalizadores, como válido, referenciado, modificado e informações de proteção e bloqueio.
- **“chain pointer”** .. este campo é nulo (talvez indicado por um bit separado) se não houver entradas encadeadas para esta entrada.
- .. caso contrário, o campo contém o valor do índice (número entre 0 e  $2^m - 1$ ) da próxima entrada na cadeia.

## 8 – Virtual Memory / 8.1 – Hardware and Control Structure

### ... 8.1.2 – Simple Paging

- “**observação**” .. neste exemplo, o endereço virtual inclui um número de página de  $n$  bits, com “ $n > m$ ” (atenção pra esta asserção).
- .. função “hash” mapeia o nro. da página de  $n$  bits em uma quantidade de  $m$  bits, que é usada para indexar na tabela de página invertida.
- “**considerações**” .. como mais de um endereço virtual pode ser mapeado na mesma entrada da tabela hash, uma técnica de encadeamento é usada para gerenciar o estouro.
- .. técnica de “hashing” resulta em cadeias que são normalmente curtas entre uma e duas entradas.
- .. estrutura da tabela de página é chamada de “invertida” porque indexa as entradas da tabela de página pelo número do “frame”, em vez do número da página virtual.

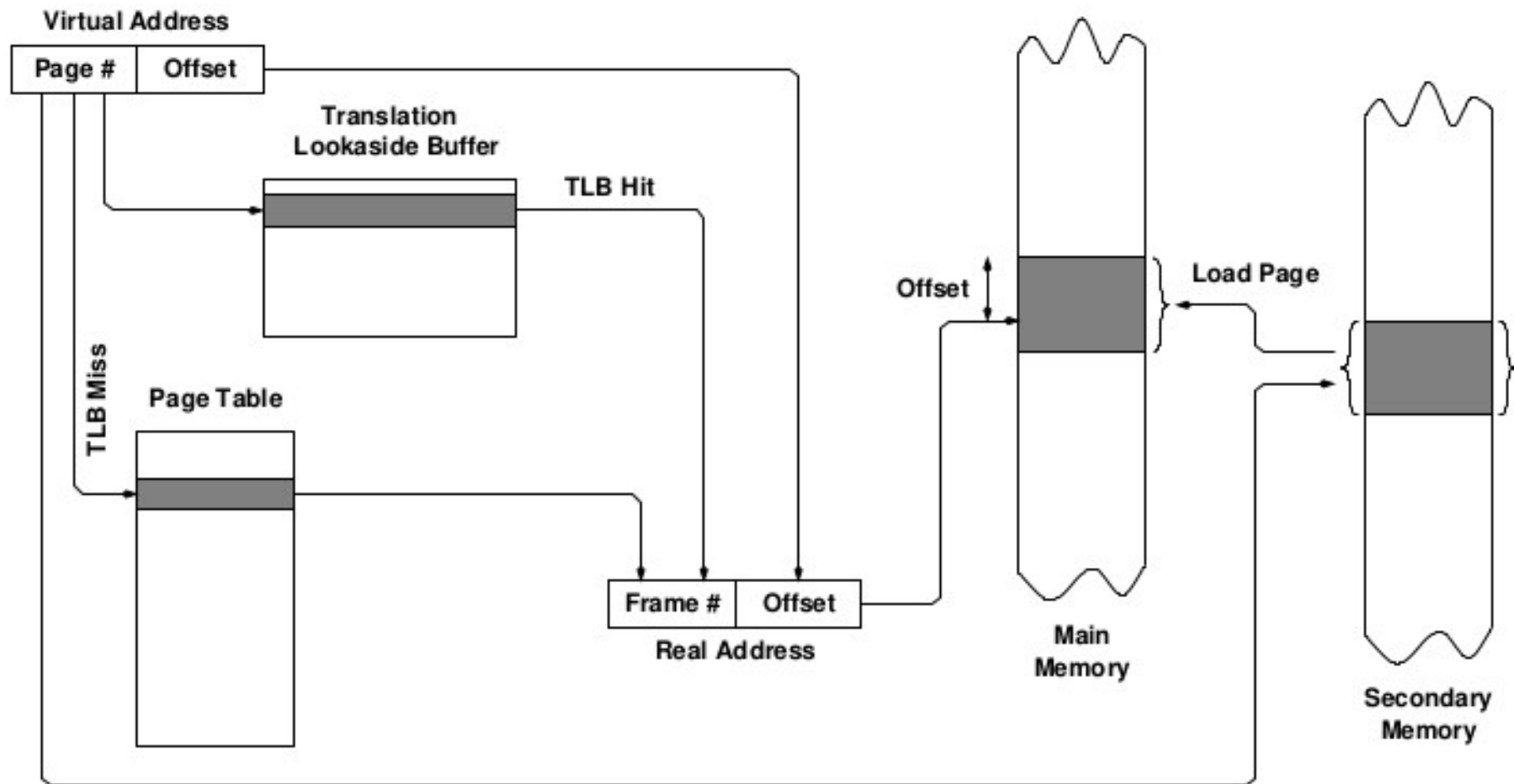
## 8 – Virtual Memory / 8.1 – Hardware and Control Structure

### ... 8.1.2 – Simple Paging

- **“virtual memory reference”** .. toda referência à memória virtual causa 02 acessos, 01 para buscar a entrada apropriada da tabela de páginas e 01 para buscar o dado desejado.
- ... assim, um esquema simples de memória virtual tem o efeito de dobrar o tempo de acesso à memória.
- **“translation lookaside buffer”** .. para superar esse problema, usa-se um cache especial de alta velocidade para entradas de tabela de página, geralmente chamado de “buffer de tradução” (TLB).
- .. estas funções da cache são as mesmas que aquelas da cache de memória principal e contém as entradas mais recentemente referenciadas da tabela de páginas.

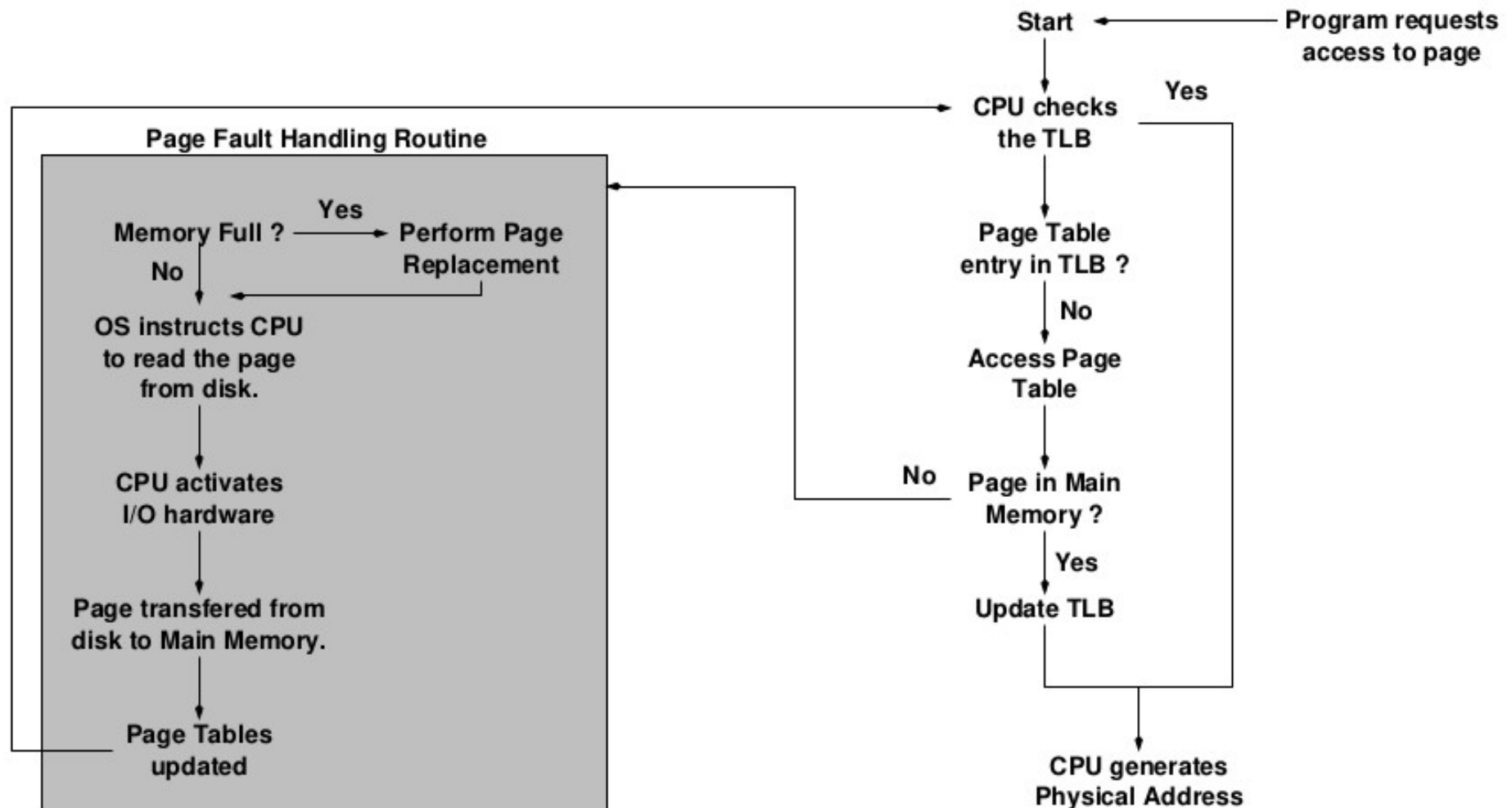
## 8 – Virtual Memory / 8.1 – Hardware and Control Structure ... 8.1.2 – Simple Paging

- Cache para entradas da Tabela de Páginas — TLB.



## 8 – Virtual Memory / 8.1 – Hardware and Control Structure ... 8.1.2 – Simple Paging

- O diagrama abaixo ilustra o uso da cache TLB:



## 8 – Virtual Memory / 8.1 – Hardware and Control Structure

### ... 8.1.2 – Simple Paging

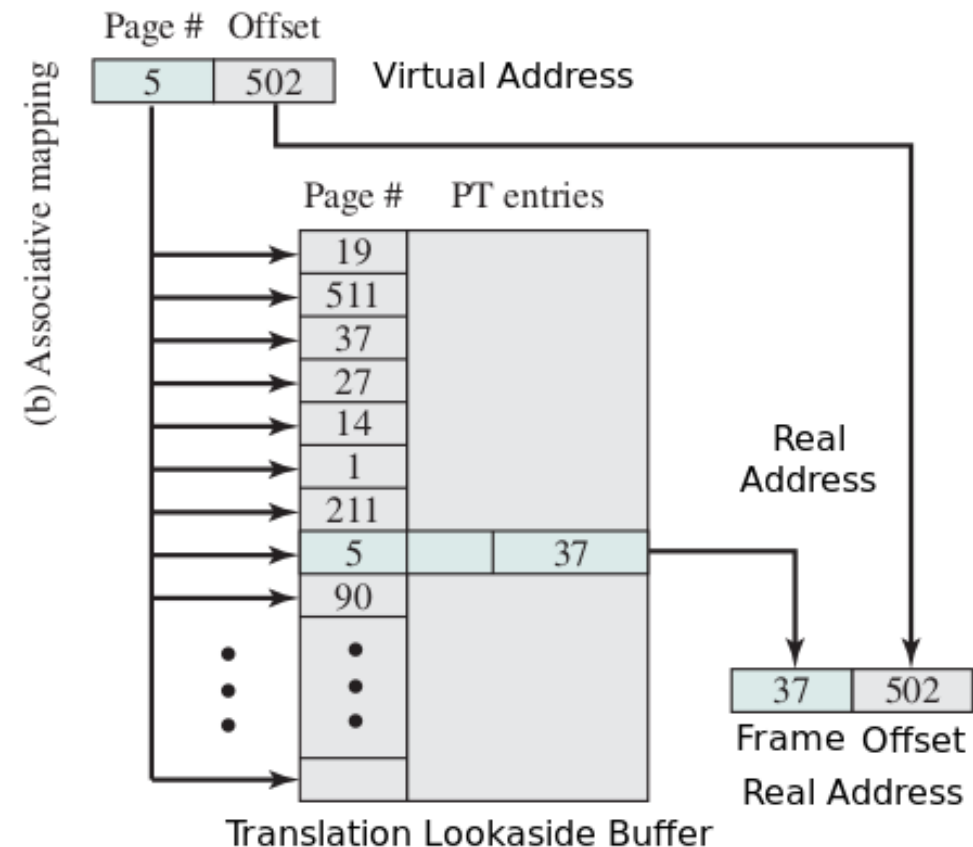
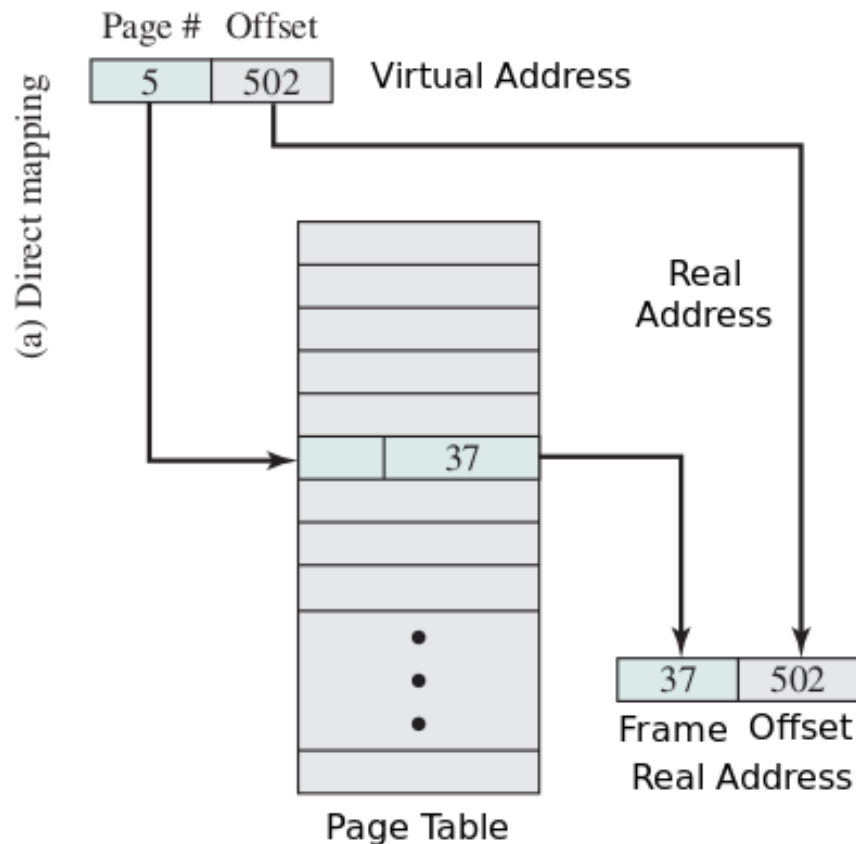
- “**additional details**” .. há vários detalhes relativos à organização real da TLB, p.ex., como indexar a TLB com base no nro. de página se a TLB contém apenas algumas entradas de uma Tabela de Páginas ?
- .. e vez disso, cada entrada no TLB deve incluir o número da página, bem como a entrada completa da tabela da página.
- .. processador está equipado com hardware que permite interrogar simultaneamente várias entradas TLB para determinar se há uma correspondência no número da página.
- .. essa técnica é conhecida como “**mapeamento associativo**” e é contrastada com o “**mapeamento direto**”, ou indexação, usado para consulta na tabela de páginas.



## 8 – Virtual Memory / 8.1 – Hardware and Control Structure

### ... 8.1.2 – Simple Paging

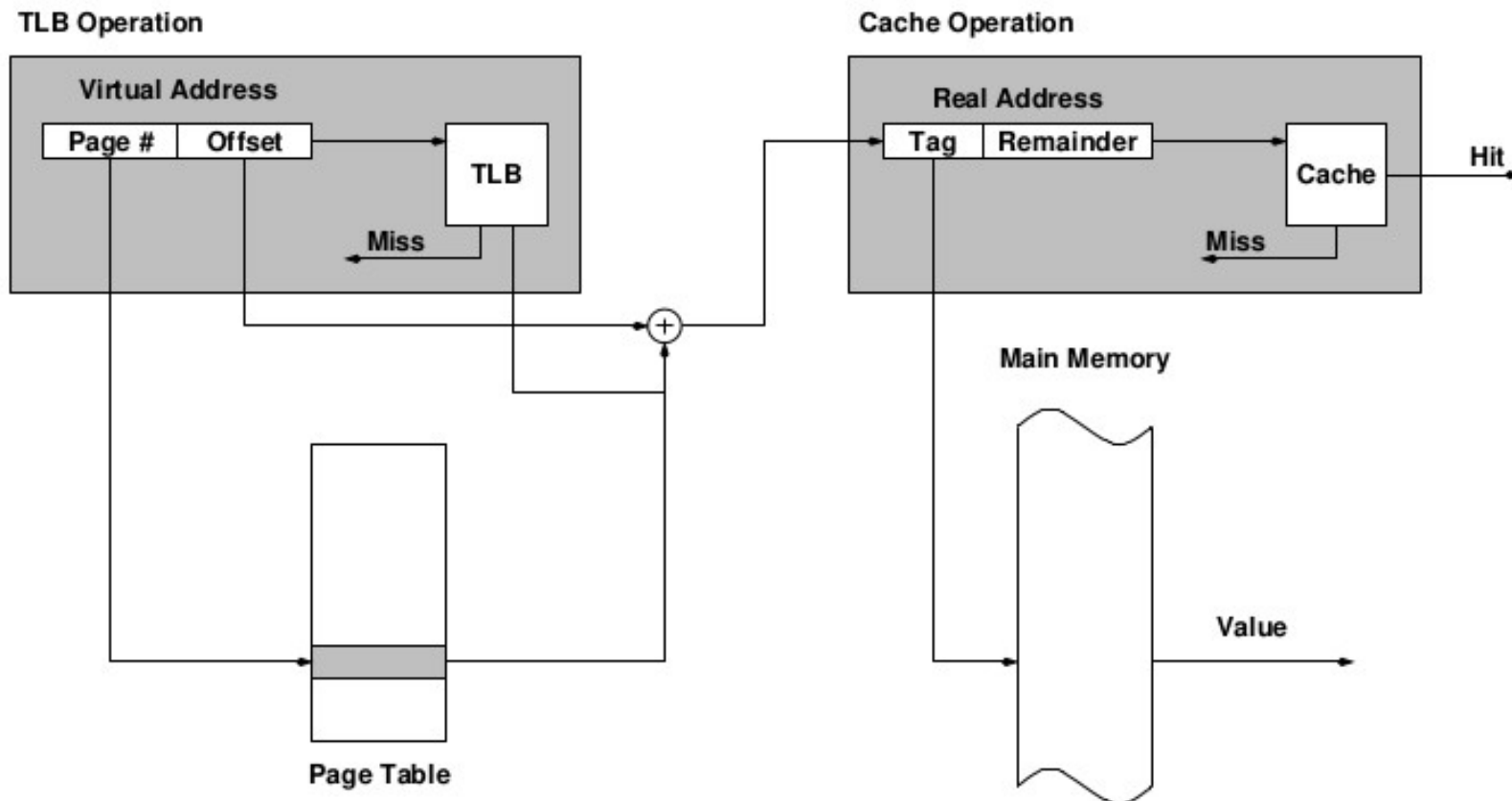
- “**design**” .. projeto da TLB deve considerar a maneira como as entradas são organizadas e qual entrada substituir quando uma nova entrada é introduzida na TLB.



## 8 – Virtual Memory / 8.1 – Hardware and Control Structure

### ... 8.1.2 – Simple Paging

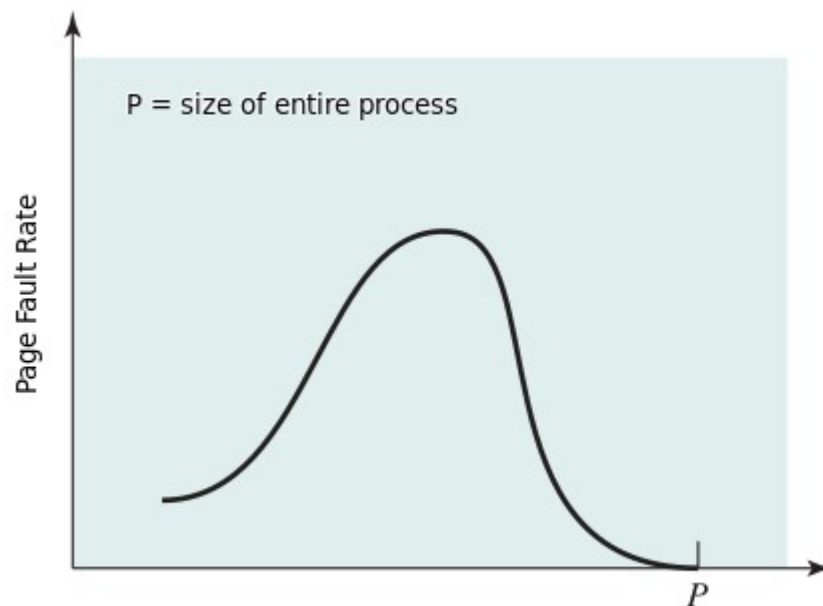
- Finalmente, o mecanismo de memória virtual precisa interagir com a cache do sistema computacional para efetuar a leitura do dado (palavra) da cache ou da memória principal.



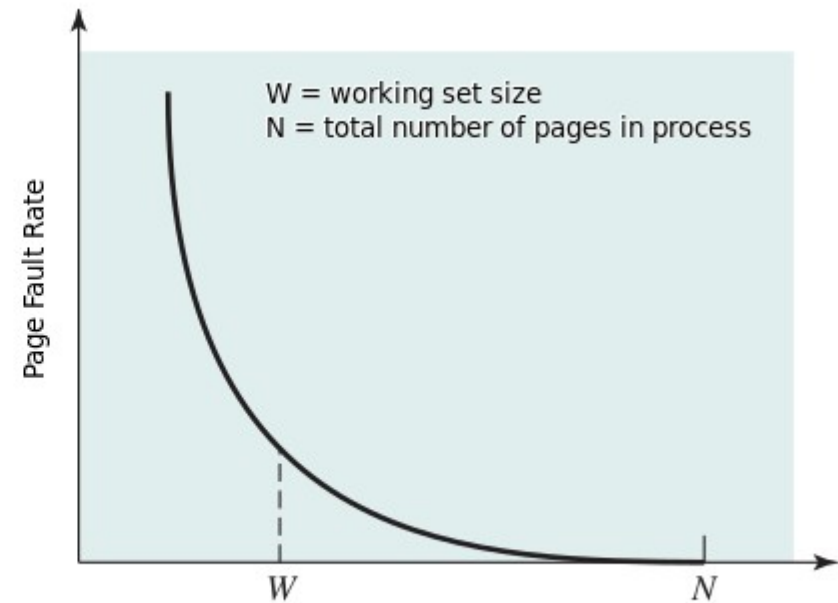
## 8 – Virtual Memory / 8.1 – Hardware and Control Structure

### ... 8.1.2 – Simple Paging

- “**page size**” .. aspecto importante de projeto no hardware é o tamanho da página e vários são os fatores a considerar:
- “**internal fragmentation**” .. por um lado, páginas menores significam menor fragmentação interna mas por outro maior será o número de páginas por processo.



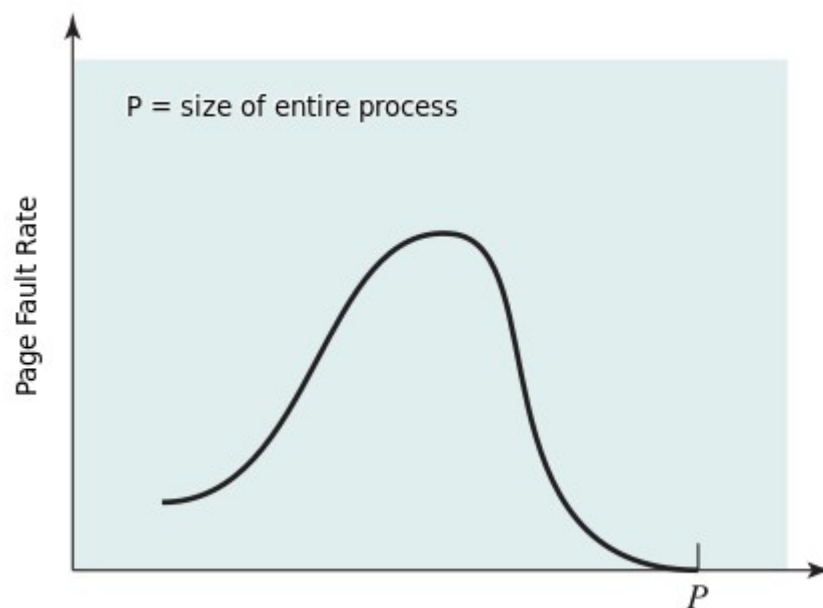
(a) Page size



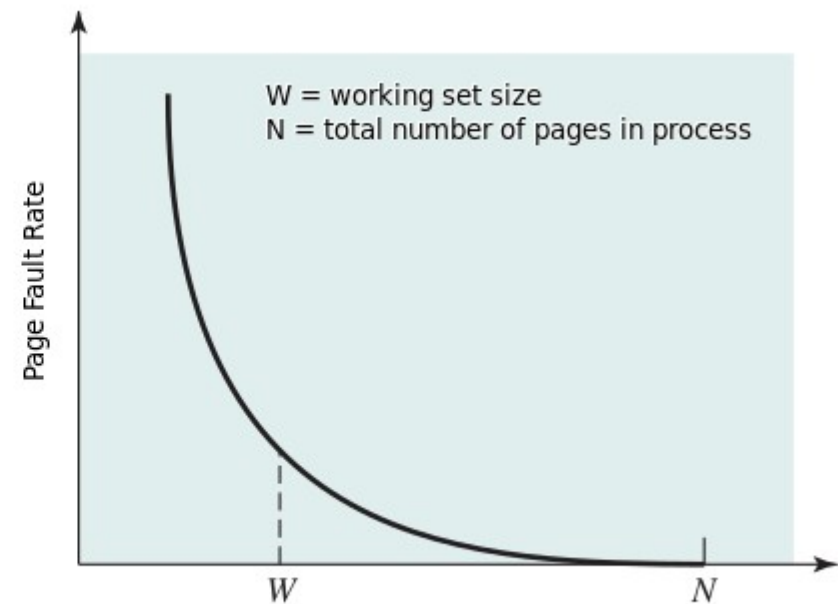
(b) Number of page frames allocated

## 8 – Virtual Memory / 8.1 – Hardware and Control Structure ... 8.1.2 – Simple Paging

- “**page size**” .. vários são os fatores a considerar:
- “**secondary memory**” .. características físicas favorecem páginas maiores quanto a eficiência na transferência dos blocos de dados.
- “**page fault frequency**” .. frequência com que faltas de páginas ocorrem é dependente do tamanho da página (gráfico abaixo)



(a) Page size



(b) Number of page frames allocated

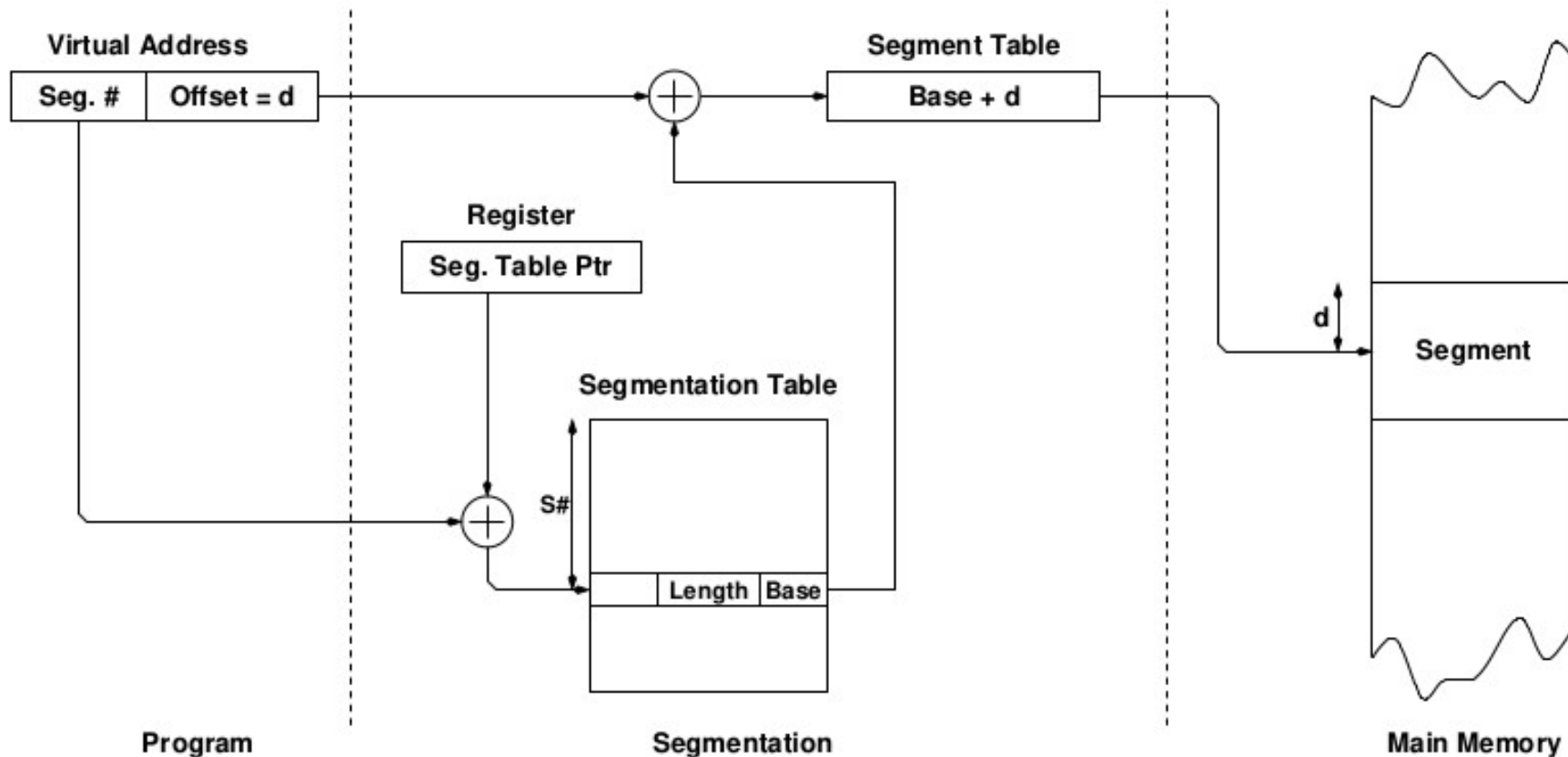
### 8.1.3 – Simple Segmentation

- **“segmentação”** .. possibilita que a memória seja particionada em múltiplos segmentos cujos tamanhos podem ser diferentes bem como variáveis no tempo (dinâmico).
- **“vantagens da segmentação”** .. sobre sistemas não segmentados:
- simplifica o tratamento de estruturas de dados que exigem mais espaço de memória durante o tempo de execução do processo.
- possibilita que programas sejam alterados e recompilados independentemente, sem exigir que todo o conjunto seja relinked ou recarregado.
- presta-se para o compartilhamento de dados entre processos, posto que esta região pode ser referenciada por mais de um processo.
- presta-se à própria proteção, posto que pode ser reservado para acomodar um conjunto de programas e dados (aspectos de segurança).

## 8 – Virtual Memory / 8.1 – Hardware and Control Structure

### ... 8.1.3 – Simple Segmentation

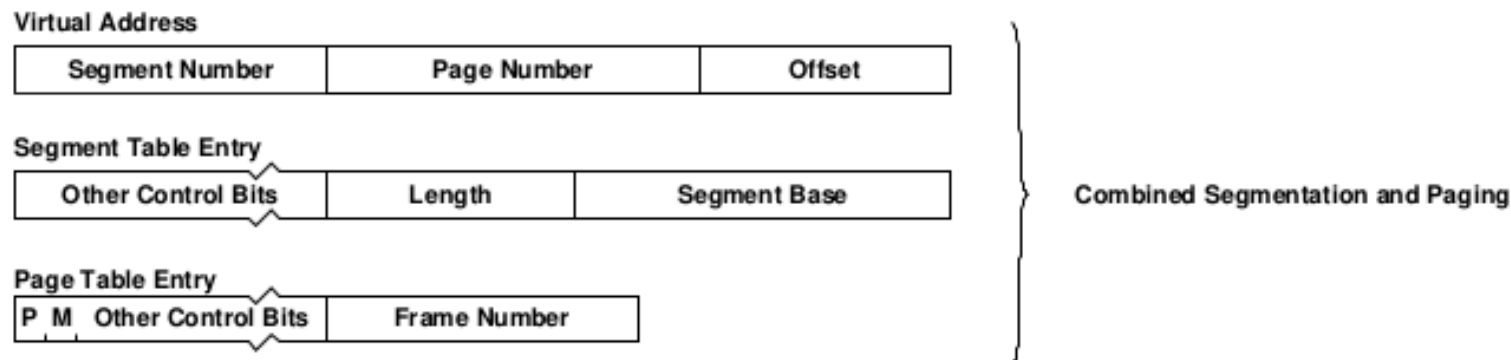
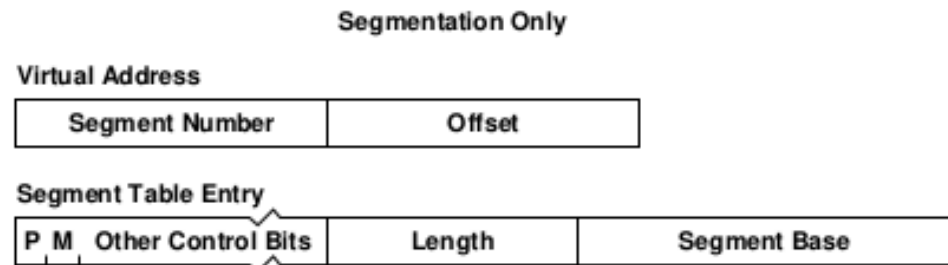
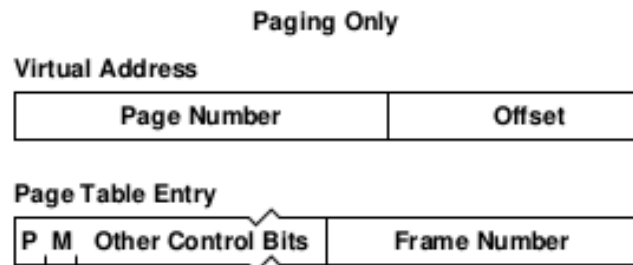
- “**implementação do hardware**” .. pelo fato da tabela de segmentos ser de tamanho variável dependendo do tamanho do processo, não se espera mantê-la em registradores, mas sim na memória principal.



## 8 – Virtual Memory / 8.1 – Hardware and Control Structure

### ... 8.1.3 – Simple Segmentation

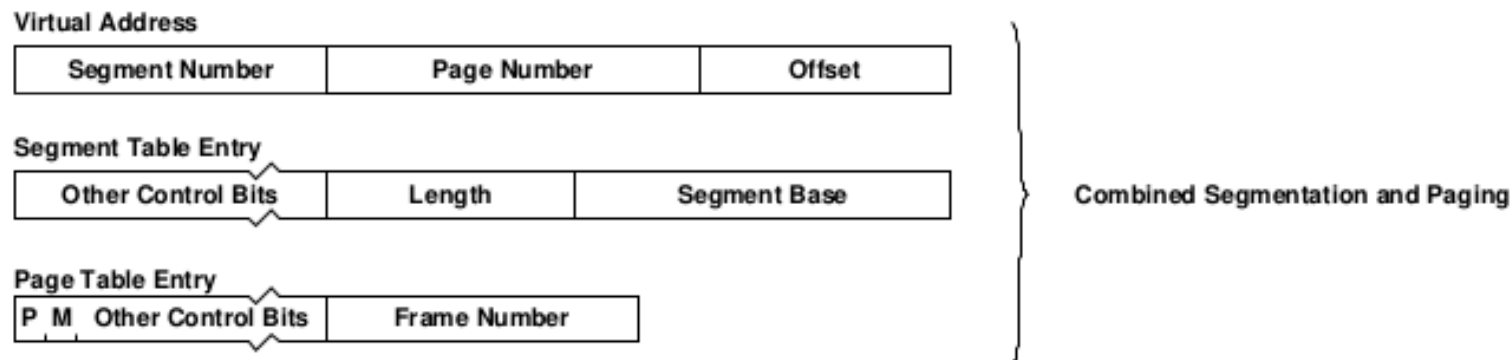
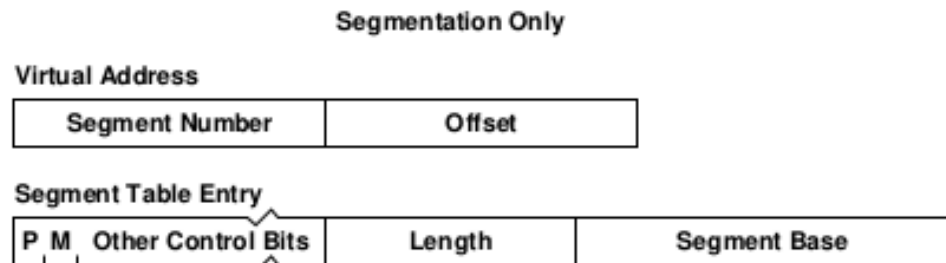
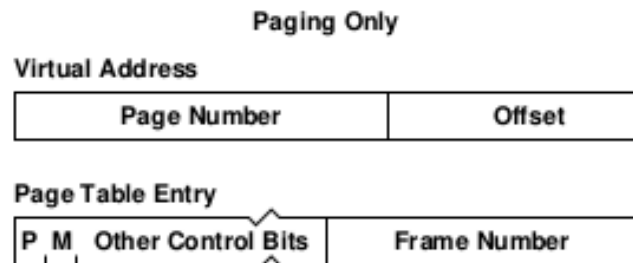
- “**vantagens**” .. tanto da paginação quanto segmentação:
- “**paginação**” .. transparente para o programador da aplicação, elimina a fragmentação externa oferecendo uso eficiente da memória principal.



## 8 – Virtual Memory / 8.1 – Hardware and Control Structure

### ... 8.1.4 – Combined Paging and Segmentation

- “**vantagens**” .. tanto da paginação quanto segmentação:
- “**segmentação**” .. visível ao programador, oferece a habilidade de tratar alocação dinâmica de dados, modularidade e suporte ao compartilhamento e proteção de dados.

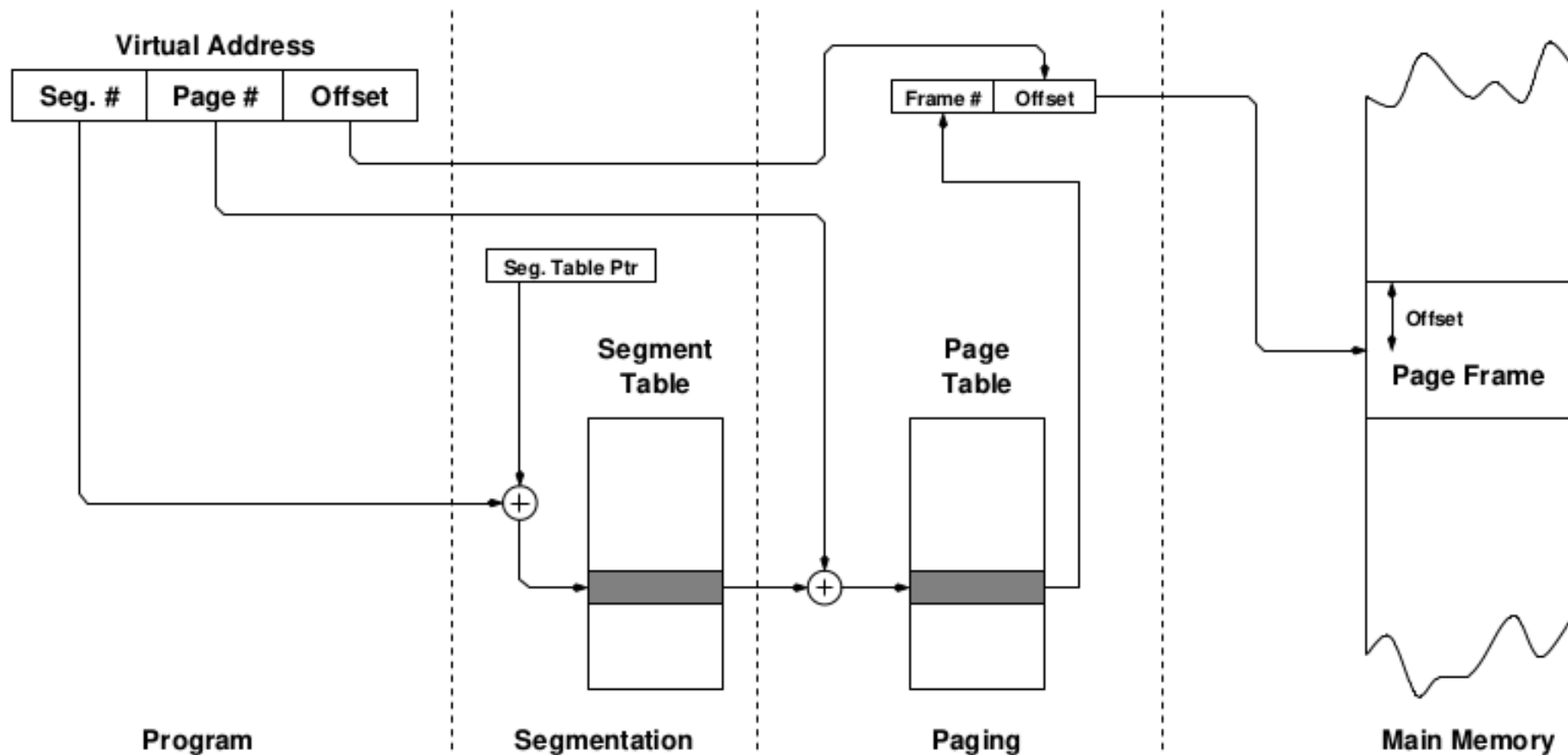




## 8 – Virtual Memory / 8.1 – Hardware and Control Structure

### ... 8.1.4 – Combined Paging and Segmentation

- Estrutura de suporte para Paginação e Segmentação combinadas:



## 8 – Virtual Memory / 8.1 – Hardware and Control Structure

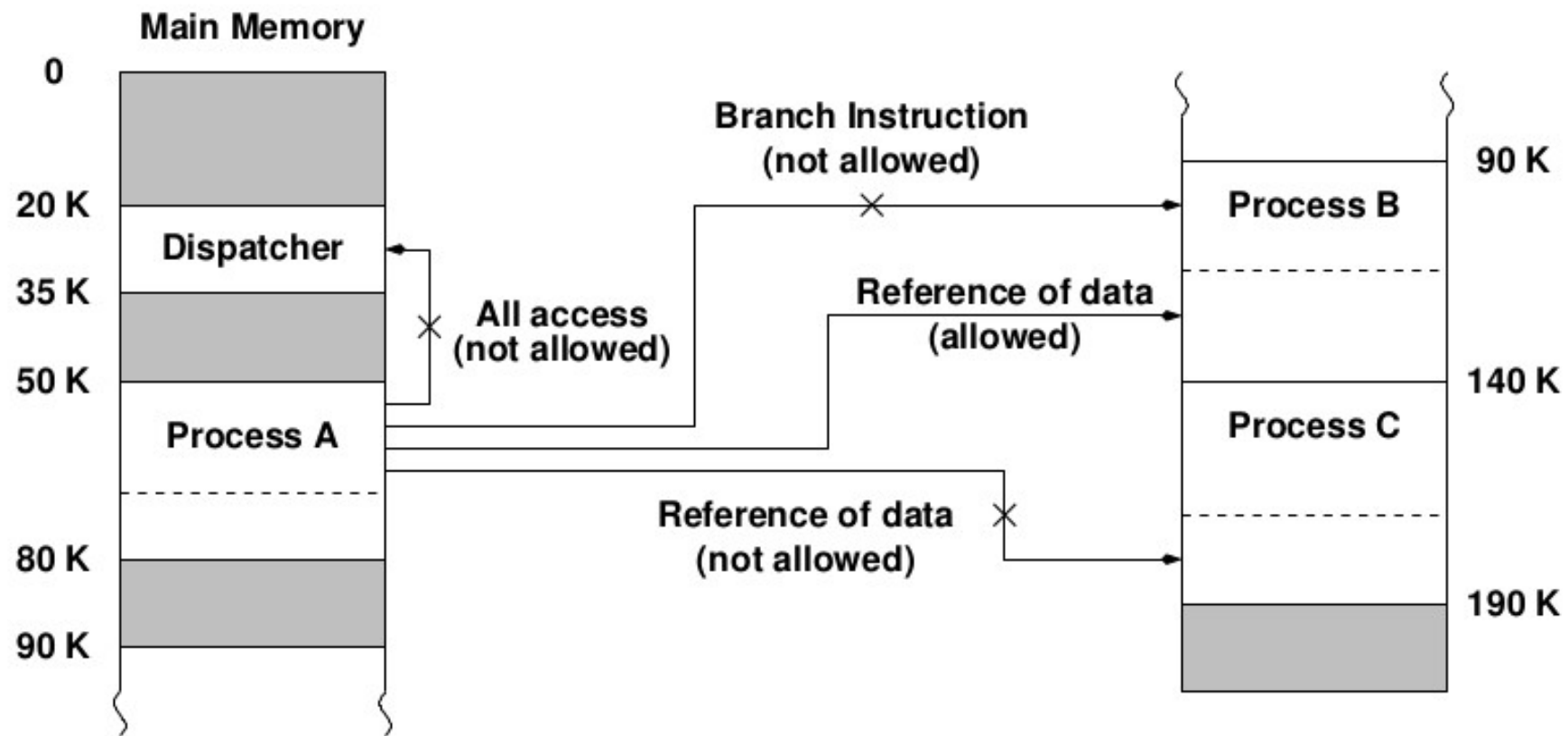
### 8.1.5 – Protection and Sharing

- .. segmentação permite a implementação de políticas de proteção e compartilhamento - “**protection and sharing policies**”
- “**protection and sharing policies**” .. como cada entrada inclui o comprimento e endereço base, um programa não consegue acessar inadvertidamente uma posição da memória fora dos limites estabelecidos.
- ... para compartilhar um segmento, faz-se necessário referenciá-lo por mais de um processo em sua tabela de segmentos bem como na tabela de regiões por processo.
- ... na paginação, um mecanismo semelhante é possível, entretanto, a estrutura de página não é visível ao programador tornando a especificação de proteção e compartilhamento um tanto quanto desajeitada.

## 8 – Virtual Memory / 8.1 – Hardware and Control Structure

### ... 8.1.5 – Protection and Sharing

- “**ring protection structure**” .. um esquema mais sofisticado de proteção é a estrutura de proteção em anel que concede privilégio maior para os anéis mais internos e menor os externos.



## 8 – Virtual Memory / 8.2 – Operating System Software

### 8.2 – Operating System Software

- **“design of memory management”** .. elementos de projeto ou políticas tem como aspecto chave a performance, ou seja, minimizar o nro. de faltas de páginas uma vez que causam considerável “overhead”.

---

<b>Fetch Policy</b> Demand Preparing	<b>Resident Set Management</b> Resident Set Size Fixed Variable Replacement Scope Global Local
<b>Placement Policy</b>	
<b>Replacement Policy</b> Basic Algorithms Optimal Least Recently Used First In, First Out Clock Page Buffering	<b>Cleaning Policy</b> Demand Precleaning  <b>Load Control</b> Degree of Multiprogramming

---

## 8 – Virtual Memory / 8.2 – Operating System Software

### ... 8.2 – Operating System Software

- **“design of memory management”** .. o projeto do gerenciamento de memória de um sist. oper. depende de 03 áreas fundamentais:
  - (1) usar ou não técnicas de memória virtual.
  - (2) uso de paginação, segmentação ou combinação de ambos.
  - (3) algoritmos empregados no gerenciamento de memória.
- .. escolhas feitas nas 02 primeiras áreas dependem da plataforma de “hardware” disponível, ou seja, sistema computacional.
- .. nenhuma dessas técnicas é aplicável sem suporte de “hardware” para tradução de endereços e outras funções básicas.
- e.g., implementações anteriores do UNIX não forneciam memória virtual porque os processadores nos quais o sistema era executado não suportavam paginação e/ou segmentação.

## 8 – Virtual Memory / 8.2 – Operating System Software

### ... 8.2 – Operating System Software

- **“design of memory management”** .. o projeto do gerenciamento de memória de um sist. oper. depende de três áreas fundamentais:
  - (1) usar ou não técnicas de memória virtual.
  - (2) uso de paginação, segmentação ou combinação de ambos.
  - (3) algoritmos empregados no gerenciamento de memória.
- .. 02 comentários adicionais sobre os 02 primeiros itens:
- 1) com exceção dos sist. oper. para computadores pessoais mais antigos, p.ex., MS-DOS, e sistemas especializados, todos os sistemas operacionais importantes fornecem memória virtual.
- 2) quando a segmentação é combinada com a paginação, a maioria dos problemas de gerenciamento de memória enfrentados pelo “designer” está na área de paginação.

## 8 – Virtual Memory / 8.2 – Operating System Software

### ... 8.2 – Operating System Software

- No outro extremo, o sistema pode tomar a decisão que páginas devem permanecer e quais devem ser movidas para a memória secundária.
- ... na seqüência, o sistema operacional deve escalonar um outro processo durante a operação de entrada/saída da página, causando o chaveamento de processo.
- .. adicionalmente, a performance de qualquer política é dependente:
- do tamanho da memória principal.
- da velocidade relativa entre memória principal e secundária.
- do número e tamanho de processos competindo por recursos.
- do comportamento de execução de cada um dos processos.

### 8.2.1 – Fetch Policy

- **“fetch policy”** .. política de busca determina quando uma página deve ser trazida para a memória principal.
- **“demand paging”** .. na paginação sob demanda, uma página é trazida para a memória principal apenas quando uma referência é feita a um local nessa página.
- .. em uma condição normal, quando um processo é iniciado pela 1ª vez, haverá uma enxurrada de faltas de página.
- .. a medida que mais e mais páginas são trazidas, o princípio da localidade sugere que a maioria das referências futuras serão a páginas que foram trazidas recentemente.
- .. assim, depois de um tempo, as coisas devem se acalmar e o número de faltas de página deve cair para um nível mais baixo.



## 8 – Virtual Memory / 8.2 – Operating System Software

### ... 8.2.1 – Fetch Policy

- **“fetch policy”** .. política de busca determina quando uma página deve ser trazida para a memória principal.
- **“prepaging”** .. com o pré-processamento, as páginas são trazidas para a memória sem que haja o gatilho da falta de página.
- .. pré-processamento explora as características da maioria dos dispositivos de memória secundária, como discos, que têm tempos de busca e latência rotacional.
- .. se as páginas de um processo forem armazenadas de forma contígua na memória secundária, será mais eficiente trazer várias páginas de uma vez do que trazê-las uma de cada vez por um longo período.
- .. naturalmente, esta política é ineficaz se a maioria das páginas extras que são trazidas não são referenciadas.

## 8 – Virtual Memory / 8.2 – Operating System Software

### 8.2.2 – Placement Policy

- “**placement policy**” .. política de localização determina onde, na memória real, uma parte do processo deve residir.
- .. em um sistema de segmentação puro, a política de localização é uma questão importante de “design” e políticas como best-fit, first-fit e assim por diante são alternativas possíveis.
- .. já em sistema com paginação pura ou paginação combinada com segmentação, o posicionamento é geralmente irrelevante.
- “**justificativa**” .. “hardware” de tradução de endereço e o “hardware” de acesso à memória principal podem executar suas funções para qualquer combinação de “page frame” com igual eficiência.

## 8 – Virtual Memory / 8.2 – Operating System Software

### ... 8.2.2 – Placement Policy

- “**placement policy**” .. política de atribuição/localização determina onde, na memória real, uma parte do processo deve residir.
- .. há uma área em que a colocação se torna uma preocupação, e este é um assunto para pesquisa e desenvolvimento.
- e.g., em um Multiprocessador NUMA (NonUniform Memory Access), a memória compartilhada e distribuída da máquina pode ser referenciada por qualquer processador na máquina.
- .. no entanto, o tempo para acessar um determinado local físico varia com a distância entre o processador e o módulo de memória.
- .. assim, o desempenho depende muito da extensão em que os dados residem perto dos processadores que os usam.

### 8.2.3 – Replacement Policy

- **“replacement policy”** .. trata da seleção de uma página na memória principal a ser substituída quando uma nova página deve ser trazida.
- **“interrelated concepts”** .. tópico contempla vários conceitos ..
- 1) quantos “frames” devem ser alocados para cada processo ativo ?
- 2) se o conjunto de páginas a ser considerado para substituição deve ser limitado àquelas do processo que causou a falta de página ou abranger todos os “frames” na memória principal ?
- 3) no conjunto de páginas consideradas, qual página particular deve ser selecionada para substituição – “replacement policy” ?
- .. 1º e 2º conceitos estão relacionados ao “resident set management” enquanto que o 3º está relacionado a política de substituição.

## 8 – Virtual Memory / 8.2 – Operating System Software

### ... 8.2.3 – Replacement Policy

- **“frame locking”** .. trata-se de uma restrição à política de substituição onde alguns dos “frames” podem estar bloqueados.
- .. quando um “frame” está bloqueado, a página atualmente armazenada naquele “frame” não pode ser substituída.
- .. muito do “kernel” do sistema operacional, bem como estruturas de controle de chave, são mantidos em “blocked frames”.
- .. além disso, os buffers de I / O e outras áreas críticas de tempo podem ser bloqueados em “frames” da memória principal.
- .. bloqueio é obtido associando um bit de bloqueio a cada “frame”, “bit” este que pode ser mantido em uma tabela de “frames”, além de ser incluído na “table page” atual.

## 8 – Virtual Memory / 8.2 – Operating System Software

### ... 8.2.3 – Replacement Policy

- “**basic algorithms**” .. independentemente da estratégia do “resident set management”, existem certos algoritmos básicos que são usados para a seleção de uma página a ser substituída.
- “**optimal**” .. trata-se da política ideal para substituição de página para a qual o tempo para a próxima referência é o mais longo e que resulta no menor número de faltas de página.
- “**least recently used**” .. substitui a página que não foi referenciada por muito tempo e que pelo princípio da localidade, deve ser a página com menos probabilidade de ser referenciada em um futuro próximo.

## 8 – Virtual Memory / 8.2 – Operating System Software

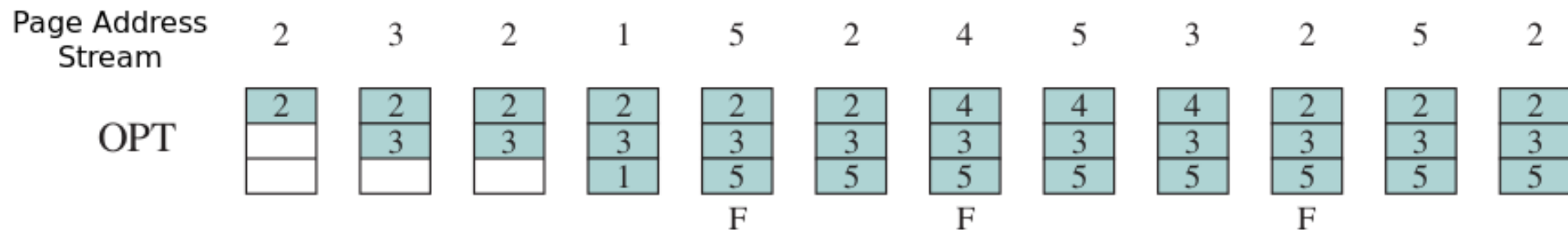
### ... 8.2.3 – Replacement Policy

- “**basic algorithms**” .. independentemente da estratégia do “resident set management”, existem certos algoritmos básicos que são usados para a seleção de uma página a ser substituída.
- “**first-in-first-out**” (FIFO) .. .. trata os “frames” de página alocados para um processo como um buffer circular e no qual as páginas são removidas no estilo “round-robin”.
- “**clock**” .. na forma mais simples requer a associação de um “bit” adicional a cada “frame”, conhecido como “use bit”.
- .. quando uma página é carregada pela 1ª vez em um “frame” na memória, o “use bit” para esse “frame” é definido como 1.

## 8 – Virtual Memory / 8.2 – Operating System Software

### ... 8.2.3 – Replacement Policy

- “**optimal**” .. trata-se da política ideal para substituição de página para a qual o tempo para a próxima referência é o mais longo e que resulta no menor número de faltas de página.
- .. trata-se de uma política impossível de se implementar, pois exige que o sist. oper. tenha conhecimento perfeito dos eventos futuros.
- .. no entanto, esta política serve como um padrão ou referência para comparar com outros algoritmos do mundo real.



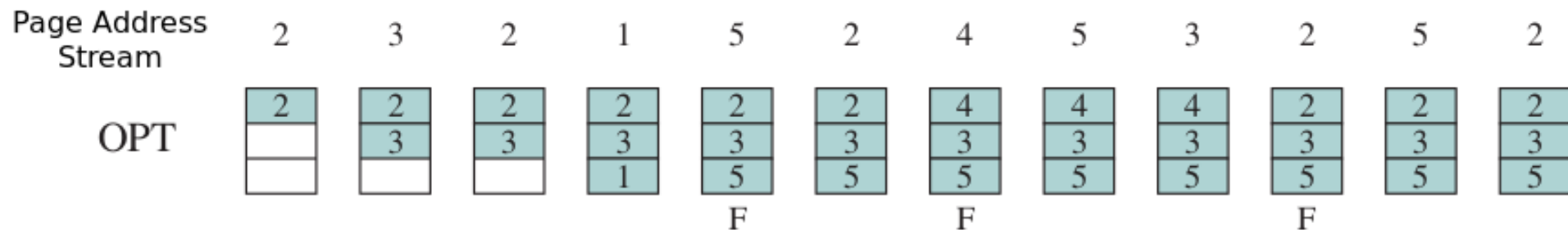
F = page fault occurring after the frame allocation is initially filled.



## 8 – Virtual Memory / 8.2 – Operating System Software

### ... 8.2.3 – Replacement Policy

- e.g., exemplo abaixo assume uma alocação de quadro fixa (tamanho do conjunto residente fixo) para este processo de 03 “frames” e execução do processo requer referência a 05 páginas distintas.
- .. fluxo de endereços de páginas na execução do programa indica que a 1ª página referenciada é 2, a 2ª página referenciada é 3, a 3ª página referenciada é 2 e assim por diante.
- .. política ideal produz 03 (três) faltas de página após o preenchimento da alocação de “frames”.

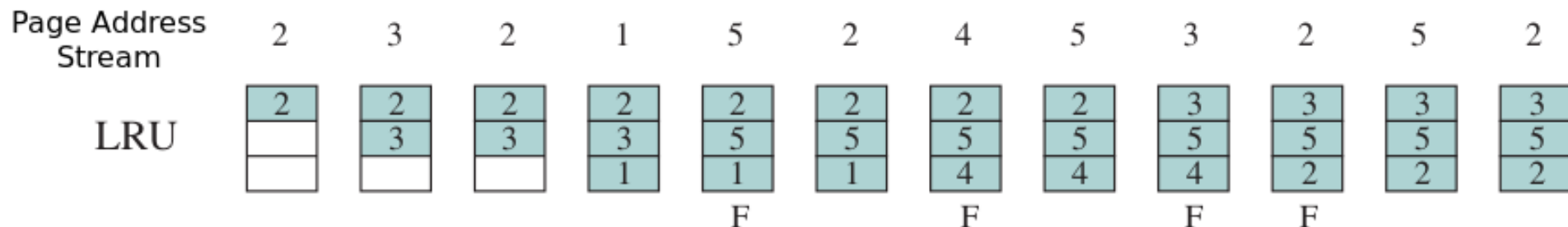


F = page fault occurring after the frame allocation is initially filled.

## 8 – Virtual Memory / 8.2 – Operating System Software

### ... 8.2.3 – Replacement Policy

- “**least recently used**” .. substitui a página que não foi referenciada por muito tempo e que pelo princípio da localidade, deve ser a página com menos probabilidade de ser referenciada em um futuro próximo.
- .. de fato, a política LRU faz quase tão bem quanto a política ótima, no entanto, o problema é a dificuldade de implementação.
- .. uma abordagem é marcar cada página com a hora de sua última referência, ou seja, a cada referência de memória, instrução e dados.

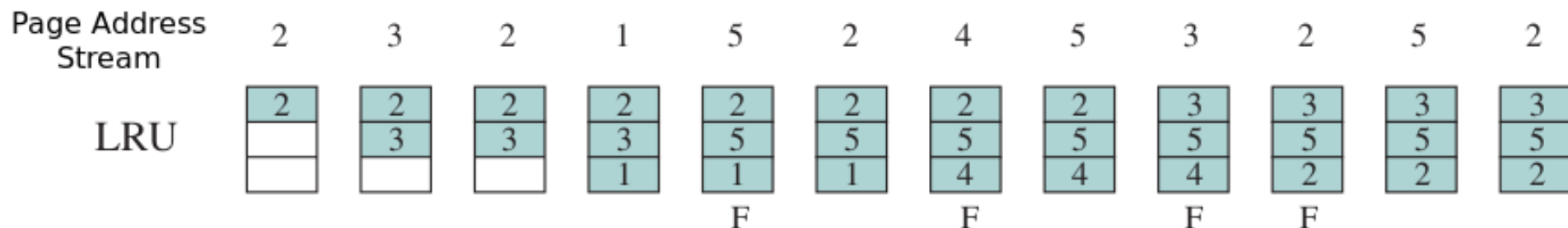


F = page fault occurring after the frame allocation is initially filled.

## 8 – Virtual Memory / 8.2 – Operating System Software

### ... 8.2.3 – Replacement Policy

- .. mesmo se o “hardware” suportar tal esquema, a sobrecarga é grande, por isso, uma alternativa é manter um pilha de referências de página, mas também parece ter um custo alto.
- .. figura abaixo mostra um exemplo do comportamento de LRU, usando o mesmo fluxo de endereço de página do exemplo de política ideal e, como pode ser observado, há 04 faltas de página.

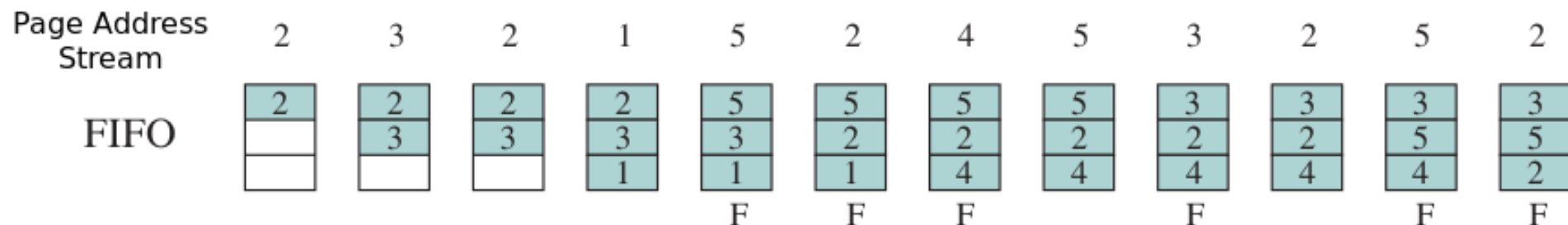


F = page fault occurring after the frame allocation is initially filled.

## 8 – Virtual Memory / 8.2 – Operating System Software

### ... 8.2.3 – Replacement Policy

- **“first-in first out”** .. trata os “frames” de página alocados para um processo como um “buffer” circular e no qual as páginas são removidas no estilo “round-robin”.
- .. tudo o que é necessário é um ponteiro que circule pelos “frames” da página do processo, ou seja, trata-se de uma das políticas de substituição de página mais simples de se implementar.
- .. além de ser uma abordagem simples, a lógica é que se está substituindo a página que está na memória há mais tempo, ou seja, uma página buscada há muito tempo pode agora ter caído em desuso.

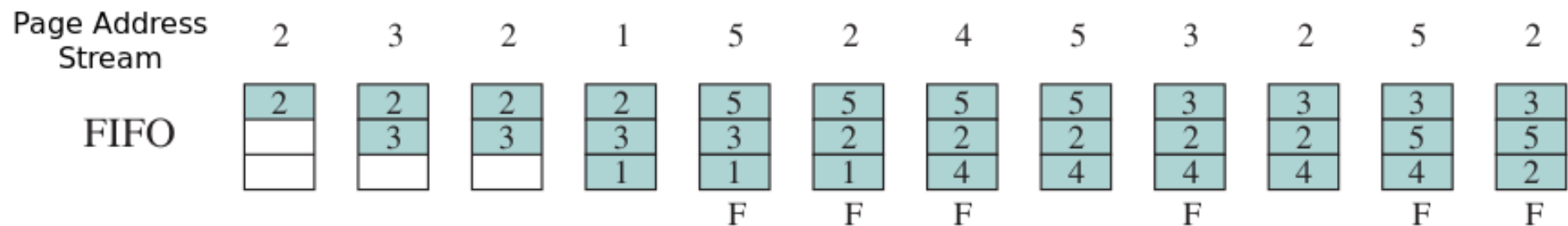


F = page fault occurring after the frame allocation is initially filled.

## 8 – Virtual Memory / 8.2 – Operating System Software

### ... 8.2.3 – Replacement Policy

- .. esse raciocínio frequentemente contém erros, porque muitas vezes há regiões do programa ou dados que são muito usados ao longo do ciclo de execução de um processo.
- .. no entanto, pela política “FIFO” essas páginas serão repetidamente paginadas para dentro e para fora pelo algoritmo FIFO.
- .. política FIFO gera 06 faltas de página, mas observe que o LRU reconhece que as páginas 2 e 5 são referenciadas com mais frequência do que as outras páginas, enquanto o FIFO não.

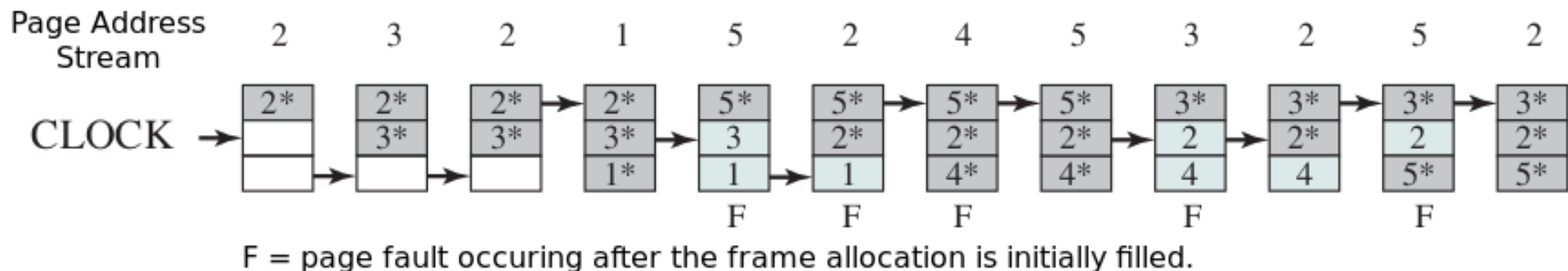


F = page fault occurring after the frame allocation is initially filled.

## 8 – Virtual Memory / 8.2 – Operating System Software

### ... 8.2.3 – Replacement Policy

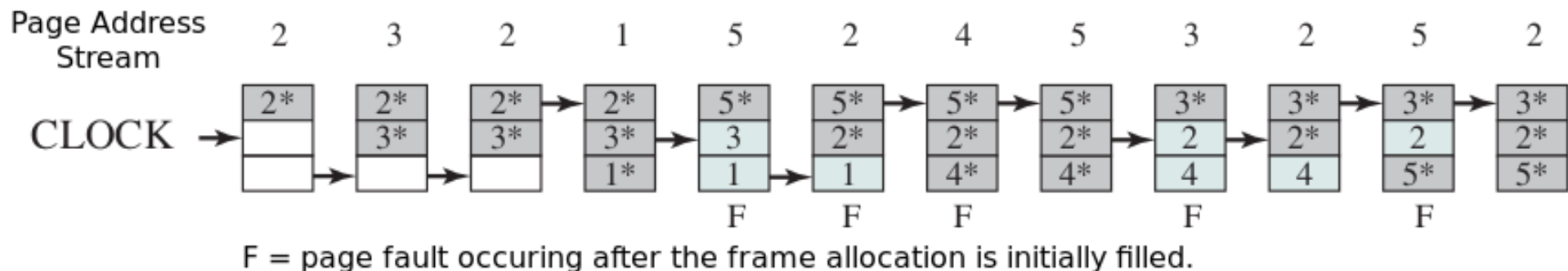
- “**clock**” .. na forma mais simples requer a associação de um “bit” adicional a cada “frame” conhecido como “use bit”.
- .. quando uma página é carregada pela 1ª vez em um “frame” na memória, o “use bit” para esse “frame” é definido como 1.
- .. sempre que a página é novamente referenciada (após a referência que gerou a falta de página), seu “use bit” de uso é definido como 1.
- .. tentativas no sentido de aproximar-se do desempenho do LRU, impõem pouca sobrecarga e boa parte desses algoritmos são variantes de um esquema conhecido como política de “clock”.



## 8 – Virtual Memory / 8.2 – Operating System Software

### ... 8.2.3 – Replacement Policy

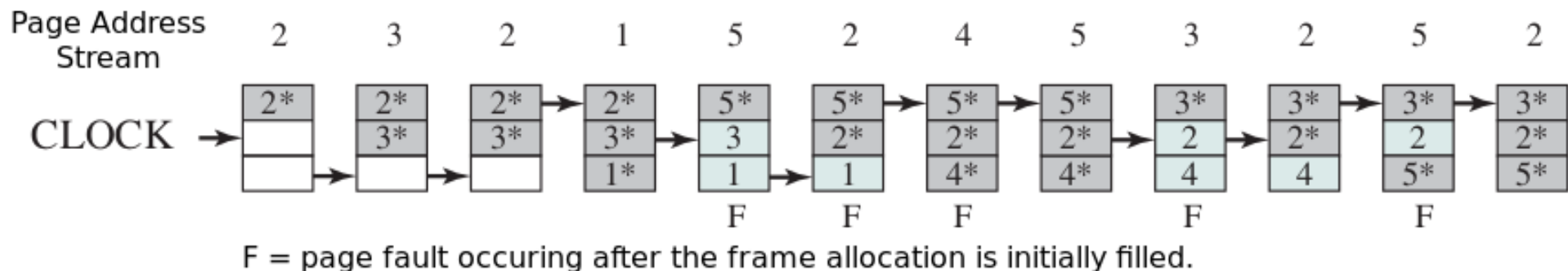
- .. para o algoritmo de substituição de página, o conjunto de “frames” candidatos à substituição é considerado um buffer circular, ao qual um ponteiro está associado.
- .. quando uma página é substituída, o ponteiro é definido para indicar o próximo “frame” no buffer após aquele que acabou de ser atualizado.
- .. quando chega a hora de substituir uma página, o sist. oper. verifica o buffer para encontrar um “frame” com um “use bit” = 0.



## 8 – Virtual Memory / 8.2 – Operating System Software

### ... 8.2.3 – Replacement Policy

- .. cada vez que ele encontra um quadro com um “use bit” = 1, ele redefine esse “bit” para 0 e continua.
- .. se qualquer um dos quadros no buffer tiver um “use bit” = 0 no início desse processo, o primeiro dentre eles é escolhido para substituição.
- .. se todos os quadros tiverem um “use bit” = 1, o ponteiro fará um ciclo completo no buffer, definindo todos os “use bits” = 0, e parará em sua posição original, substituindo a página naquele “frame”.

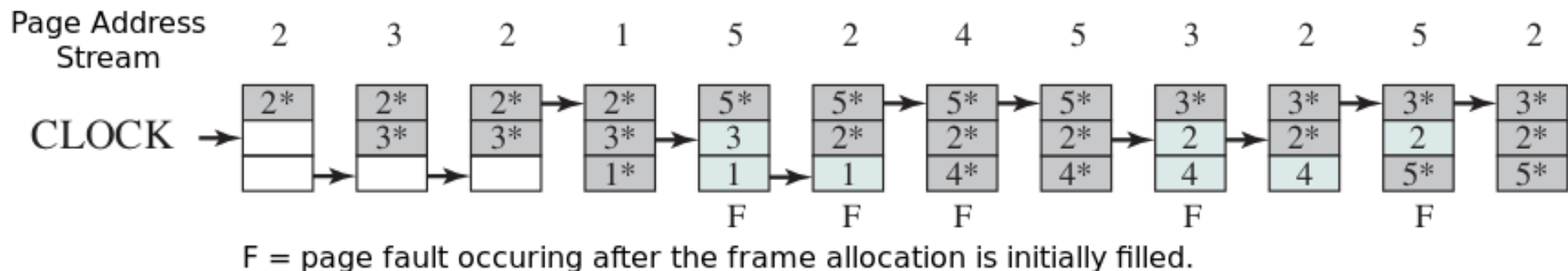




## 8 – Virtual Memory / 8.2 – Operating System Software

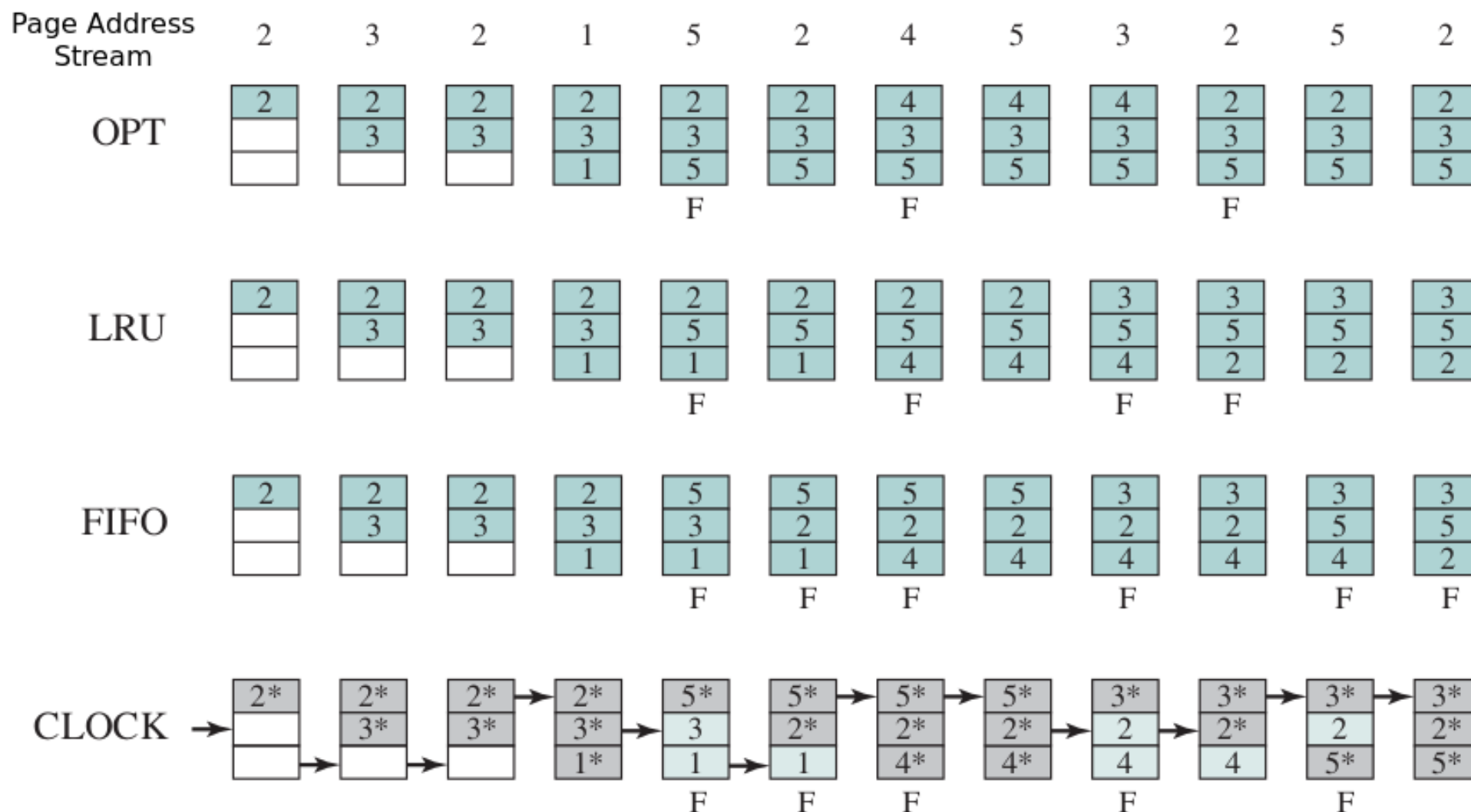
### ... 8.2.3 – Replacement Policy

- .. política do “clock” é semelhante à FIFO, exceto que, na política do relógio, qualquer frame com “use bit” = 1 é passado pelo algoritmo.
- .. a política é conhecida como “clock police” porque podemos visualizar os frames da página dispostos em um círculo.
- .. vários sistemas operacionais empregaram alguma variação dessa política de relógio considerada simples.



# 8 – Virtual Memory / 8.2 – Operating System Software

## ... 8.2.3 – Replacement Policy

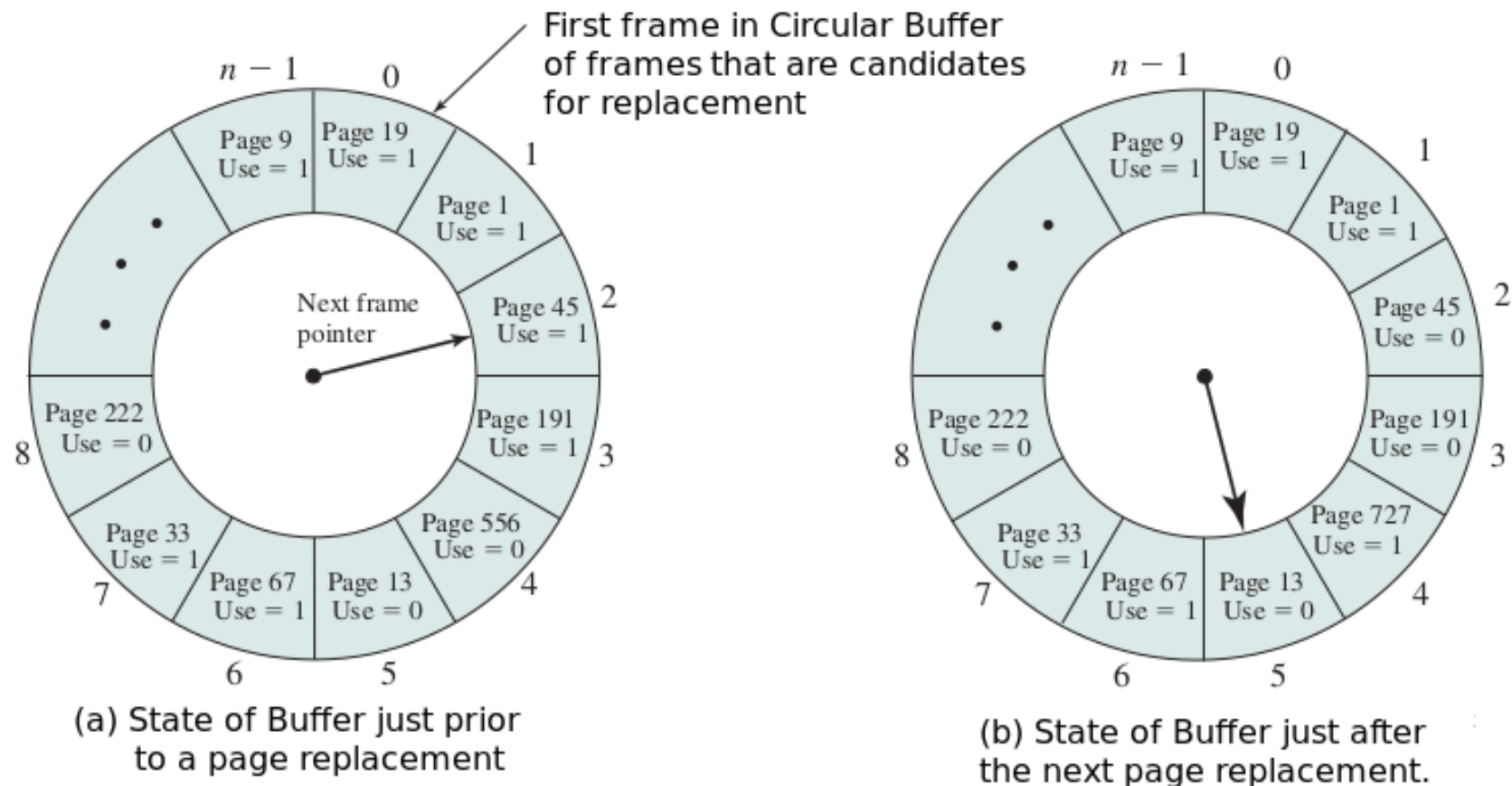


F = page fault occuring after the frame allocation is initially filled.

## 8 – Virtual Memory / 8.2 – Operating System Software

### ... 8.2.3 – Replacement Policy

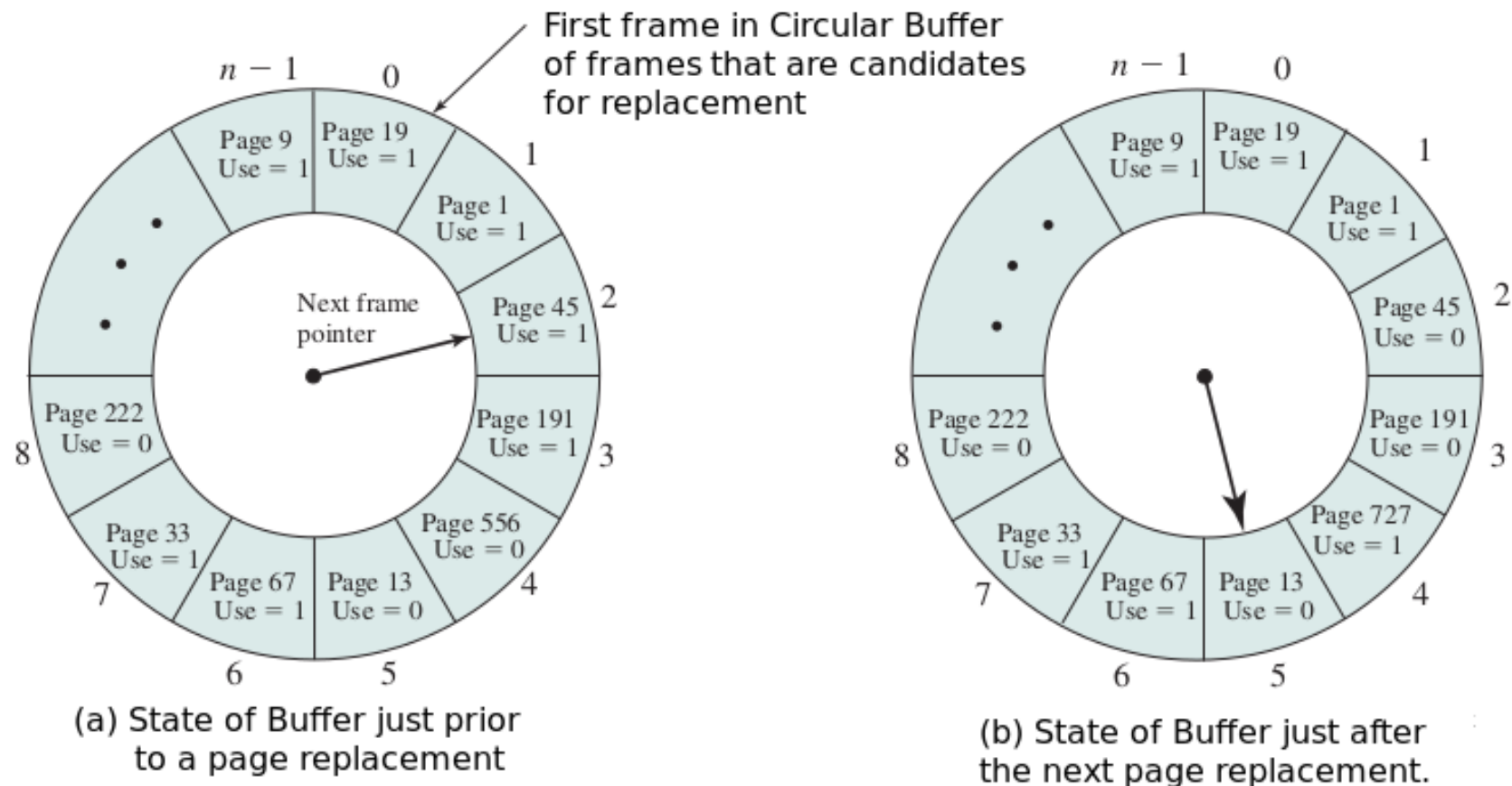
- e.g., figura a seguir ilustra um mecanismo simples de política de “clock”, por meio de um “buffer” circular de “n” “frames” de memória principal está disponível para substituição de página.



## 8 – Virtual Memory / 8.2 – Operating System Software

### ... 8.2.3 – Replacement Policy

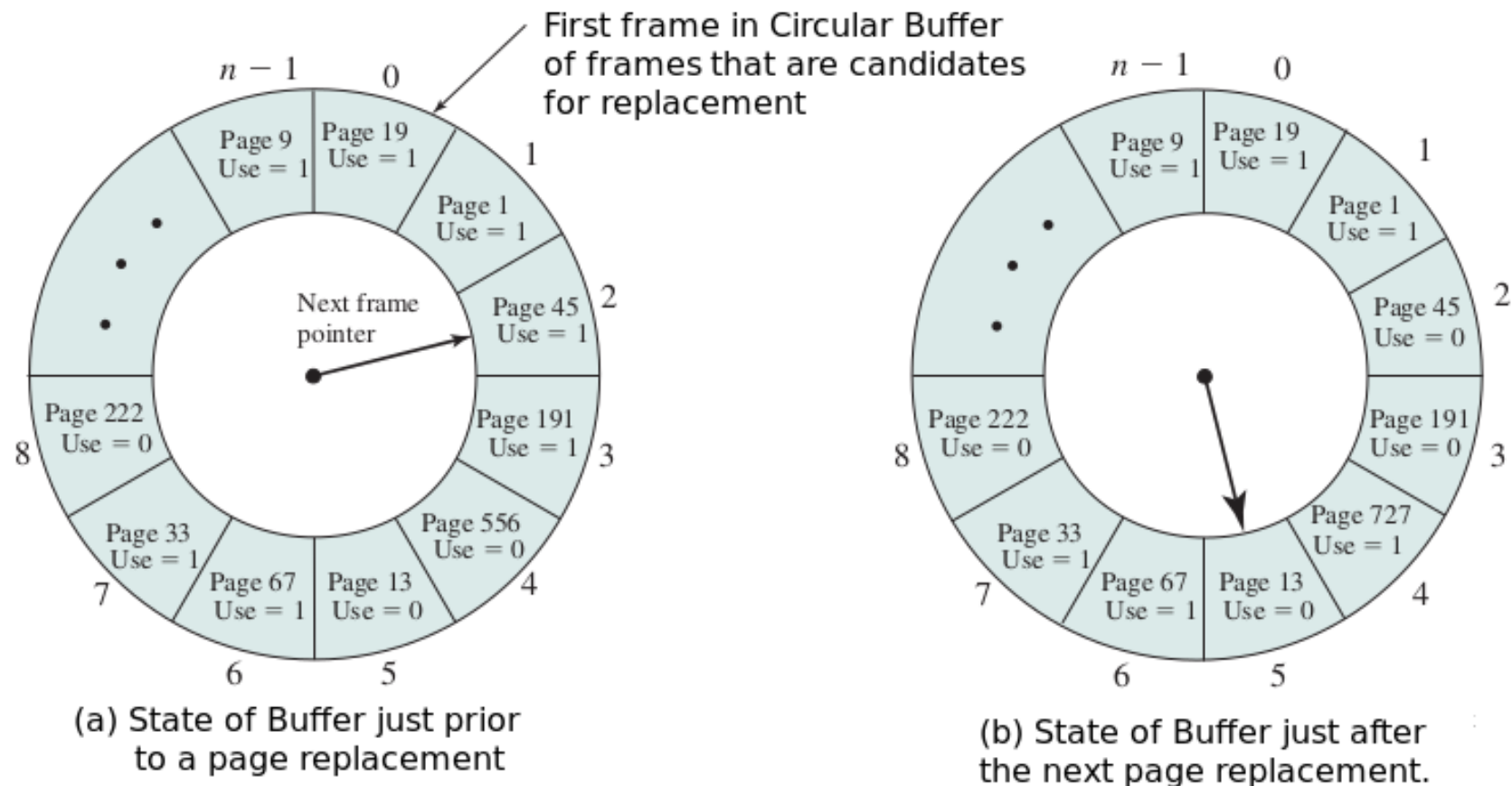
- .. pouco antes da substituição de uma página do buffer pela página 727 de entrada, o próximo ponteiro do “frame” aponta para o “frame” 2, que contém a página 45 – “clock” pode ser executado.



## 8 – Virtual Memory / 8.2 – Operating System Software

### ... 8.2.3 – Replacement Policy

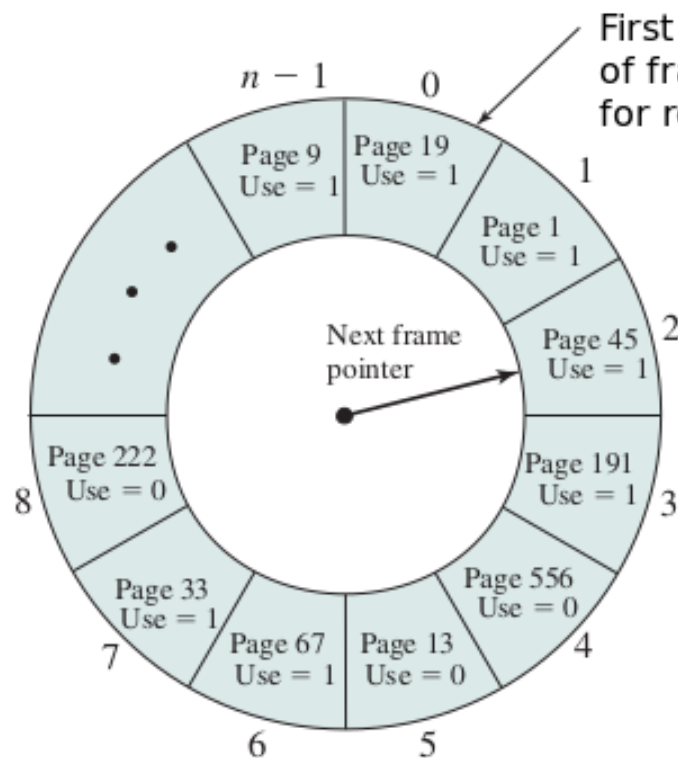
- .. como o “use bit” para a página 45 no “frame” 2 é igual a 1, esta página não é substituída, mas em vez disso, o “use bit” é definido como 0 e o ponteiro avança.



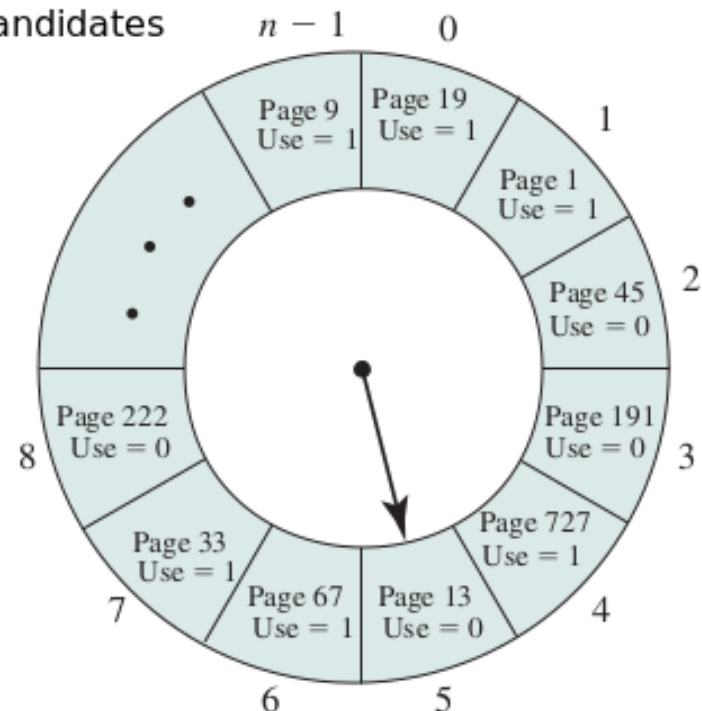
## 8 – Virtual Memory / 8.2 – Operating System Software

### ... 8.2.3 – Replacement Policy

- .. da mesma forma, a página 191 no “frame” 3 não é substituída, mas seu “use bit” passa a ser 0 e o ponteiro avança, para que no próximo “frame”, “frame” 4, com “use bit” = 0 a página é substituída.



(a) State of Buffer just prior to a page replacement

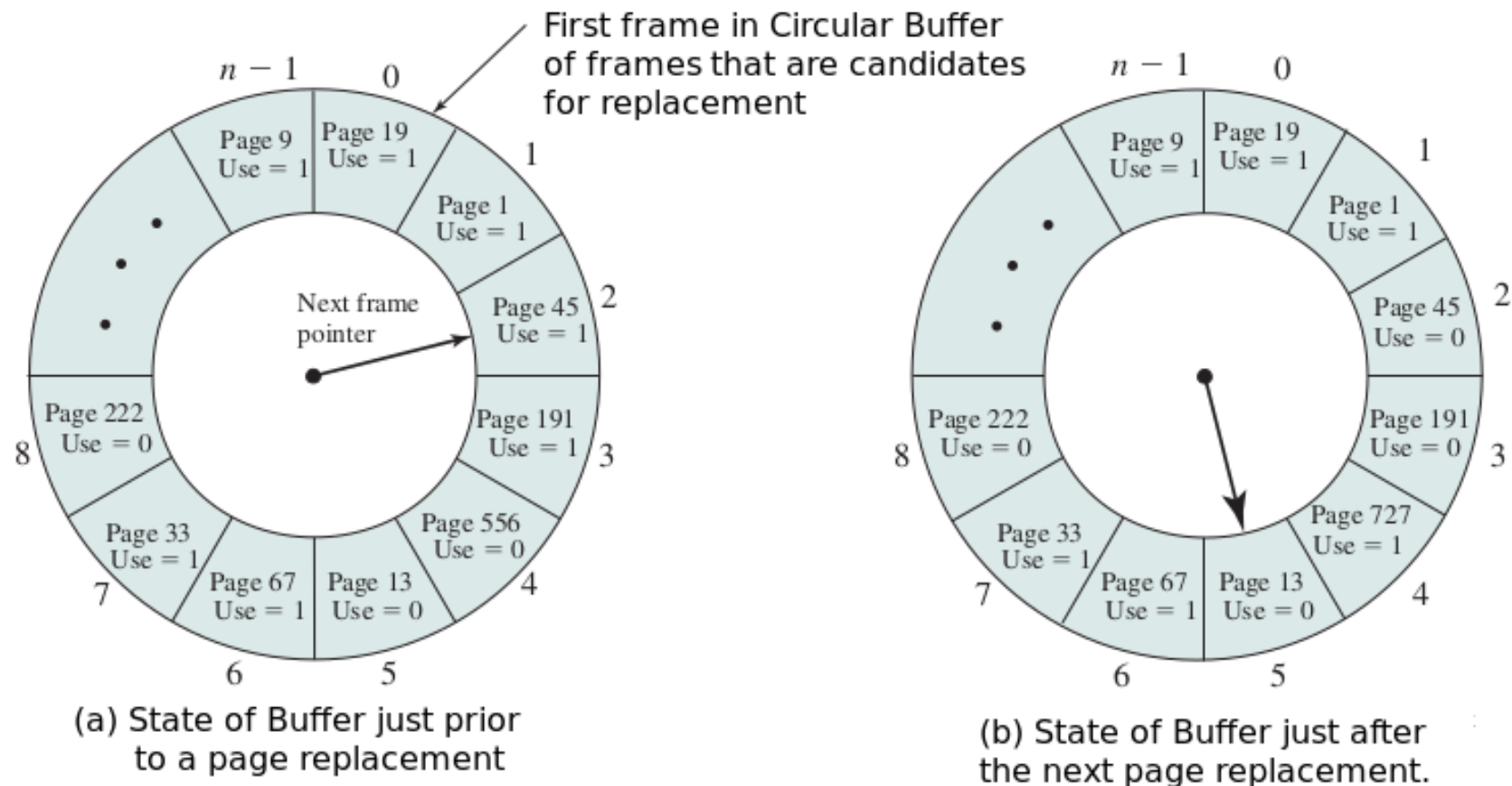


(b) State of Buffer just after the next page replacement.

## 8 – Virtual Memory / 8.2 – Operating System Software

### ... 8.2.3 – Replacement Policy

- .. portanto, a página 556 é substituída pela página 727, o “use bit” é definido como 1 para este “frame” e o ponteiro avança para o “frame” 5, completando o procedimento de substituição de página.

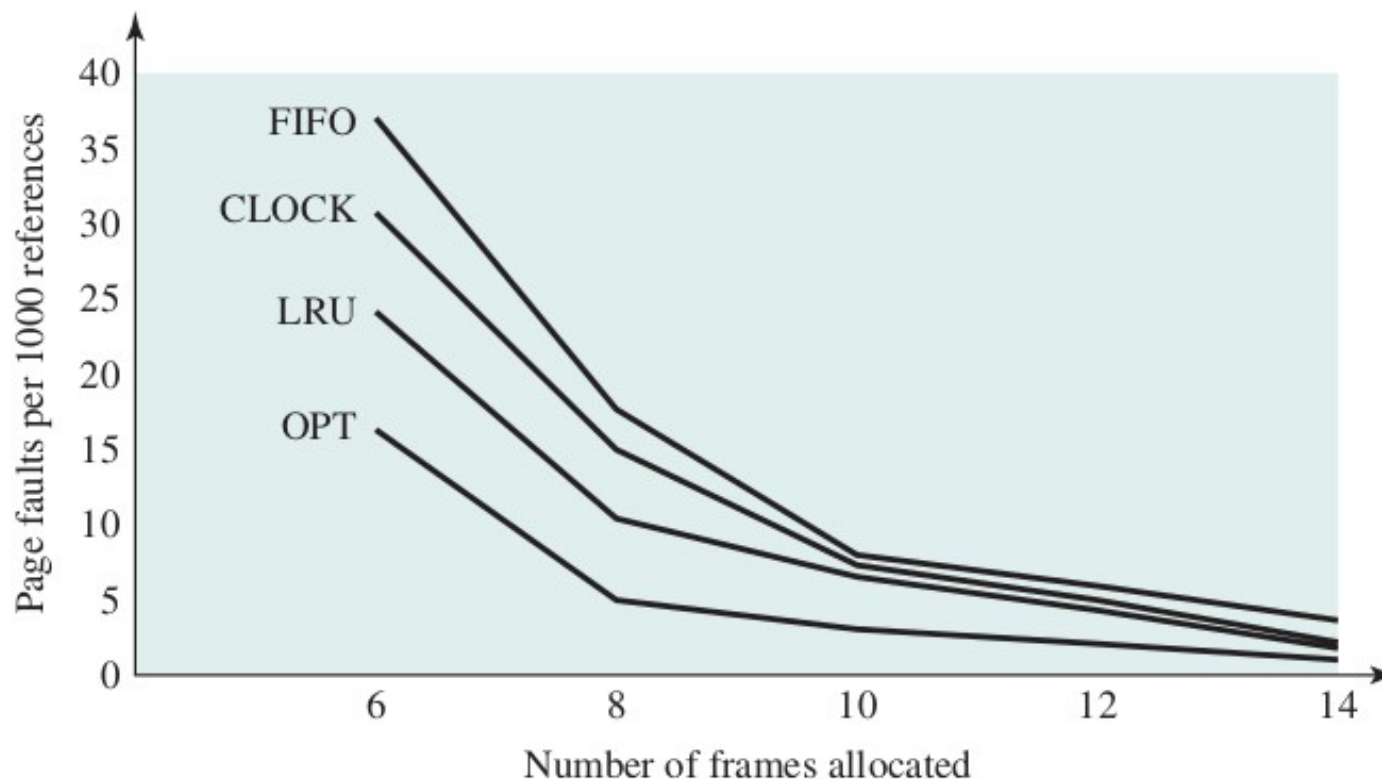




## 8 – Virtual Memory / 8.2 – Operating System Software

### ... 8.2.3 – Replacement Policy

- e.g., figura abaixo ilustra um experimento que mostra os resultados de um experimento relatado em [BAER80], que compara os 04 algoritmos e no qual o nro. de “frames” atribuídos a um processo seja fixo.

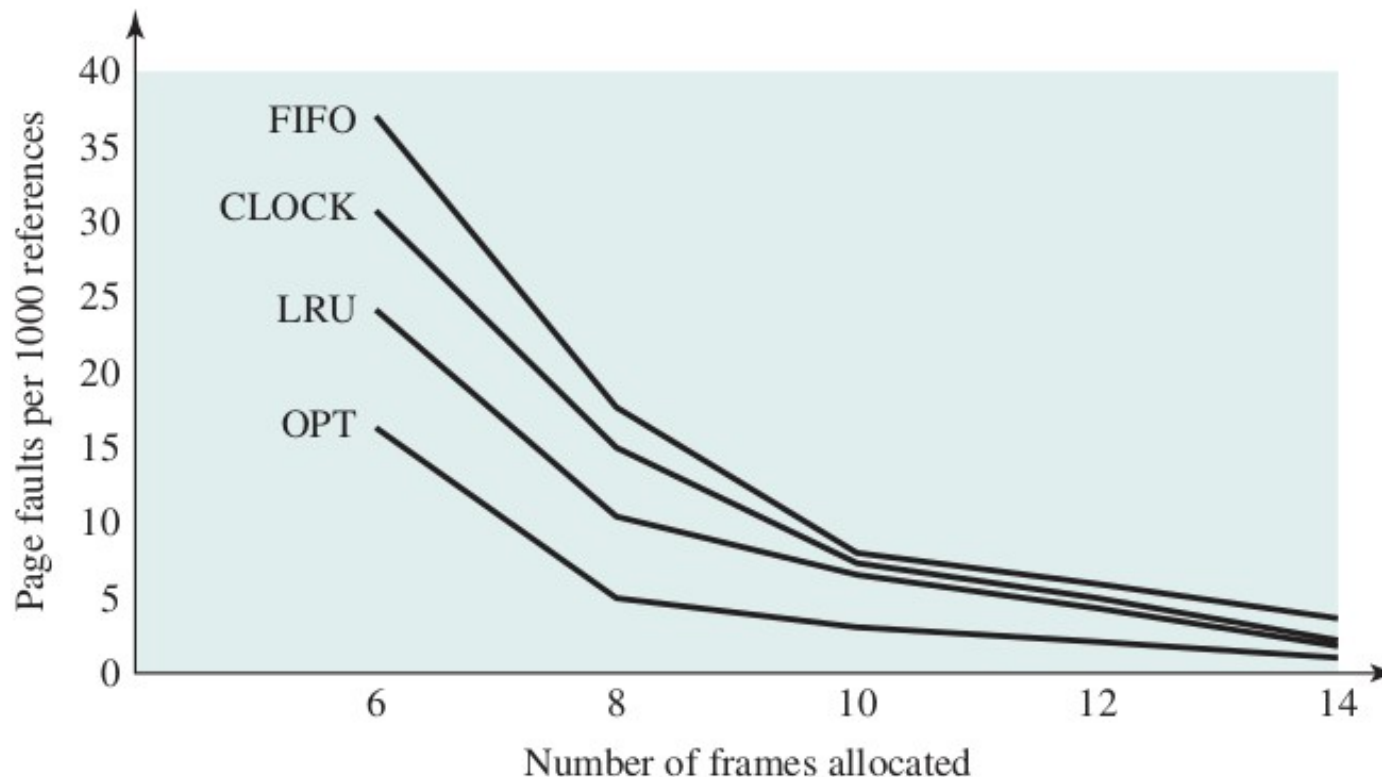




## 8 – Virtual Memory / 8.2 – Operating System Software

### ... 8.2.3 – Replacement Policy

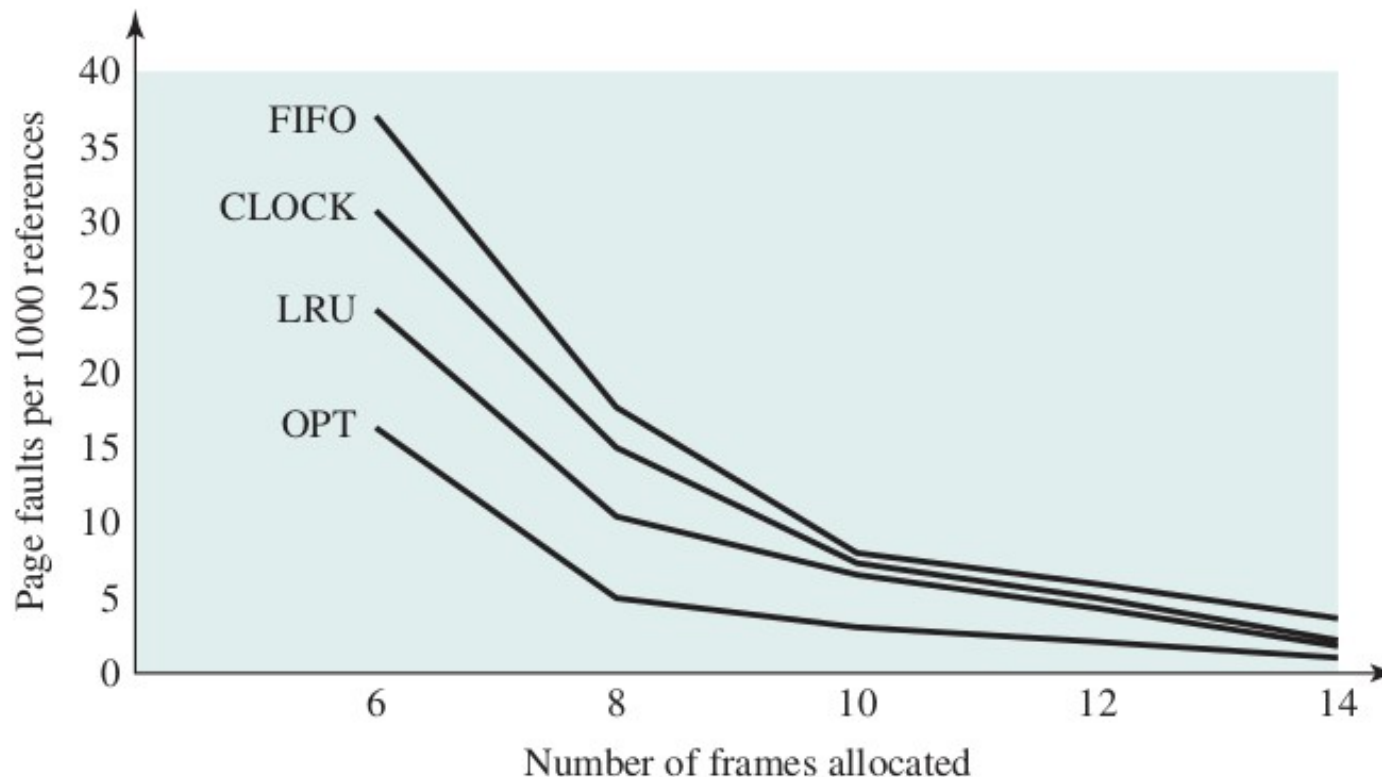
- .. resultados são baseados na execução de  $0,25 \times 10^6$ , ou seja,  $250 \times 10^3$  referências em um programa FORTRAN, usando um tamanho de página de 256 palavras.



## 8 – Virtual Memory / 8.2 – Operating System Software

### ... 8.2.3 – Replacement Policy

- .. executa-se o experimento com alocações de quadros de 6, 8, 10, 12 e 14 quadros e as diferenças entre as 04 políticas são marcantes em pequenas alocações, com FIFO 02 vezes pior do que o ideal.



### 8.2.4 – Resident Set Management

- **“resident set size”** .. com a memória virtual paginada, não é necessário e, de fato, pode não ser possível trazer todas as páginas de um processo para a memória principal para prepará-lo para execução.
- .. assim, o sistema operacional deve decidir quantas páginas trazer, ou seja, quanta memória principal alocar para um determinado processo.
- **“several factors”** .. vários aspectos devem ser considerados:
- 1) quanto menor for a quantidade de memória alocada a um processo, mais processos poderão residir de uma só vez na memória principal.
- 2) se o nro. relativo de páginas de um processo mantidas em memória for baixo, a taxa com que ocorreram faltas de páginas será alta.
- 3) depois de um dado tamanho, alocações adicionais de memória para um processo não irão diminuir a taxa de faltas de páginas.

## 8 – Virtual Memory / 8.2 – Operating System Software

### ... 8.2.4 – Resident Set Management

- **“fixed-allocation policy”** .. uma política de alocação fixa fornece a um processo um número fixo de “frames” na memória principal dentro dos quais ele pode ser executado.
- .. esse número é decidido no momento da criação do processo e pode ser determinado com base no tipo de processo, p.ex., interativo, lote, ou pode ser baseado na orientação do programador ou sistema.
- .. com uma política de alocação fixa, sempre que ocorrer uma falha de página na execução de um processo, uma das páginas desse processo deve ser substituída pela página necessária.

## 8 – Virtual Memory / 8.2 – Operating System Software

### ... 8.2.4 – Resident Set Management

- “**variable-allocation policy**” .. uma política de alocação variável permite que o número de “page frames” alocados a um processo varie ao longo do tempo de vida do processo.
- .. idealmente, um processo que está sofrendo de altos níveis de falta de página, indicando que o princípio da localidade só se mantém em uma forma fraca, receberá frames de página adicionais.
- .. já um processo com uma taxa de falta de página excepcionalmente baixa, indicando que o processo está se comportando bem do ponto de vista da localidade, receberá uma alocação reduzida.
- .. uso de uma política de alocação variável está relacionado ao conceito de escopo de substituição, conforme explicado na próxima subseção.

## 8 – Virtual Memory / 8.2 – Operating System Software

### ... 8.2.4 – Resident Set Management

- “**variable-allocation policy**” .. parece ser mais robusta, no entanto, a dificuldade dessa abordagem é que ela exige que o sist. oper. avalie o comportamento dos processos ativos.
- .. isso inevitavelmente requer sobrecarga de software no sistema operacional e depende dos mecanismos de “hardware” fornecidos pela plataforma do processador.

## 8 – Virtual Memory / 8.2 – Operating System Software

### ... 8.2.4 – Resident Set Management

- **“replacement scope”** .. escopo de uma estratégia de substituição pode ser categorizado como global ou local.
- .. ambos os tipos de políticas são ativados por uma falta de página quando não há “page frames” livres.
- **“local replacement policy”** .. uma política de substituição local escolhe apenas entre as páginas residentes do processo que gerou a falta de página ao selecionar uma página para substituir.
- **“global replacement policy”** .. considera todas as páginas desbloqueadas na memória principal como candidatas para substituição, independentemente de qual processo possui uma página específica.

## 8 – Virtual Memory / 8.2 – Operating System Software

### ... 8.2.4 – Resident Set Management

- **“local replacement policy”** .. uma política de substituição local escolhe apenas entre as páginas residentes do processo que gerou a falta de página ao selecionar uma página para substituir.
- **“global replacement policy”** .. considera todas as páginas desbloqueadas na memória principal como candidatas para substituição, independentemente de qual processo possui uma página específica.
- **“análises e considerações”** ..
  - .. embora as políticas locais sejam mais fáceis de analisar, não há evidências convincentes de que tenham um desempenho melhor do que as políticas globais.
  - .. já as políticas globais são atraentes em razão de sua simplicidade de implementação e sobrecarga mínima.



## 8 – Virtual Memory / 8.2 – Operating System Software

### ... 8.2.4 – Resident Set Management

- “**correlation**” .. entre “replacement scope” e “resident set size”
- “**fixed resident set**” .. um conjunto de “frames” residentes fixos implica uma política de substituição local ..
- .. p.ex., para manter o tamanho de um conjunto residente fixo, uma página que é removida da memória principal deve ser substituída por outra página do mesmo processo.

	Local Replacement	Global Replacement
Fixed Allocation	<ul style="list-style-type: none"><li>• Number of frames allocated to a process is fixed.</li><li>• Page to be replaced is chosen from among the frames allocated to that process.</li></ul>	<ul style="list-style-type: none"><li>• Not possible.</li></ul>
Variable Allocation	<ul style="list-style-type: none"><li>• The number of frames allocated to a process may be changed from time to time to maintain the working set of the process.</li><li>• Page to be replaced is chosen from among the frames allocated to that process.</li></ul>	<ul style="list-style-type: none"><li>• Page to be replaced is chosen from all available frames in main memory; this causes the size of the resident set of processes to vary.</li></ul>

## 8 – Virtual Memory / 8.2 – Operating System Software

### ... 8.2.4 – Resident Set Management

- “**variable-allocation policy**” .. uma política de alocação variável pode claramente empregar uma política de substituição global.
- .. p.ex., substituição de uma página de um processo na memória principal por outra faz com que a alocação de um processo aumente em 01 página e a do outro diminua em 01 página.

	Local Replacement	Global Replacement
Fixed Allocation	<ul style="list-style-type: none"><li>• Number of frames allocated to a process is fixed.</li><li>• Page to be replaced is chosen from among the frames allocated to that process.</li></ul>	<ul style="list-style-type: none"><li>• Not possible.</li></ul>
Variable Allocation	<ul style="list-style-type: none"><li>• The number of frames allocated to a process may be changed from time to time to maintain the working set of the process.</li><li>• Page to be replaced is chosen from among the frames allocated to that process.</li></ul>	<ul style="list-style-type: none"><li>• Page to be replaced is chosen from all available frames in main memory; this causes the size of the resident set of processes to vary.</li></ul>

### 8.2.5 – Cleaning Policy

- **“cleaning policy”** .. uma política de limpeza é o oposto de uma política de busca, posto que se preocupa em determinar quando uma página modificada deve ser gravada na memória secundária.
- **“duas alternativas”** .. “demand cleaning” e “precleaning”
- **“demand cleaning”** .. uma página é gravada na memória secundária apenas quando é selecionada para substituição.
- **“precleaning”** .. uma política de pré-limpeza grava páginas modificadas antes que seus quadros de página sejam necessários e, assim, as páginas são gravadas em lotes de páginas.

## 8 – Virtual Memory / 8.2 – Operating System Software

### ... 8.2.5 – Cleaning Policy

- “**execução estrita da política**” .. há algumas implicações / efeitos.
- “**precleaning**” .. uma página é gravada ou salvaguardada, mas permanece na memória principal até que o algoritmo de substituição da página determine que ela seja removida.
- .. “precleaning” permite a gravação de páginas em lotes, mas faz pouco sentido escrever 100s de páginas apenas para descobrir que a maioria delas foi modificada novamente antes de serem substituídas.
- .. capacidade de transferência da memória secundária é limitada e não deve ser desperdiçada com operações de limpeza desnecessárias.

## 8 – Virtual Memory / 8.2 – Operating System Software

### ... 8.2.5 – Cleaning Policy

- **“execução estrita da política”** .. há algumas implicações / efeitos.
- **“demand cleaning”** .. com a limpeza por demanda, a escrita de uma página suja é acoplada e antecede a leitura de uma nova página.
- .. essa técnica pode minimizar as gravações de página, mas significa que um processo que sofre uma falta de página pode ter que esperar por 02 transferências de página antes de ser desbloqueado.
- **“better approach”** .. incorpora o buffer de páginas.
- .. isso permite a adoção da seguinte política, limpe apenas as páginas substituíveis, mas desacople as operações de limpeza e substituição.
- .. com o “buffer” de páginas, as páginas substituídas podem ser colocadas em duas listas, modificadas e não modificadas.

## 8 – Virtual Memory / 8.2 – Operating System Software

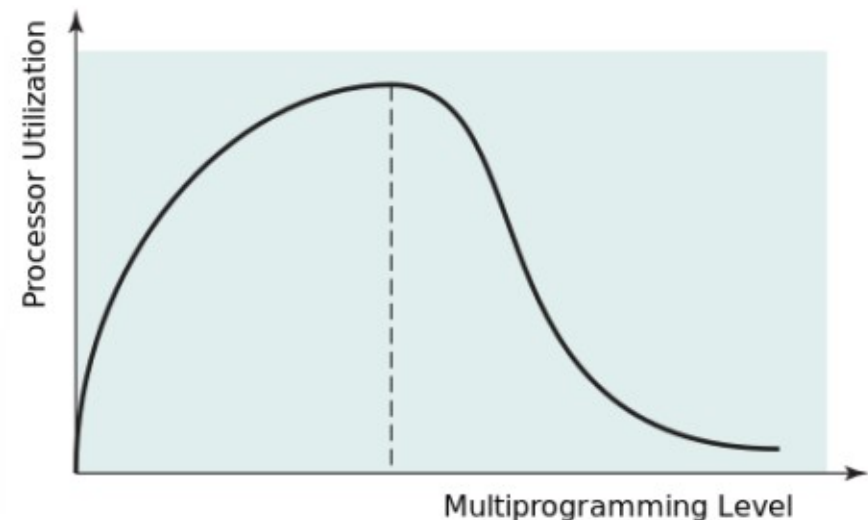
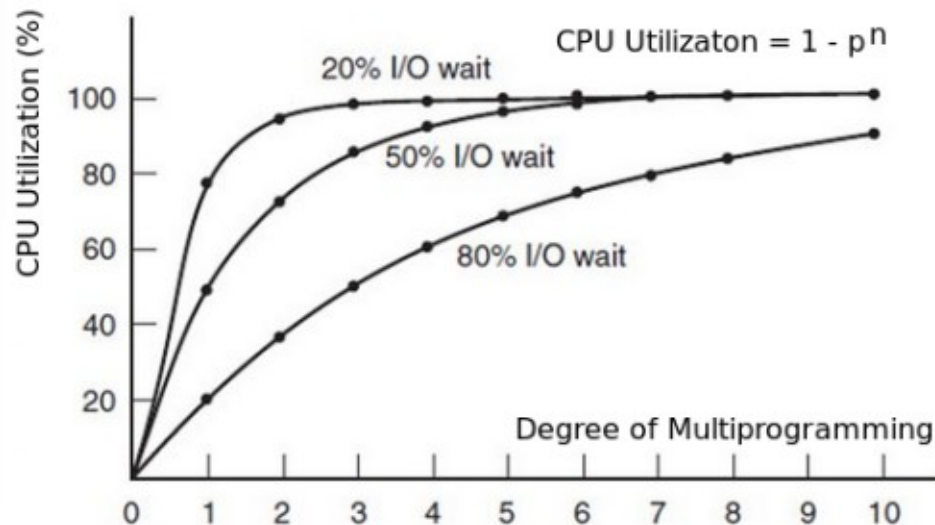
### 8.2.6 – Load Control

- **“load control”** .. está relacionado à determinação do número de processos que ficarão residentes na memória principal, o que é conhecido como “nível de multiprogramação”.
- **“load control policy”** .. crítica no gerenciamento eficaz de memória.
- e.g., se poucos processos estiverem residentes ao mesmo tempo, haverá muitas ocasiões em que todos os processos estão bloqueados, e muito tempo será gasto na troca.
- e.g. se muitos processos forem residentes, então, em média, o tamanho do conjunto residente de cada processo será inadequado e ocorrerão faltas frequentes de páginas » resultado também é ruim.

## 8 – Virtual Memory / 8.2 – Operating System Software

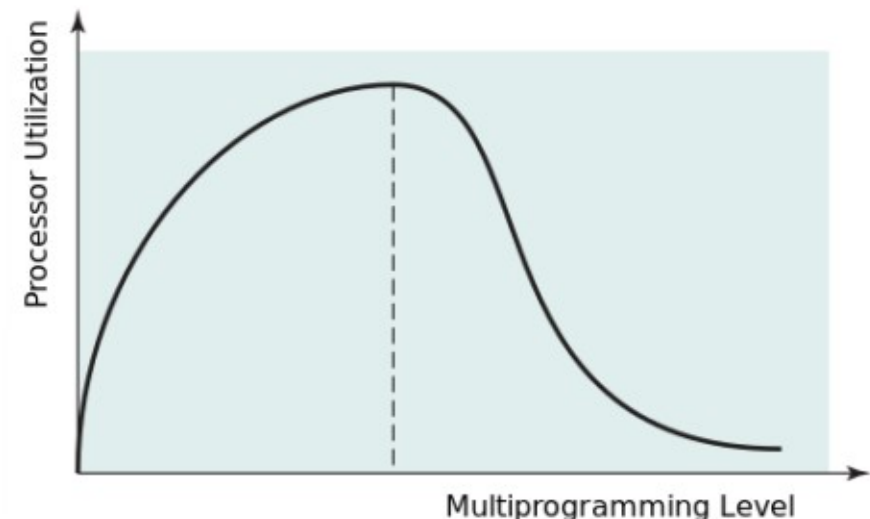
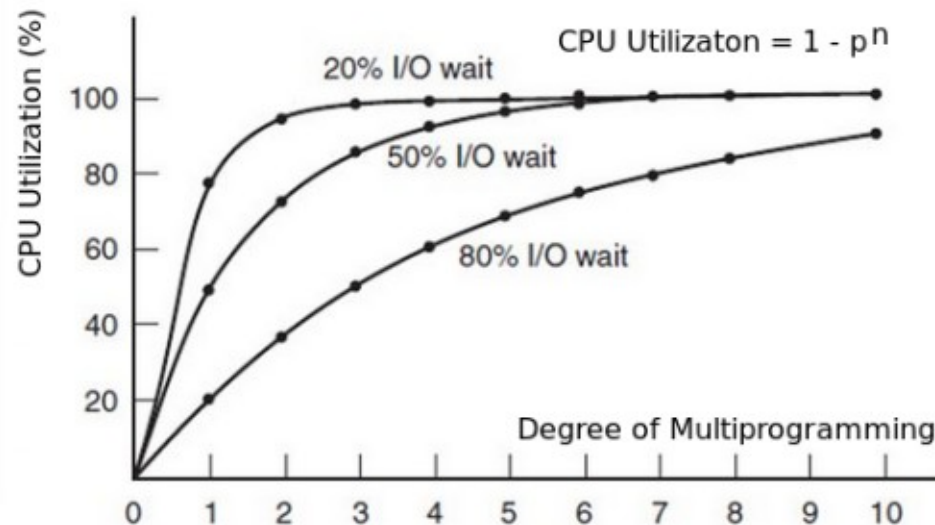
### ... 8.2.6 – Load Control

- **“approach this problem”** .. “working set” ou “Page Fault Frequency” (PFF) incorporam implicitamente o controle de carga.
- .. neste caso, apenas processos cujo “working set” é suficientemente grande podem ser executados.
- .. ao fornecer o tamanho do “working set” necessário para cada processo ativo, a política determina automática e dinamicamente o número de instâncias de programas ativos.



## 8 – Virtual Memory / 8.2 – Operating System Software ... 8.2.6 – Load Control

- “**multiprogramming level**” .. conforme o aumento do nível de multiprogramação, a expectativa é de um aumento na utilização do processador, porque há menos chance de processos bloqueados.
- .. no entanto, lança-se um ponto em que o conjunto médio de residentes é inadequado – máximo da curva.
- .. nesse ponto, o número de falhas de página aumenta drasticamente e a utilização do processador diminui .. decréscimo exponencial.





## 8 – Virtual Memory / 8.2 – Operating System Software

### ... 8.2.6 – Load Control

- “**multiprogramming level**” .. se o grau da multiprogramação for reduzido, um ou mais processos residentes em memória principal devem ser suspensos - “swapped out”.
- .. entretanto, outras possibilidades incluem a suspensão de:
- processos com baixa prioridade através de política de escalonamento.
- processos com o menor conjunto de páginas residentes, pois penaliza programa cujo princípio da localidade está restrito a poucas páginas.
- processos grandes, pois liberam muitos frames na memória alocada tornando improváveis desativações adicionais.

## 8 – Virtual Memory / 8.3 – UNIX and Solaris Memory Management

### 8.3 – UNIX and Solaris Memory Management

- .. LEITURA COMPLEMENTAR !

## 8 – Virtual Memory / 8.4 – LINUX Memory Management

### 8.4 – LINUX Memory Management

- .. LEITURA COMPLEMENTAR !

## 8 – Virtual Memory / 8.5 – WINDOWS Memory Management

### 8.5 – WINDOWS Memory Management

- .. LEITURA COMPLEMENTAR !