

UFRJ – IM - DCC



Sistemas Operacionais I

Unidade III – Memória Virtual



ORGANIZAÇÃO DA UNIDADE

- Processador
- Memória Primária
- Memória Virtual
 - Fundamentos
 - Organização Lógica
 - Processos de Mapeamento
 - Algoritmos de Re-alocação
 - Estratégias de Gerenciamento



Consiste na utilização de espaços do disco rígido como extensão lógica da memória primária.

- ❖ A memória virtual é transparente para o programador e para o processador.*
- ❖ A memória virtual expande o tamanho da memória primária.*
- ❖ A memória virtual não é ilimitada.*
- ❖ O sistema ganha em flexibilidade e perde em termos de desempenho.*



Gerenciamento de Recursos I

Características

- ***Todas as referências a memória passam a ser com endereços lógicos virtuais (VA), que são traduzidos em endereços físicos, em tempo de execução.***
- ***Uma tarefa é dividida em partes (páginas ou segmentos), não necessariamente localizados em áreas contíguas na memória.***
- ***Com a memória virtual acaba a necessidade de todas as partes de uma tarefa estarem carregadas em memória primária.***
- ***Uma tarefa pode ocupar diferentes áreas de memória durante a sua execução***
- ***O uso da memória virtual é transparente ao usuário e à própria CPU***



Gerenciamento de Recursos I

Características

- ***Maior tempo de resposta para as referências à memória.***
- ***Maior complexidade do hardware e do esquema de gerenciamento.***
- ***Impossibilidade de estimar de forma precisa e segura, o tempo a ser gasto em qualquer referência à memória.***
- ***Uma mesma referência à memória pode consumir tempos diferentes de execução.***



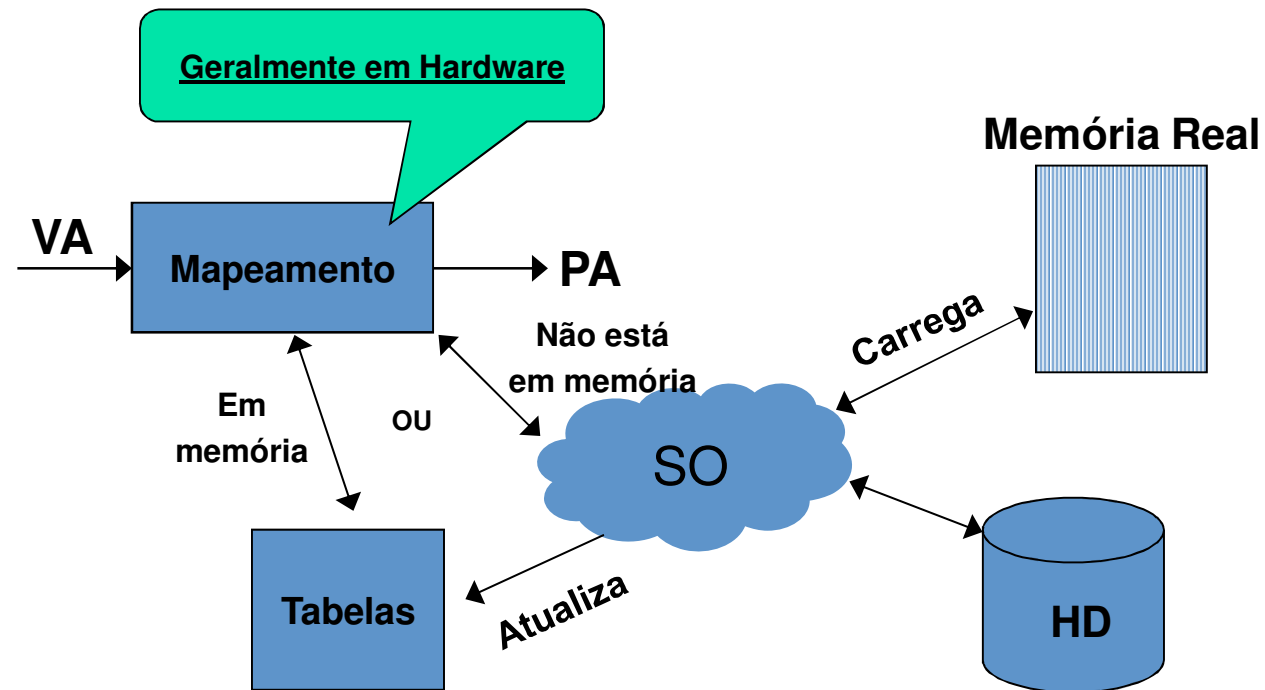
- ***Mais processos mantidos em MP***
 - ***os processos são carregados parcialmente***
 - ***maior eficiência na utilização do processador***
- ***Processos podem ser maiores que a memória principal***
 - ***Todo programador tem disponível uma memória de trabalho (virtual) de tamanho igual a todo espaço de endereçamento disponível.***
- ***O SO se encarrega de trazer para a memória física as partes necessárias para a execução do programa.***



Gerenciamento de Recursos I

Endereçamento

- ❖ Cada referência virtual é convertida para o endereçamento físico em tempo de execução. Este processo de conversão é chamado de mapeamento.





Gerenciamento de Recursos I

Suporte Necessário

- Hardware:
 - Tradução (mapeamento) eficiente de endereços.
 - Movimentação eficaz dos trechos de informação entre a memória virtual e a física.
- SO:
 - Re-alocação eficiente da memória física.
 - Princípio da Localidade x Trashing

Memória virtual \neq Swapping



Gerenciamento de Recursos I

Princípio da Localidade

As referências de memória tendem a ser agrupadas em termos espaciais e/ou temporais.

“Em um programa bem construído, a probabilidade média da próxima instrução a ser executada ser a seguinte ou estar nas redondezas da instrução corrente é mais elevada do que outras hipóteses.”

- **Somente alguns trechos do código são necessários para a execução num curto espaço de tempo;**
- **É possível ter uma razoável noção dos trechos de código que serão utilizados num futuro próximo, reduzindo os riscos de trashing.**



- **Trashing** é a situação onde o sistema passa a maior parte do tempo removendo e trazendo partes de processos em lugar de executando instruções dos mesmos.
- A memória normalmente está toda ocupada com partes de diversos processos.
- Quando o SO precisa carregar uma nova parte, um outro pode precisar ser removido para abrir espaço.
- Se for removida uma parte que seja referenciada logo a seguir, esta precisará ser carregada novamente.



Gerenciamento de Recursos I

Organização Lógica

- Organização Lógica da Memória
- *Paginada*
 - *Segmentada*
 - *Segmentada com Paginação*

- Processo de Mapeamento
- Unidade de Conversão VA → PA
 - Tabelas de Referência
 - Cache para Tabelas de Referência (TLB)

- Gerenciamento
- Estratégias de Busca
 - Estratégias de Alocação
 - Estratégias de Realocação
 - Estratégias de Definição da Área de Trabalho
 - Estratégias de Limpeza
 - Estratégias de Carga



Gerenciamento de Recursos I

Paginação

- Cada processo tem sua própria tabela de páginas
- Cada entrada contém um bit de presença (P) indicando se a página se encontra na memória física ou não
- Se a página está presente, a entrada contém o número da moldura da página correspondente
- Cada entrada também contém um bit de modificação (M)
páginas não modificadas não precisam ser gravadas em disco quando removidas

Endereço Virtual

Número Página	Offset
---------------	--------

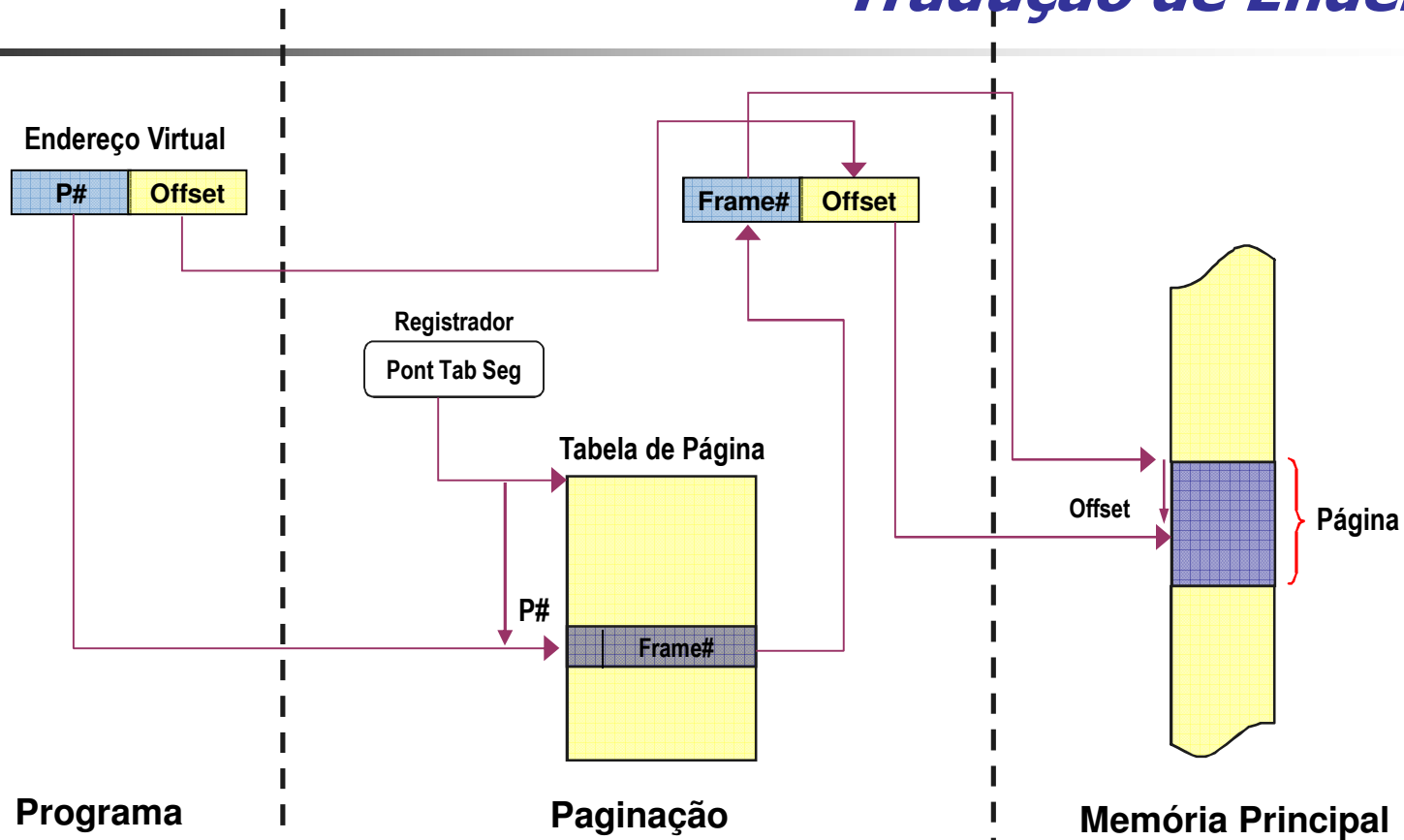
Entrada da Tabela de Página

P	M	Outros bits controle	Número Moldura
---	---	----------------------	----------------



Gerenciamento de Recursos I

Tradução de Endereço



- ☑ **Problema:** → Espaço ocupado pela PT (proporcional ao tamanho da memória virtual)
- ☑ **Solução:** → Tabela de páginas invertidas
Múltiplas tabelas de página



Gerenciamento de Recursos I

Múltiplas Tabelas

- ▶ As tabelas de páginas são mantidas em memória
- ▶ O espaço de endereçamento virtual pode ser muito grande ou esparso: a tabela de páginas pode ocupar muito espaço na memória

⇒ Solução: as tabelas de páginas são mantidas na memória virtual

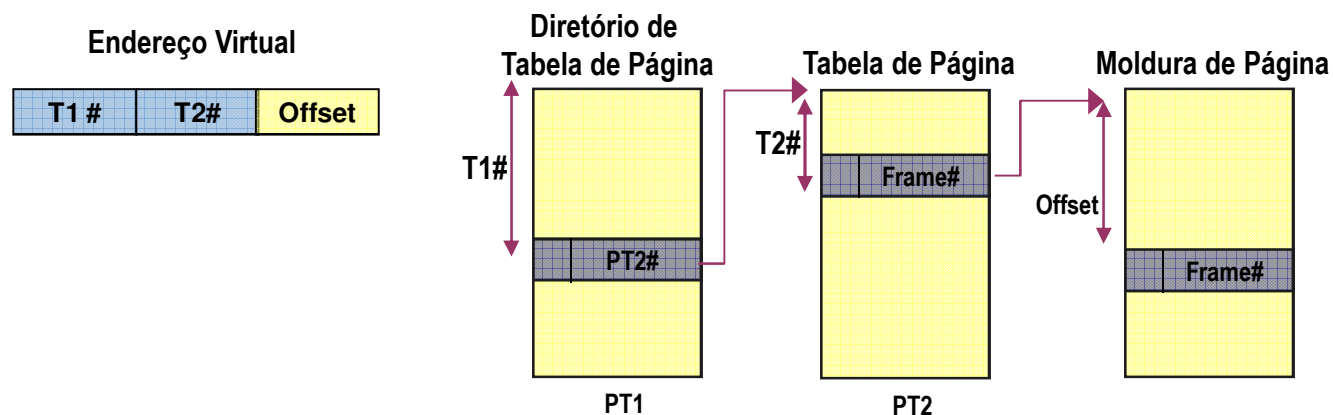
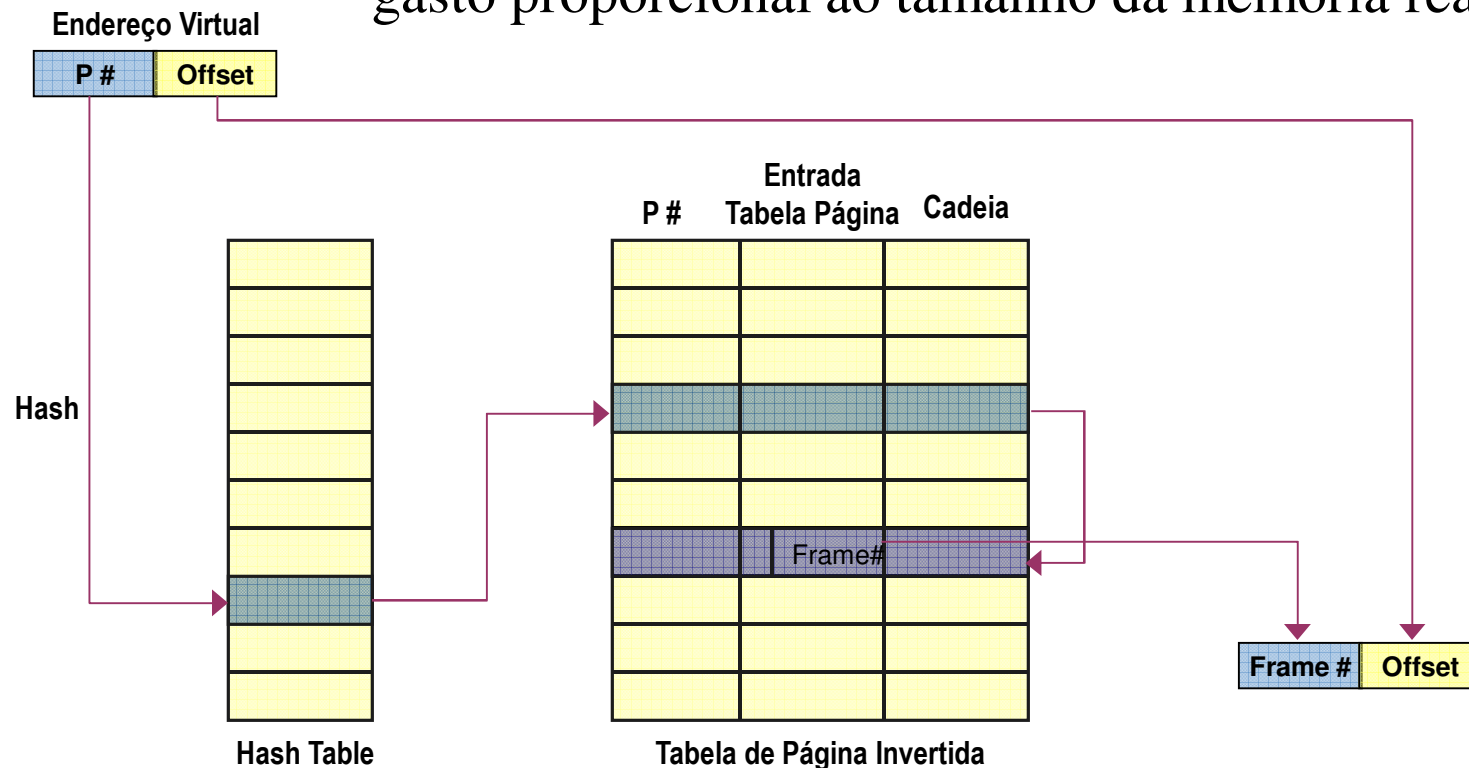




Tabela Invertida

- Técnica para reduzir o gasto com tabelas de páginas
 - gasto proporcional ao tamanho da memória real



(*) Número de entradas na Tabela Invertida é aproximadamente igual ao número de páginas na memória real



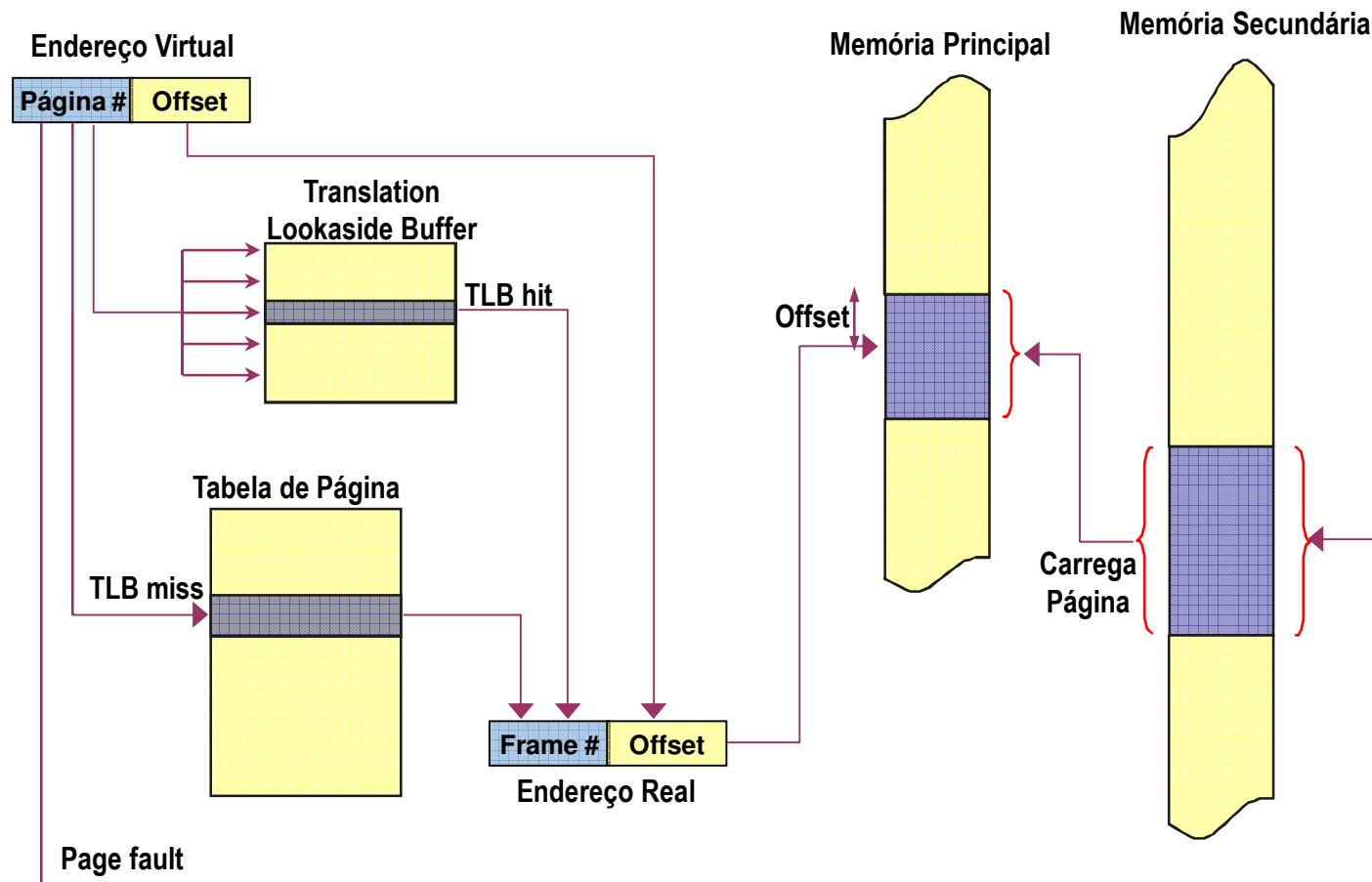
Transaction Lookaside Buffer - TLB

- **Cada referência à memória virtual pode gerar dois ou mais acessos à memória real:**
 - 1 - Busca a entrada da tabela de página apropriada
 - 2 - Busca o dado necessário
- **Uso de uma cache para entradas da tabela de páginas (TLB) aumenta o desempenho**
 - Contém as entradas usadas mais recentemente
 - número da página
 - número da moldura (frame)
 - bit de modificação e outros bits de controle
 - Funciona como memória cache



Gerenciamento de Recursos I

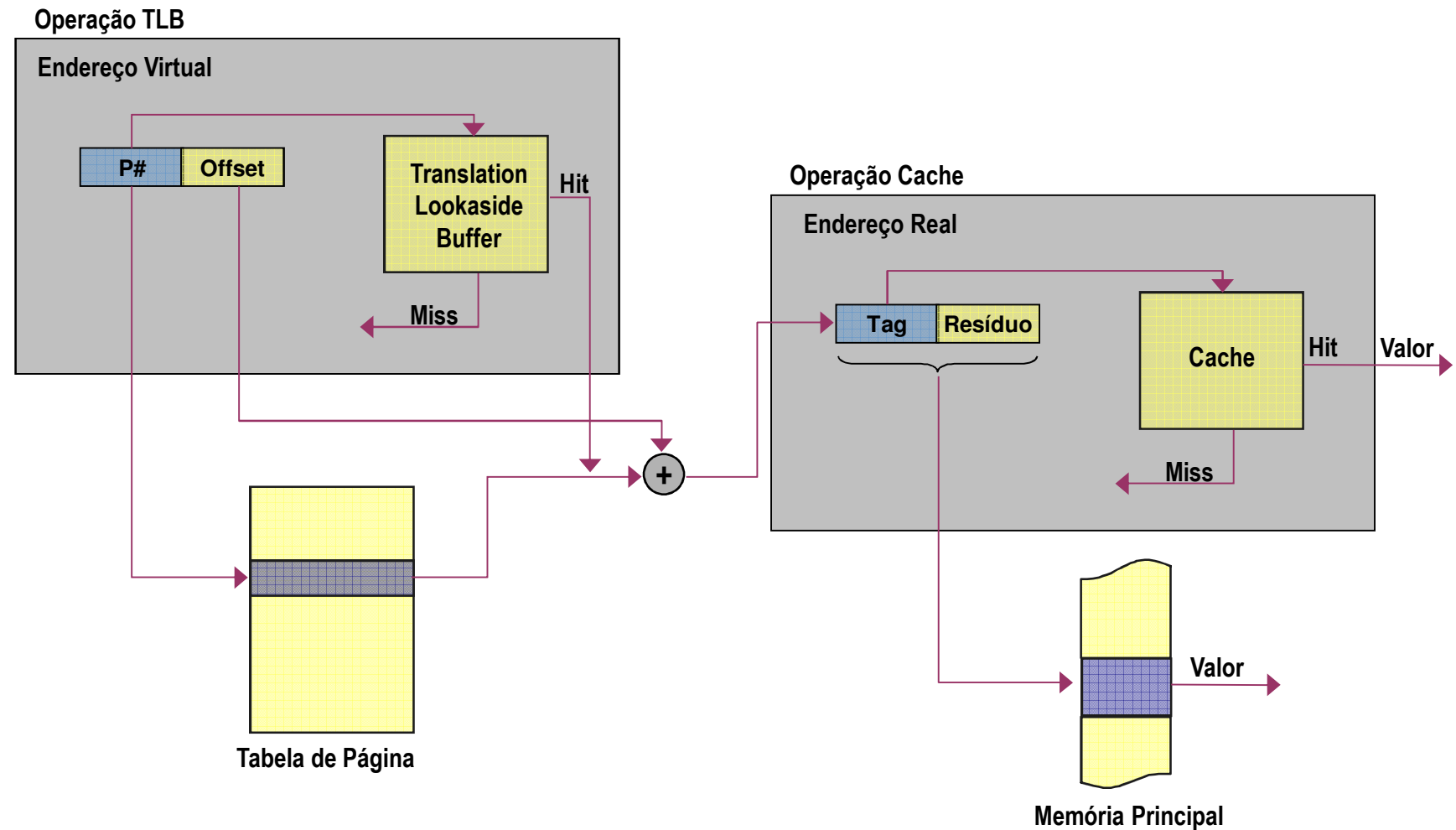
Uso da TLB





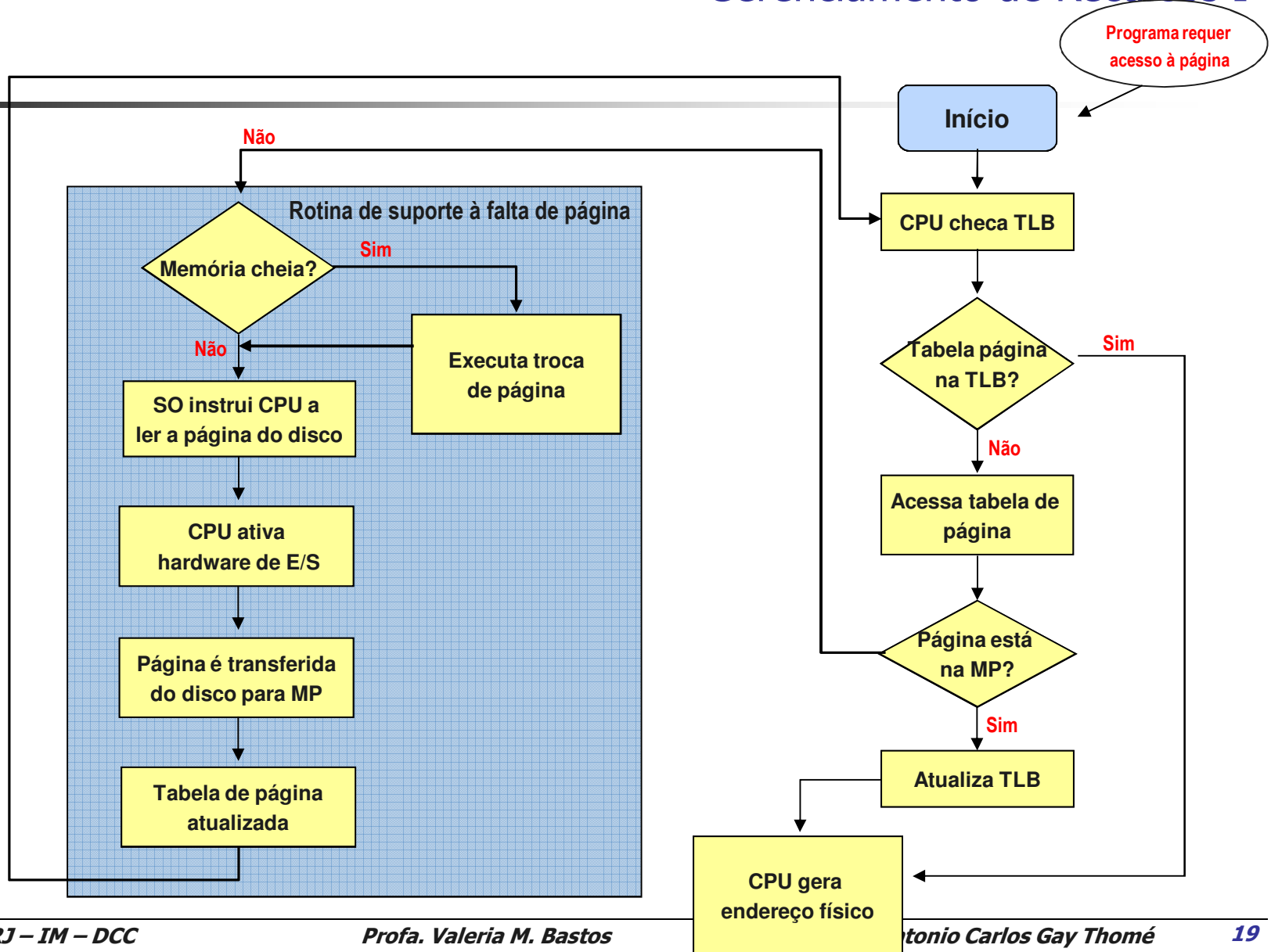
Gerenciamento de Recursos I

TLB + Cache





Gerenciamento de Recursos I





Gerenciamento de Recursos I

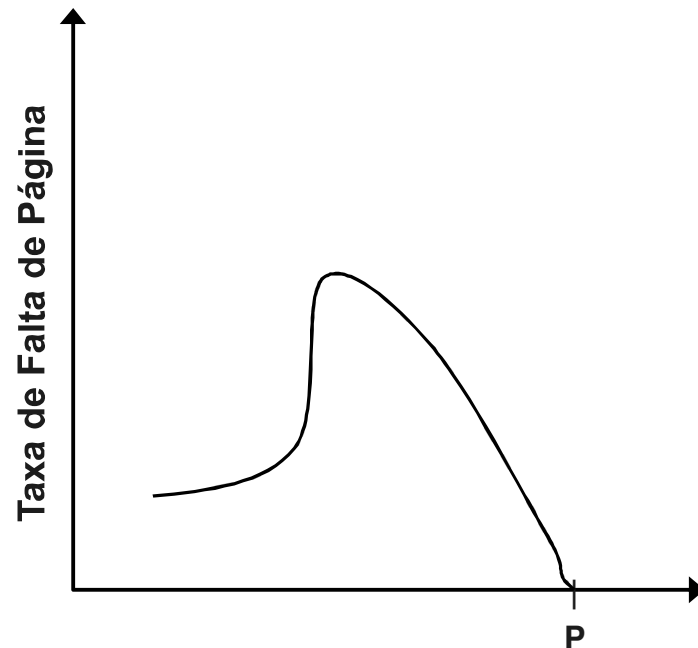
Tamanho da Página

- ▤ Quanto menor a página = Menor a fragmentação interna
- ▤ Quanto menor a página = Mais páginas por processo
 - ▤ maior a tabela de páginas
 - ▤ mais tabelas na memória secundária
 - ▤ mais falta de páginas (page faults)
- ▤ A transferência de dados com a memória secundária é mais eficiente com blocos maiores
- ▤ O número de falta de páginas (page faults):
 - ▤ diminui à medida que o tamanho da página aumenta (até certo ponto)
 - ▤ depois deste ponto, começa a baixar (fenômeno da saturação)

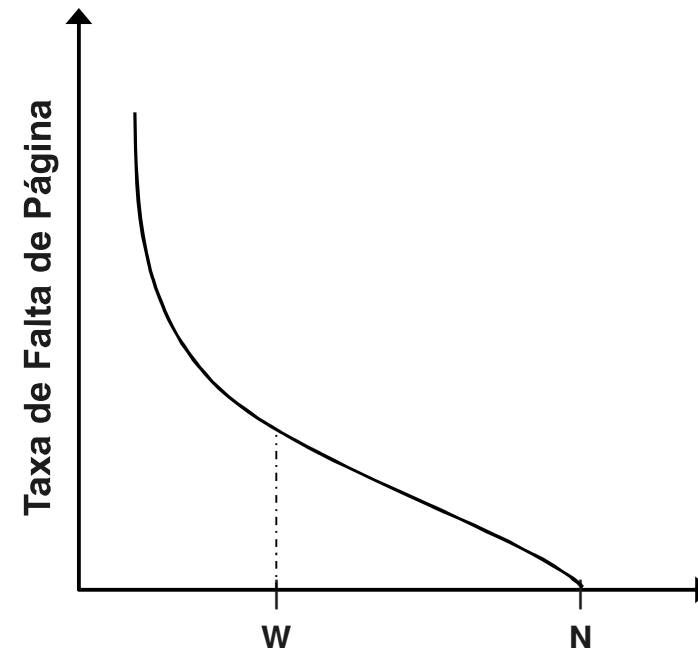


Gerenciamento de Recursos I

Comportamento Típico da Paginação



(a) Tamanho da Página



(b) Número de Molduras de Páginas Alocadas



- Aplicativos
 - maior uso de memória
 - menor localidade (programação orientada a objetos, multithreading)
- A TLB precisa ser maior
- Alternativa: páginas de vários tamanhos
 - páginas maiores para regiões de alta localidade
 - páginas menores para regiões de baixa localidade



Gerenciamento de Recursos I

Segmentação

■ Vantagens

- Simplifica o tratamento de estruturas dinâmicas
- Recompilação separada de programas
- Facilita o compartilhamento de memória
- Facilita a proteção apropriada da memória

■ Tabela de segmentos

- Endereço e tamanho do segmento
- Bit de presença
- Bit de modificação

→ **Segmentos têm tamanho variável e dinâmico**

Endereço Virtual

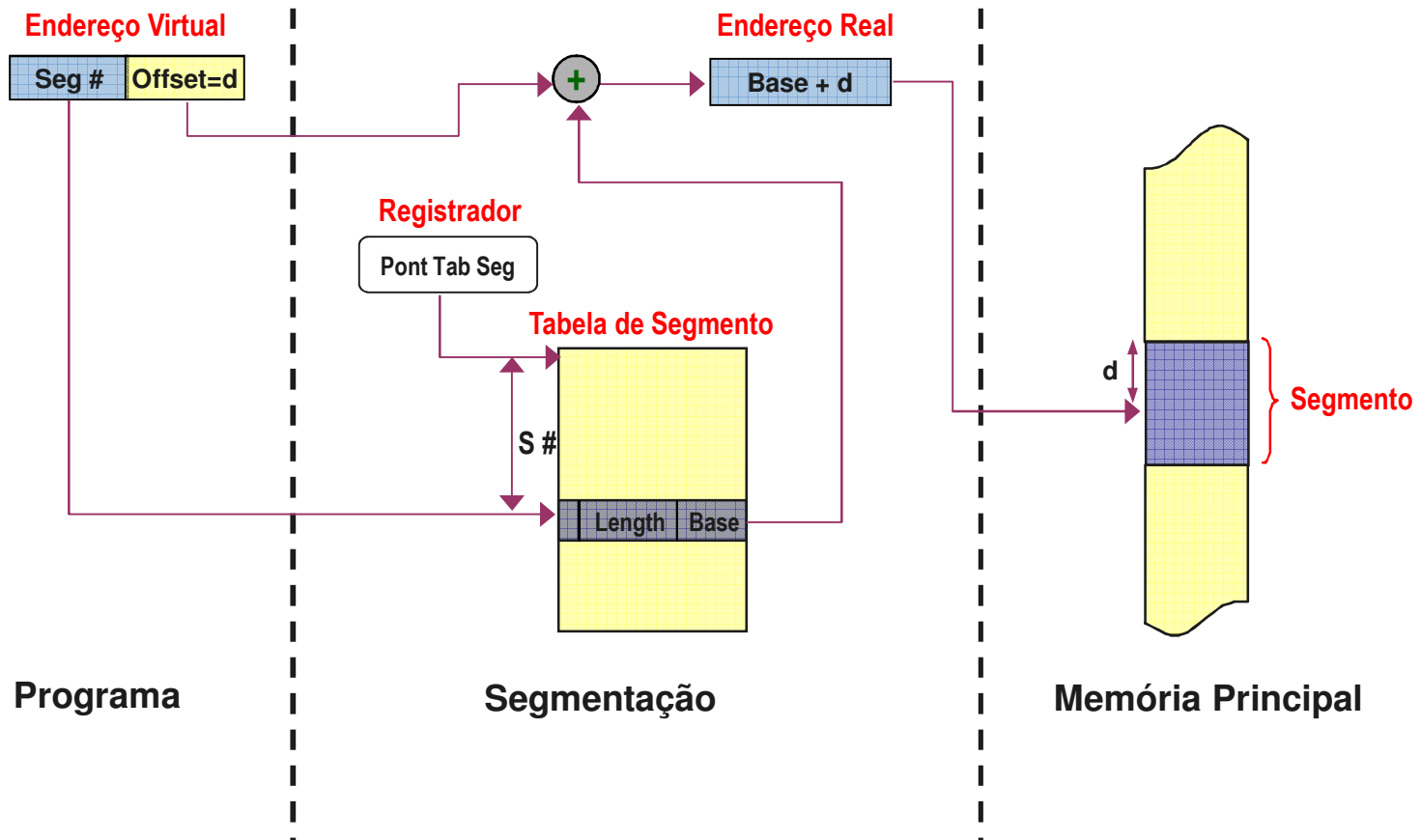
Número Segmento	Offset
-----------------	--------

Tabela de Entrada de Segmento

P	M	Outros bits controle	Tamanho	Base Segmento
---	---	----------------------	---------	---------------



Tradução de Endereços na Segmentação





Gerenciamento de Recursos I

Segmentação Paginada

■ Vantagens

- Paginação elimina a fragmentação externa
- Segmentação permite estruturas dinâmicas, modularidade e suporte para compartilhamento e proteção

→ Cada segmento é quebrado em páginas de tamanho igual

Endereço Virtual

Número Segmento	Número Página	Offset
-----------------	---------------	--------

Tabela de Entrada de Segmento

Outros bits controle	Tamanho	Base Tabela de Página
----------------------	---------	-----------------------

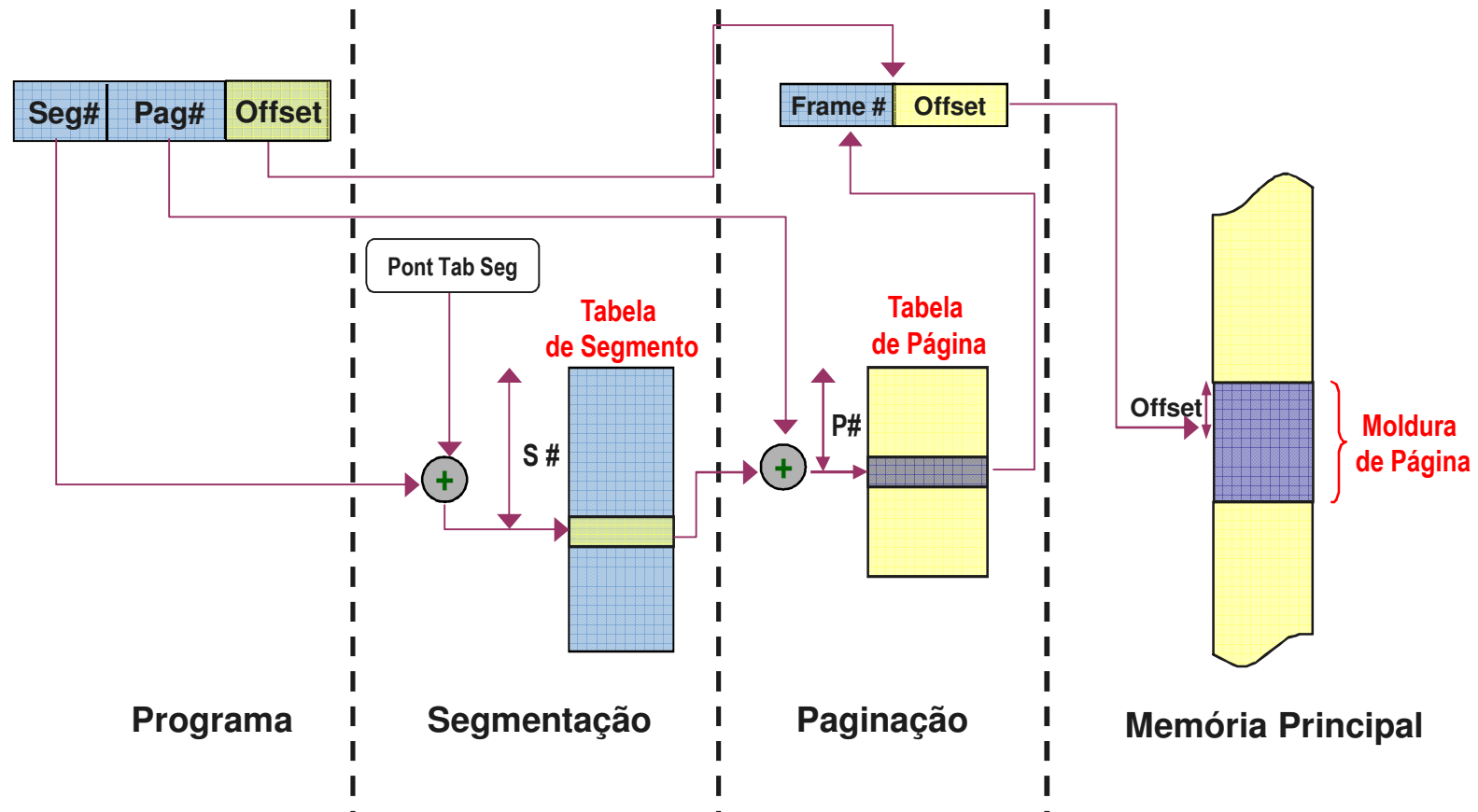
Tabela de Entrada de Página

P	M	Outros bits controle	Número Moldura
---	---	----------------------	----------------



Gerenciamento de Recursos I

Tradução de Endereços





PS - Paginação Simples
SS - Segmentação Simples
PMV - Paginação de Memória Virtual
SMV - Segmentação de memória Virtual

Gerenciamento de Recursos I

Paginação x Segmentação

- MP é particionada em pedaços fixos (frames)
- Programa é quebrado em páginas pelo compilador (ou gerenciamento de memória)
- Segmentos de programas são especificados pelo programador
- Fragmentação interna
- Fragmentação externa
- SO necessita manter tabela de (p)áginas/(s)egmentos para cada processo, mostrando em que moldura o processo se encontra
- SO necessita manter uma lista de (m)olduras/(e)spaços livres na MP
- Processador utiliza o número de (p)áginas/(s)egmentos, offset para calcular o endereço absoluto
- TODAS as (p)áginas/(s)egmentos precisam estar na memória principal para ser executado, a menos que use overlays

PS	SS	PMV	SMV
X		X	
X		X	
	X		X
X		X	
	X		X
P	S	P	S
M	E	M	E
P	S	P	S
P	S		



Gerenciamento de Recursos I

Estratégias do SO

BUSCA

- Por demanda
- Antecipada

ALOCACÃO

RE-ALOCACÃO

- Ótima
- LRU
- FIFO
- Do relógio

CONJUNTO RESIDENTE

- Tamanho fixo
- Tamanho variável
 - ✓ Substituição global
 - ✓ Substituição local

LIMPEZA

- Por demanda
- Antecipada

CONTROLE DO NÚMERO DE PROCESSOS



Determina quando uma página deve ser carregada

- Por demanda
 - **Somente traz as páginas referenciadas**
 - **Existem muitas faltas de página quando o processo começa**
- Por carga antecipada
 - **Traz mais páginas do que o necessário**
 - **É mais eficiente trazer várias páginas contíguas em disco do que cada uma individualmente**
 - **Se torna ineficiente se são trazidas páginas que não serão referenciadas**



Gerenciamento de Recursos I

Estratégia de Alocação

Determina onde será carregada a página ou segmento na memória



- Irrelevante em sistemas paginados
=> A eficiência do hardware é a mesma para qualquer combinação página-moldura
- Em sistemas com segmentação pura deve ser usado um dos algoritmos first-fit, worst-fit, next-fit



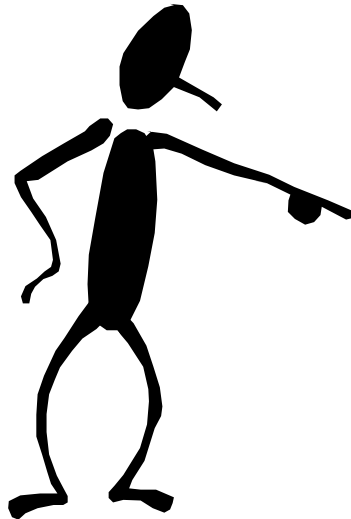
Determina a página a ser removida quando uma nova página está sendo carregada

- A página substituída pode ser:
 - uma das páginas residentes do próprio processo (LOCAL)
 - qualquer página residente
 - definido pela política de gerência do conjunto residente
- Algumas molduras podem ser marcadas como não removíveis (locked ou pinned)
 - código do kernel
 - estruturas críticas do S.O.
 - buffers de E/S



Gerenciamento de Recursos I

Algoritmos



ÓTIMO
LRU
FIFO



**Seleciona a página cuja
próxima referência será a mais
distante**

- ❖ **Resulta no menor número de falta de páginas**
- ❖ **Irrealizável já que não é possível prever o futuro**
- ❖ **Útil para avaliar a eficiência de outras políticas**



**Substitui a página que não é
referenciada há mais tempo
(Least Recently Used)**

- Pelo princípio de localidade, esta deve ser a página com menos probabilidade de ser referenciada no futuro próximo

↑ O desempenho pode ser quase tão bom quanto a política ótima

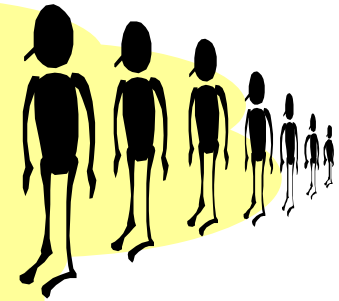
↓ Implementação computacionalmente cara

- Manter o tempo do último acesso para cada moldura
- Manter as molduras ordenadas por acesso



Algoritmo FIFO

**Substitui a página carregada
há mais tempo
(First in, First out)**



✓ As molduras formam um buffer circular



Algoritmo extremamente simples



A página residente há mais tempo na memória não significa que não será mais utilizada

É possível que hajam trechos utilizados constantemente durante a toda execução do programa



Aproximação do algoritmo LRU

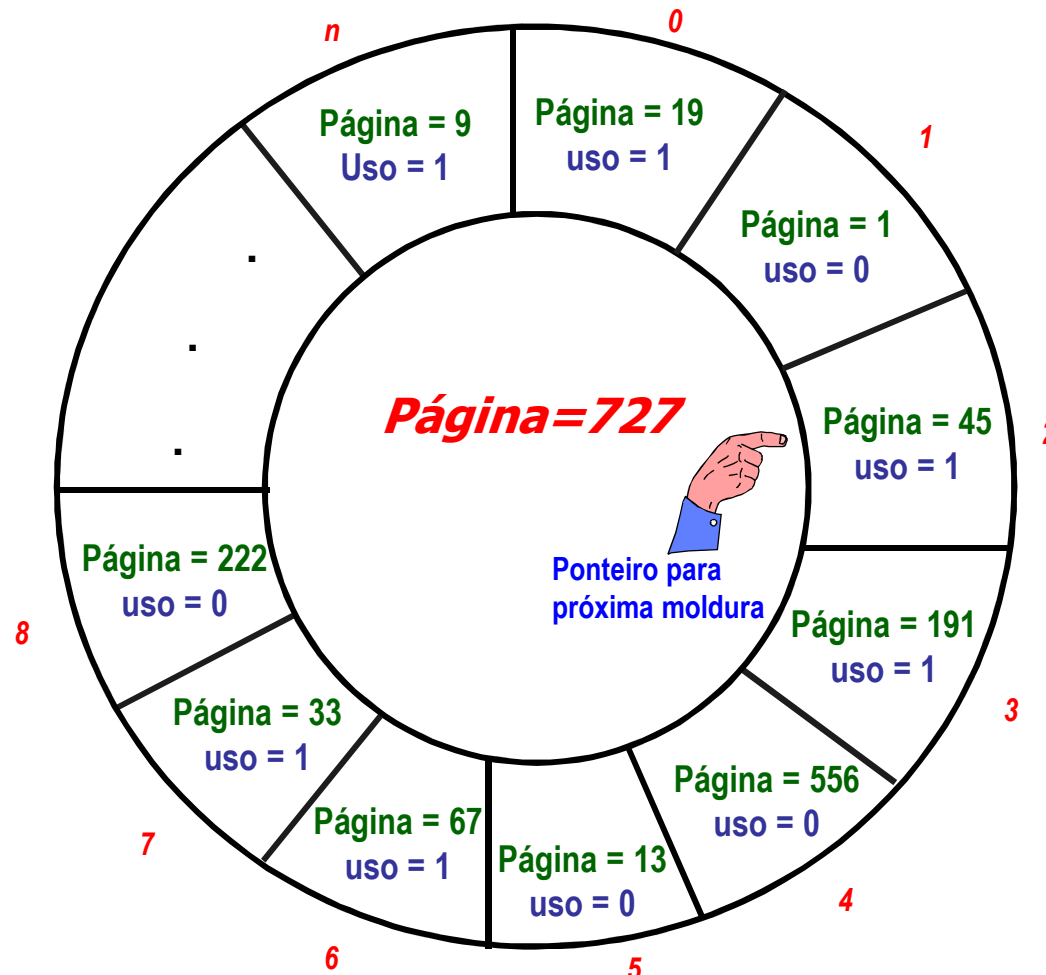


- ↪ Variações conhecidas como NRU (Not Recently Used)
- ↪ Requer um bit adicional na tabela de páginas: o bit de uso (ou referência)
- ↪ Quando a página é carregada, o bit de uso é desligado
- ↪ Quando a página é referenciada, o bit é ligado
- ↪ A primeira página com o bit de uso igual a zero é removida
- ↪ Durante a procura da página a ser substituída, os bits de uso das páginas pesquisadas são desligados



Gerenciamento de Recursos I

Algoritmo do Relógio

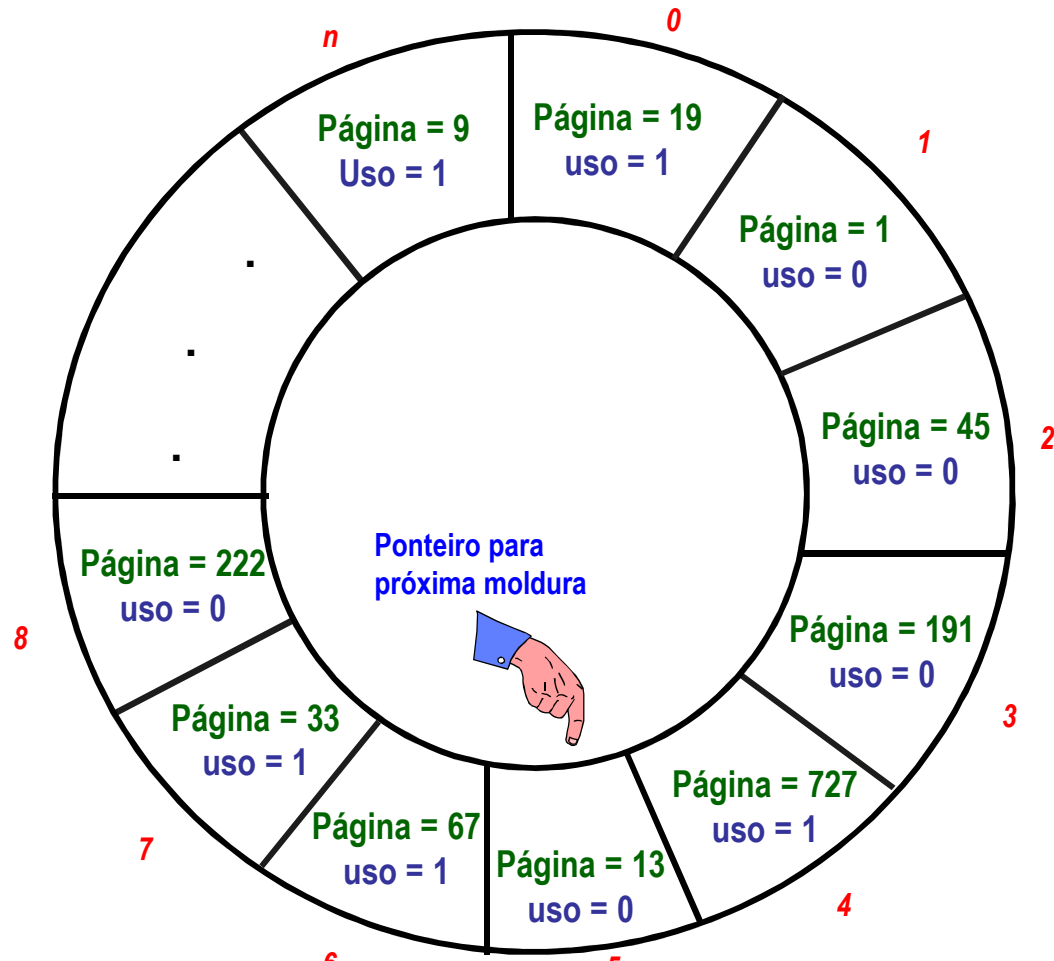


Status do buffer momentos antes de uma substituição de página



Gerenciamento de Recursos I

Algoritmo do Relógio



Status do buffer momentos após a próxima substituição de página



Gerenciamento de Recursos I

Comportamento dos 4 Algoritmos

Seqüência de
Endereçamento de
Página

2 3 2 1 5 2 4 5 3 2 5 2

OTIMO

2	2	2	2	2	2	2	4	4	2	2	2
	3	3	3	3	3	3	3	3	3	3	3
			1	5	5	5	5	5	5	5	5
				F		F			F		

LRU

2	2	2	2	2	2	2	2	3	3	3	3
	3	3	3	5	5	5	5	5	5	5	5
			1	1	1	4	4	4	2	2	2
				F		F		F	F		

FIFO

2	2	2	2	5	5	5	5	3	3	3	3
	3	3	3	3	2	2	2	2	2	5	5
			1	1	1	4	4	4	4	4	2
				F	F	F		F		F	F

RELÓGIO

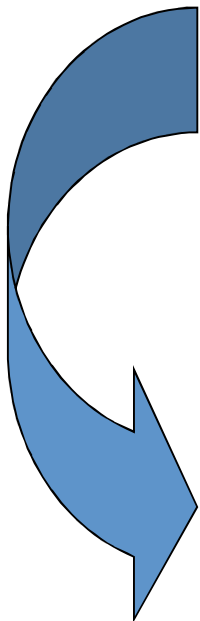
2*	2*	2*	2*	5*	5*	5*	5*	3*	3*	3*	3*
	3*	3*	3*	3	2*	2*	2*	2	2*	2	5*
			1*	1	1	4*	4*	4	4	5*	2*
				F	F	F		F		F	



Gerenciamento de Recursos I

Algoritmo do Relógio

Possível modificação no algoritmo:



- Uso do bit de modificação
- Inicialmente procura por uma página não referenciada e não modificada
- Procura a seguir por uma página não referenciada mas modificada, desligando o bit de uso durante a procura
- Repete em ordem as duas procuras
- Diminui o número de acessos a disco, mas aumenta o tempo de procura



Gerenciamento de Recursos I

Política de Alocação Fixa

- O tamanho do conjunto residente de cada processo é fixo
- Esse número é determinado no tempo de carga do processo, baseado:
 - Tipo do processo
 - Informações do compilador ou programador
- Quando há uma falta de página, uma das páginas do processo é escolhida para ser removida da memória



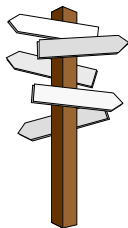
Problema:

O número de molduras alocadas pode ser muito pequeno ou desnecessariamente grande



Política de Alocação Variável

- O número de molduras alocadas a um processo pode variar durante a execução deste
- O tamanho do conjunto residente é dinamicamente ajustado de maneira a evitar uma taxa alta de falta de páginas ou um desperdício de memória
- É exigido um overhead maior para monitorar o comportamento dos processos



A implementação depende do escopo das substituições de páginas:

Local: é escolhida uma página do processo

Global: é escolhida uma página qualquer



Política de Substituição Global

- É o mecanismo mais simples e muito utilizado
- Quando há memória disponível, uma falta de página aumentará o tamanho do conjunto residente do processo em questão
- Quando não há memória disponível, uma falta de página diminuirá o tamanho do conjunto residente de um processo qualquer
- O processo que perdeu a página pode não ser o ótimo
- O mau comportamento de um processo degrada o desempenho dos outros processos



Gerenciamento de Recursos I

Política de Substituição Local

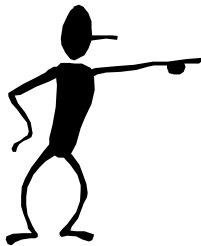
Estratégia:

- Carregar o processo com um número pré-determinado de molduras
- Em caso de falta de página, substituir uma das páginas do próprio processo
- De tempos em tempos, reavaliar o tamanho do conjunto residente, aumentando-o ou diminuindo-o.



Modelo do Conjunto de Trabalho

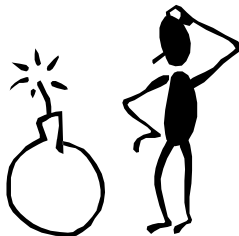
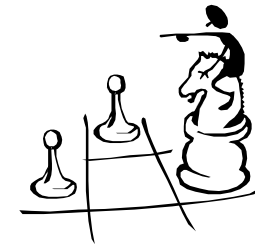
Definição:



- Conjunto de trabalho (working set) de um processo
 - O conjunto de páginas referenciadas num tempo Δ
 - É uma função do tempo e de Δ
 - Usualmente os programas alternam períodos de estabilidade com períodos de mudanças bruscas

Estratégia:

- **Monitorar o conjunto de trabalho e periodicamente remover da memória as páginas que não façam parte do mesmo**



Problemas:

- Nem sempre o passado prediz o futuro
- É impraticável medir o conjunto de trabalho real
- O valor ótimo de Δ é desconhecido



Gerenciamento de Recursos I

Política de Limpeza

Determina quando uma página modificada (suja) é escrita em disco

Por demanda:

- Somente quando é substituída
- Uma falta de página pode implicar em duas E/S em disco

Antecipada:

- São escritas periodicamente e em lotes
- Pode ser necessário escreve-las novamente

Buffer de páginas:

- as páginas na lista de livres modificadas são escritas periodicamente em lotes



Gerenciamento de Recursos I

Controle de Carga

- Determina o número de processos residentes na memória
- Poucos processos podem causar uma baixa utilização do processador (nenhum processo pronto)
- Muitos processos causam trashing no sistema



Solução:

- Suspende processos quando:
 - Seu conjunto de trabalho não cabe na memória
 - O tempo médio entre faltas é menor que o tempo médio para processar uma falta
 - O dispositivo de paginação tem uma taxa de utilização maior que 50%
 - O ponteiro do relógio avança muito rapidamente



- Processo candidato à suspensão
 - Processo com menor prioridade
 - Processo sofrendo a falta de página
 - Último processo carregado
 - Processo com o menor conjunto residente
 - O processo com o maior uso de memória
 - O processo com o maior tempo restante de execução



≡ Sistema de paginação

- ♦ memória virtual para processos
- ♦ buffer para blocos de disco

≡ Estruturas de dados

- ♦ tabela de páginas (por processo)
- ♦ descritor do bloco em disco (por página virtual)
- ♦ tabela de molduras de página
- ♦ tabela de uso dos blocos de swap



▄ Substituição de páginas

- ♦ Lista de molduras livres
- ♦ Kernel mantém um mínimo de páginas livres
- ♦ Algoritmo do relógio modificado (2 ponteiros)
 - Um ponteiro é usado para desligar o bit de uso
 - O outro ponteiro retira as páginas com o bit de uso igual a 0
 - Quanto maior a distância entre os dois ponteiros maior a chance da página ser referenciada
- ♦ Quanto menor a memória disponível, mais rápido os ponteiros circulam pela memória



- Espaço de endereçamento de um processo
 - 2 GBytes para o usuário
 - 2 GBytes para o S.O. (inacessíveis em modo usuário)
 - O espaço para o usuário pode ser de 3 Gbytes

- Estado das páginas virtuais
 - Disponíveis
 - Reservadas (definidas nos VADs)
 - Confirmadas (espaço alocado no arquivo de paginação)
 - Em uso (entradas na tabela de páginas alocadas)



- Política do conjunto residente
 - Alocação variável com substituição local
 - O conjunto residente é chamado de working set
 - O working set varia entre um limite mínimo e um limite máximo
 - Se há memória livre, uma falta de página aumenta o working set
 - Quando há pouca memória livre, uma thread do kernel diminui o working set dos processos
 - Mesmo com pouca memória livre, um grande número de falta de páginas aumenta o working set de um processo



- Política de substituição
 - Arquitetura Intel com um processador
 - Algoritmo do relógio
 - Arquitetura Alpha e arquitetura Intel com múltiplos processadores
 - Algoritmo FIFO
 - A arquitetura Alpha não implementa o bit de uso
 - O algoritmo do relógio fica muito caro em SMPs devido à necessidade de invalidar as TLBs de outros processadores toda vez que o bit de uso é desligado



- Estados de uma página
 - ativa: parte de um working set
 - standby: retirada de um working set e limpa
 - modificada: retirada de um working set e suja
 - livre: disponível para uso
 - zerada: livre e contendo zeros
 - páginas livres tem seu conteúdo desligado quando há poucas páginas zeradas
 - páginas modificadas são gravadas em lotes quando há muitas páginas modificadas ou pouca memória disponível