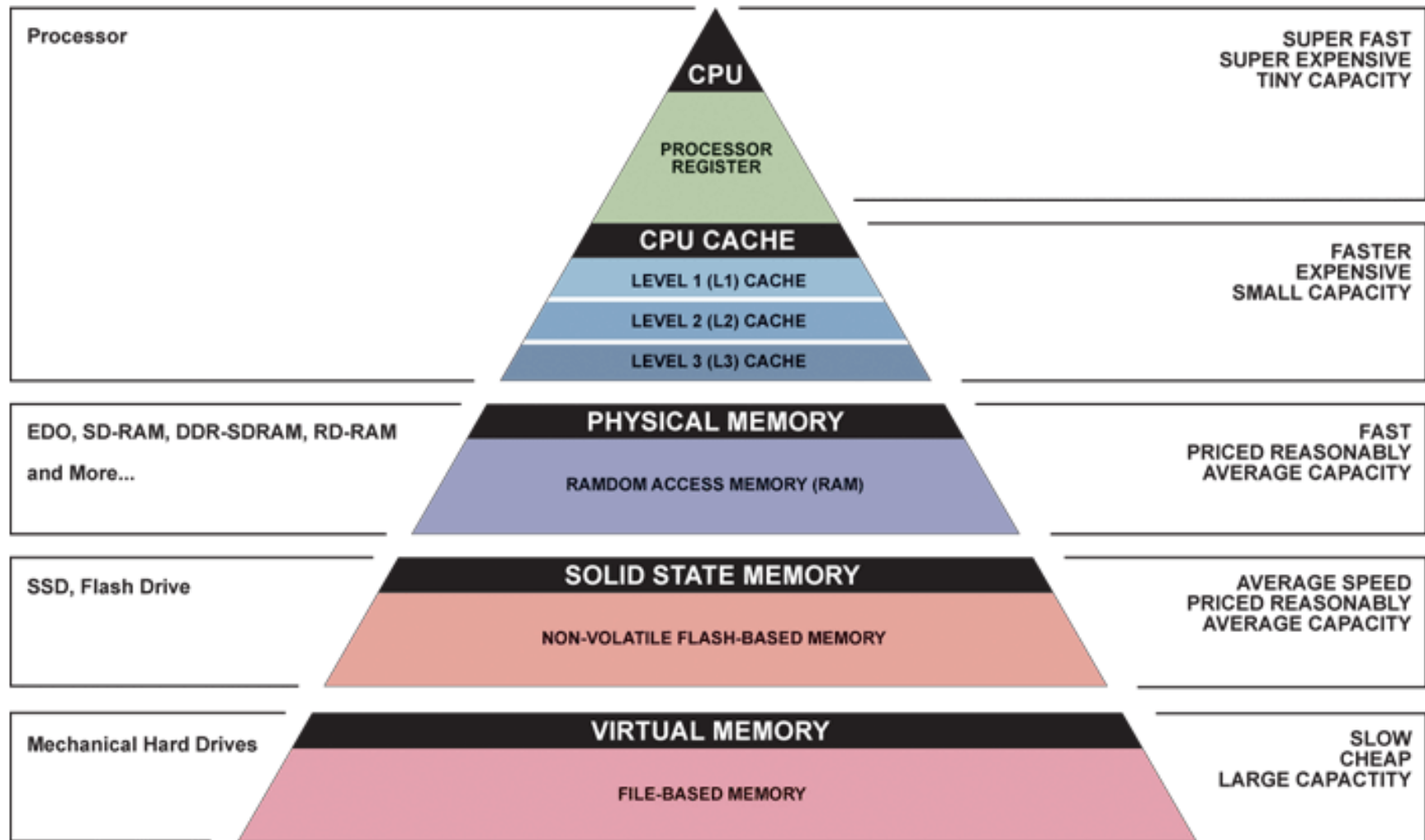


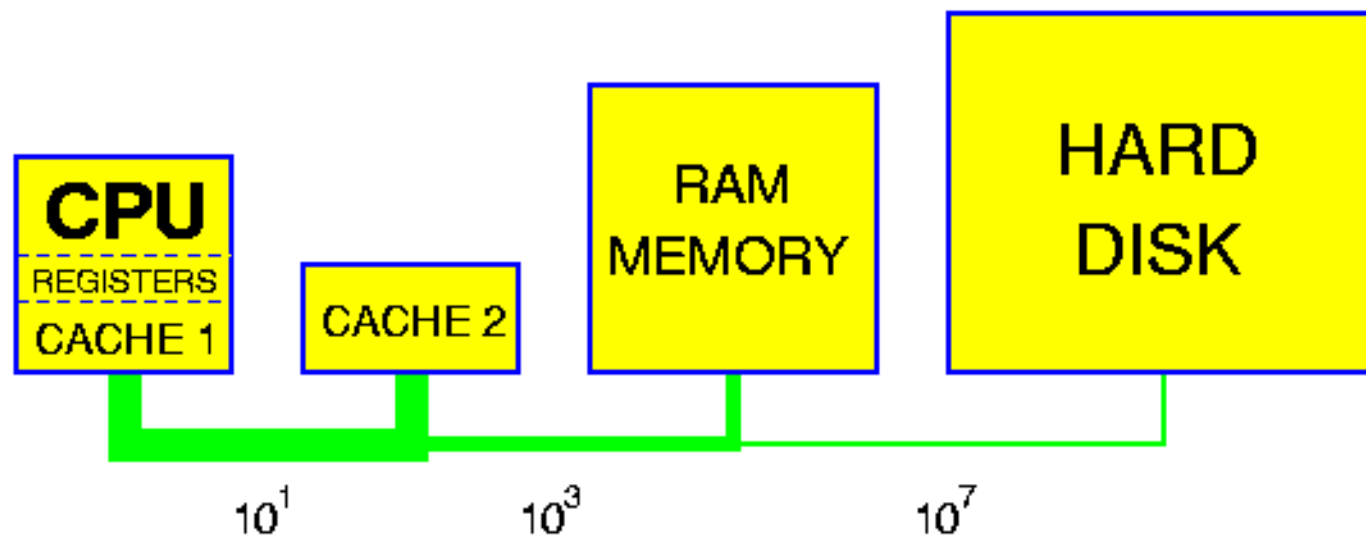
Hierarquia de memória



▲ Simplified Computer Memory Hierarchy
Illustration: Ryan J. Leng

Hierarquia de memória

MEMORY HIERARCHY



Indicated are approximate numbers of clock cycles to access the various elements of the memory hierarchy

Localidade de Referência

- Padrões de acesso a dados observados desde os primeiros sistemas de computação:
 - **Temporal**: se um dado é acessado uma vez, há uma probabilidade grande dele ser acessado novamente num futuro próximo.
 - Ex: instruções dentro do corpo de um loop; árvores...
 - **Espacial**: se um dado é acessado uma vez, há uma probabilidade grande do seu vizinho ser acessado em breve.
 - Ex: percorrendo vetor[0], vetor[1], vetor[2], ...

Pesquisa em Memória Secundária

- Localidade de Referência Temporal:
 - Sistemas de hierarquia de memória
- Localidade de Referência Espacial:
 - Estruturas de paginação
- Projeto de um sistema de memória secundária
 - Dimensionamento de cada nível de memória
 - Benefício x custo
 - Dimensionamento de páginas
 - Pequenas demais: muitos “misses”, poucos “hits”
 - Grande demais: fragmentação: memória ocupada por dados que não serão acessados

Introdução

- **Pesquisa em memória secundária:** arquivos contém mais registros do que a memória interna pode armazenar.
- Custo para acessar um registro é algumas ordens de grandeza maior do que o custo de processamento na memória primária.
- Medida de complexidade: custo de transferir dados entre a memória principal e secundária (minimizar o número de transferências).
- Memórias secundárias: apenas um registro pode ser acessado em um dado momento (acesso seqüencial).
- Memórias primárias: acesso a qualquer registro de um arquivo a um custo uniforme (acesso direto).
- O aspecto sistema de computação é importante.
- As características da arquitetura e do sistema operacional da máquina tornam os métodos de pesquisa dependentes de parâmetros que afetam seus desempenhos.

Pesquisa em Memória Secundária

- O que fazer quando os dados (e.g. vetor a ser ordenado) não cabem na memória principal?
- Como implementar de forma eficiente a interface entre dois níveis de memória, e.g., RAM e disco?
- Solução: VIRTUALIZAÇÃO DE MEMÓRIA

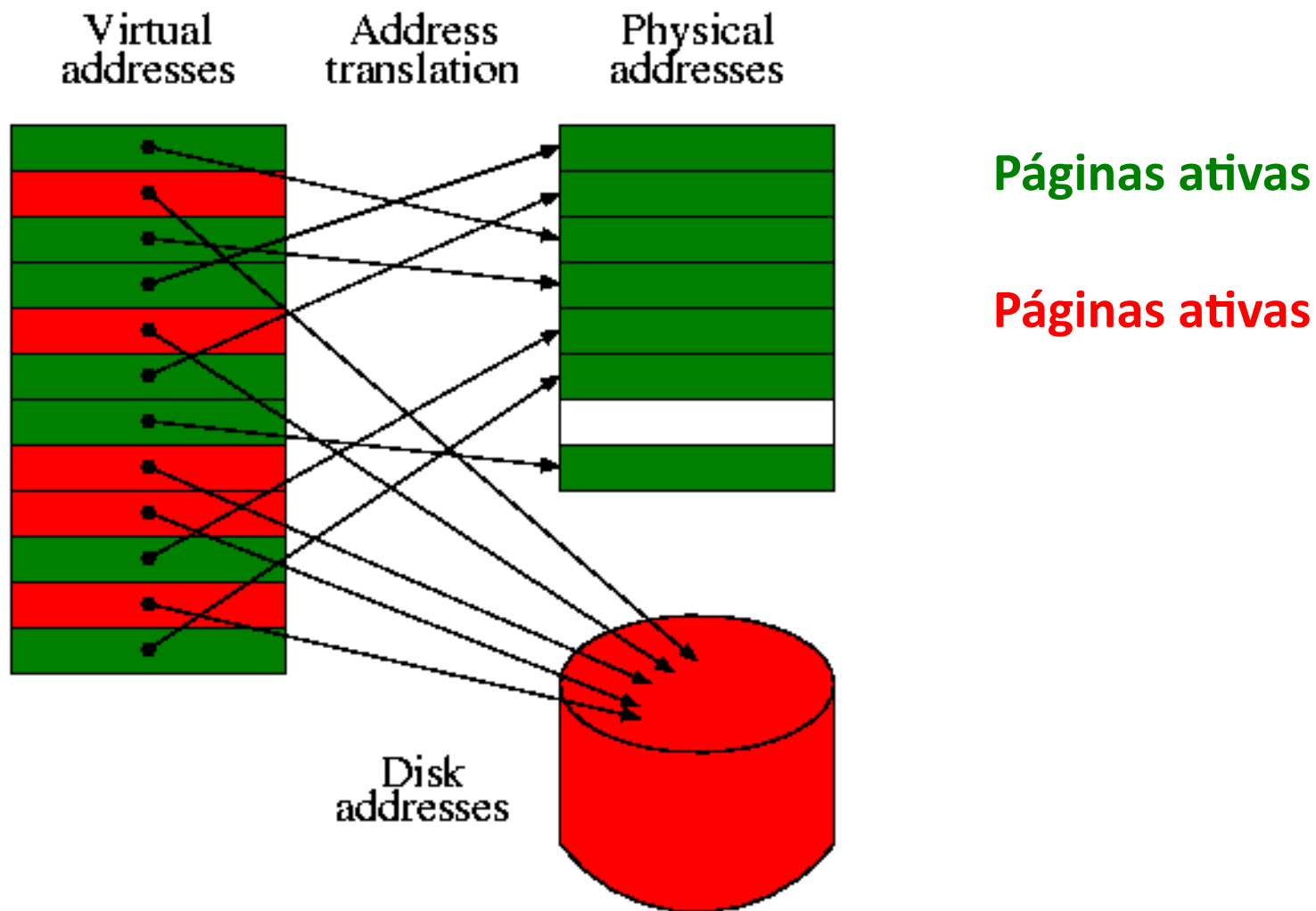
Modelo de computação para Memória Secundária: Memória Virtual

- Memória virtual é um esquema de endereçamento de memória implementado por hardware e software juntos.
 - Controla para o que um endereço de memória aponta
 - Cria a ilusão de um espaço de memória maior e contíguo
 - Tarefas básicas:
 - Mapeamento de endereços entre dois níveis de memória: “imitação de um espaço maior”
 - Transferência eficiente de páginas entre os dois níveis de memória: “qual página remover qdo não há espaço na memória?”

Memória Virtual

- *Espaço de endereçamento virtual:*
 - Espaço (independente) de endereçamento **de um programa**
 - Não possui relação com o espaço físico de memória
 - Permite o uso de dispositivos adicionais, tais como discos rígidos, como se fosse memória
 - Permite o uso de mais “memória” do que o fisicamente instalado no sistema. (Resolve o problema de falta de memória)
 - Permite que cada programa tenha um espaço de endereçamento contíguo que começa com 0x0000

Memória Virtual



Memória Virtual

- *Unidade de Gerenciamento de Memória (paginada) UGM:*
 - Responsável pelo mapeamento entre o espaço de endereçamento virtual e o espaço de endereçamento físico.
 - Quando o sistema de paginação está ativado, TODAS as requisições de memória passam primeiro pela UGM.

Memória Virtual

- *Página virtual*: bloco de memória de tamanho fixo no espaço de endereçamento
 - Descreve o conteúdo e localização de um bloco de memória (de tipicamente 4KB)
- *Moldura de página*: bloco de tamanho fixo (512B-4KB) em memória principal (*frame*, página ativa)
- *Entrada na tabela de páginas*: representa uma página virtual. Ex. formato arquitetura x86:
 - Bit 0 (P): flag de presença em memória
 - Bit 1 (R/W): permissão de leitura/escrita
 - Bit 5 (A): flag de acesso
 - Bit 6 (D): flag de “sujeira”, ou escrita
 - Bits 12-31 (*frame*): endereço da moldura de página
- *Tabela de páginas*: vetor de páginas virtuais. Realiza o mapeamento entre endereços físicos e virtuais

Modelo de Computação para Memória Secundária - Memória Virtual

- Normalmente implementado como uma função do sistema operacional.
- Modelo de armazenamento em dois níveis, devido à necessidade de grandes quantidades de memória e o alto custo da memória principal.
- Uso de uma pequena quantidade de memória principal e uma grande quantidade de memória secundária.
- Programador pode endereçar grandes quantidades de dados, deixando para o sistema a responsabilidade de transferir o dado da memória secundária para a principal.
- Boa estratégia para algoritmos com pequena localidade de referência.
- Organização do fluxo entre a memória principal e secundária é extremamente importante.

Memória Virtual

- Organização de fluxo \rightarrow transformar o endereço usado pelo programador na localização física de memória correspondente.
- *Espaço de Endereçamento* \rightarrow endereços usados pelo programador.
- *Espaço de Memória* \rightarrow localizações de memória no computador.
- O espaço de endereçamento N e o espaço de memória M podem ser vistos como um mapeamento de endereços do tipo: $f : N \rightarrow M$.
- O mapeamento permite ao programador usar um espaço de endereçamento que pode ser maior que o espaço de memória primária disponível.

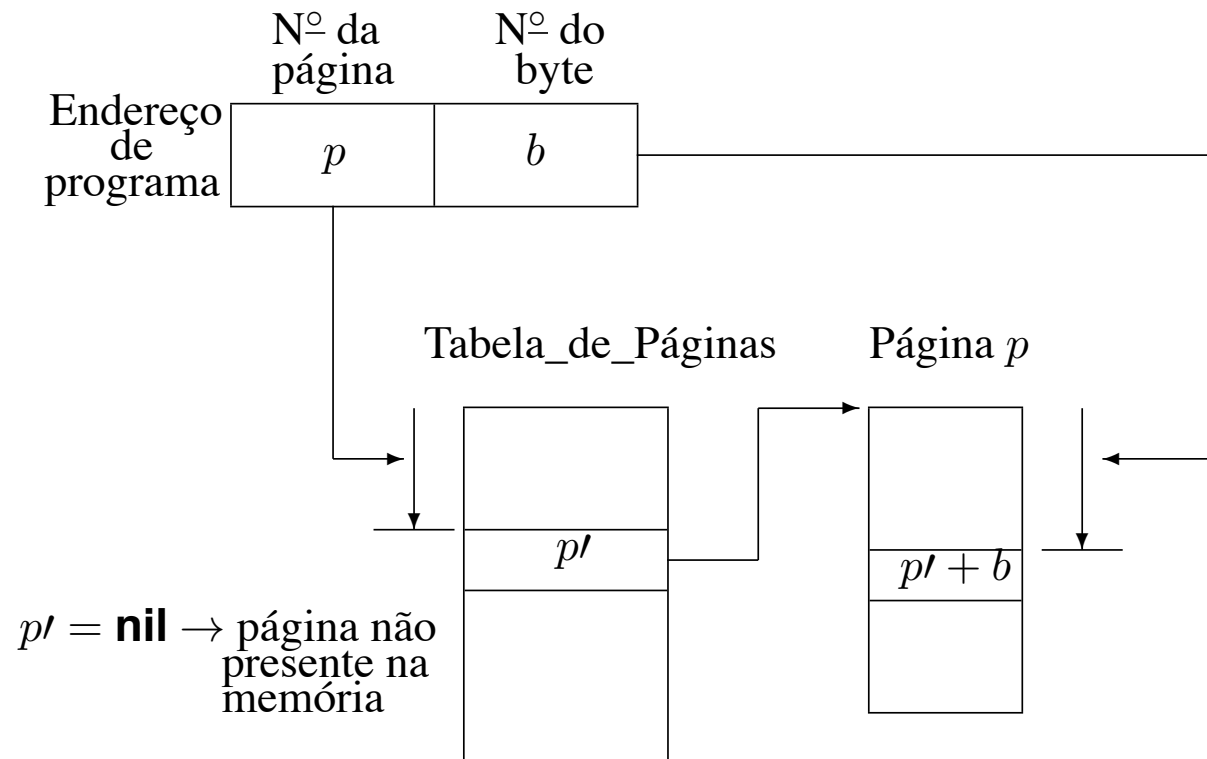
Memória Virtual: Sistema de Paginação

- O espaço de endereçamento é dividido em páginas de tamanho igual, em geral, múltiplos de 512 Kbytes.
- A memória principal é dividida em molduras de páginas de tamanho igual.
- As molduras de páginas contêm algumas páginas ativas enquanto o restante das páginas estão residentes em memória secundária (páginas inativas).
- O mecanismo possui duas funções:
 1. Mapeamento de endereços → determinar qual página um programa está endereçando, encontrar a moldura, se existir, que contenha a página.
 2. Transferência de páginas → transferir páginas da memória secundária para a memória primária e transferí-las de volta para a memória secundária quando não estão mais sendo utilizadas.

Memória Virtual: Sistema de Paginação

- Endereçamento da página → uma parte dos bits é interpretada como um número de página e a outra parte como o número do byte dentro da página (*offset*).
- Mapeamento de endereços → realizado através de uma Tabela de Páginas.
 - a p -ésima entrada contém a localização p' da Moldura de Página contendo a página número p desde que esteja na memória principal.
- O mapeamento de endereços é: $f(e) = f(p, b) = p' + b$, onde e é o endereço do programa, p é o número da página e b o número do byte.

Memória Virtual: Mapeamento de Endereços



Mapeamento de endereços

- Exemplo:
 - Espaço de endereçamento: 24 bits e
 - Tamanho de página: 512 bytes (2^9):
 - Tamanho da memória virtual: 2^{24} bytes
 - 9 bits são usados para representar o número do *byte* dentro da página
 - 15 bits são usados para representar o número da página
 - Mapeamento: $f(e) = f(p, b) = p' + b$, onde:
 - e = endereço de programa,
 - p = entrada na tabela de páginas,
 - p' = localização da moldura de página,
 - b = número do byte

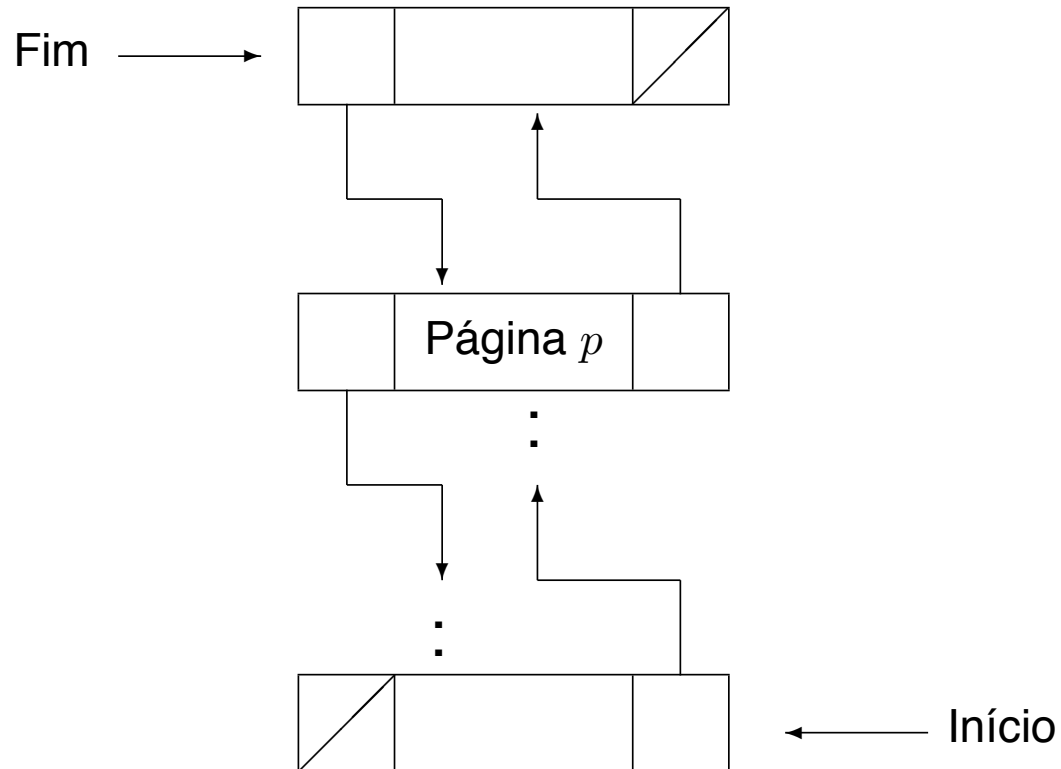
Memória Virtual: Reposição de Páginas

- Se não houver uma moldura de página vazia → uma página deverá ser removida da memória principal.
- Ideal → remover a página que não será referenciada pelo período de tempo mais longo no futuro.
 - tentamos inferir o futuro a partir do comportamento passado.

Memória Virtual: Políticas de Reposição de Páginas

- **Menos Recentemente Utilizada (LRU):**
 - um dos algoritmos mais utilizados,
 - remove a página menos recentemente utilizada,
 - parte do princípio que o comportamento futuro deve seguir o passado recente.
- **Menos Frequentemente Utilizada (LFU):**
 - remove a página menos frequentemente utilizada,
 - inconveniente: uma página recentemente trazida da memória secundária tem um baixo número de acessos e pode ser removida.
- **Ordem de Chegada (FIFO):**
 - remove a página que está residente há mais tempo,
 - algoritmo mais simples e barato de manter,
 - desvantagem: ignora o fato de que a página mais antiga pode ser a mais referenciada.

Memória Virtual: Política LRU



- Toda vez que uma página é utilizada ela é removida para o fim da fila.
- A página que está no início da fila é a página LRU.
- Quando uma nova página é trazida da memória secundária ela deve ser colocada na moldura que contém a página LRU.

Memória Virtual: Estrutura de Dados

```
#define TAMANHODAPAGINA  512
#define ITENSPORPAGINA    64  /* TamanhodaPagina / TamanhodoItem */

typedef struct TipoRegistro {
    TipoChave Chave;
    /* outros componentes */
} TipoRegistro;

typedef struct TipoEndereco {
    long p;
    char b;
} TipoEndereco;

typedef struct TipoItem {
    TipoRegistro Reg;
    TipoEndereco Esq, Dir;
} TipoItem;

typedef TipoItem TipoPagina[ItensPorPagina];
```

Memória Virtual

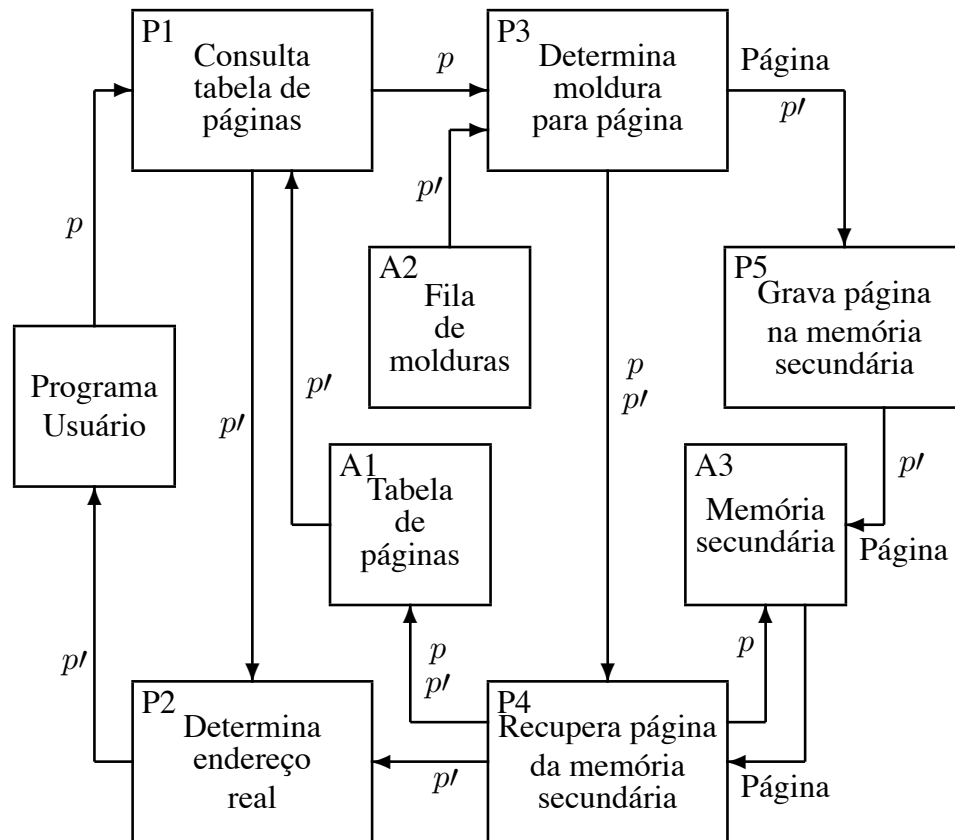
- Em casos em que precisamos manipular mais de um arquivo ao mesmo tempo:
 - A tabela de páginas para cada arquivo pode ser declarada separadamente.
 - A fila de molduras é única → cada moldura deve ter indicado o arquivo a que se refere aquela página.

```
typedef struct TipoPagina {  
    char tipo; /* armazena o código do tipo:0,1,2 */  
    union {  
        TipoPaginaA Pa;  
        TipoPaginaB Pb;  
        TipoPaginaC Pc;  
    }P;  
} TipoPagina;
```

Memória Virtual

- Procedimentos para comunicação com o sistema de paginação:
 - ObtemRegistro → torna disponível um registro.
 - EscreveRegistro → permite criar ou alterar o conteúdo de um registro.
 - DescarregaPaginas → varre a fila de molduras para atualizar na memória secundária todas as páginas que tenham sido modificadas.

Memória Virtual - Transformação do Endereço Virtual para Real



- Quadrados → resultados de processos ou arquivos.
- Retângulos → processos transformadores de informação.

Memória Virtual: conclusão

- Cria a ilusão de um espaço de memória maior e contíguo
- Tarefas básicas:
 - Mapeamento de endereços entre dois níveis de memória: “imitação de um espaço maior”
 - Transferência eficiente de páginas entre os dois níveis de memória: “qual página remover qdo não há espaço na memória?”
- Funciona bem para algoritmos que possuem uma localidade de referência espacial pequena, já que esta propriedade diminui o número de transferências de páginas entre a memória principal e a memória secundária.
- Tema do TP6: implementar um sistema de memória virtual usando a caracterização dos padrões de acesso a dados do TP5.