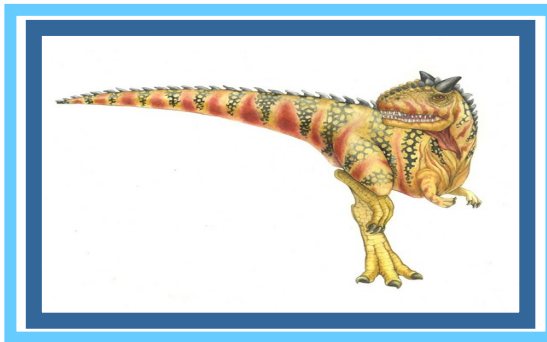


# Paginação x Página Invertida

---





# Paginação x Página Invertida

Cada processo tem uma tabela de páginas.  
*n* processos, *n* tabelas de páginas, cada tabela com  $2^{m-n}$  células,  $p \in 0, \dots, 2^{m-n} - 1$   
Com 64 bits, com páginas de tamanho 4Mb =  $2^{22}$  teríamos  $2^{64}/2^{22} = 2^{42}$  páginas, algo em torno de 4,3 trilhões de páginas para cada processo

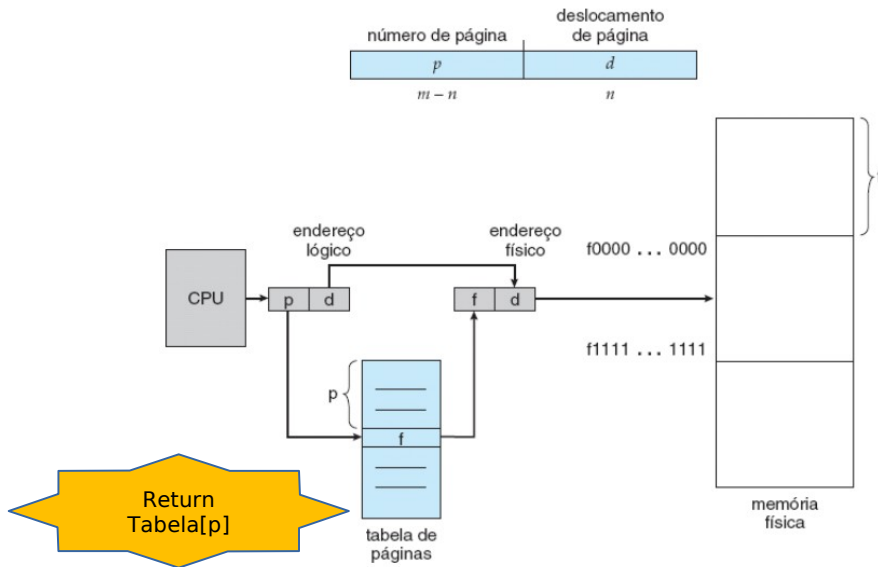


Figura 8.10 Hardware de paginação.

Aqui teremos uma **única tabela de páginas** com todos os processos abertos.  
Esta técnica é chamado de invertida pois a indexação é feita em relação ao número do frame em vez da número da página virtual.  
*n* processos, uma única tabela de páginas  
A tabela de páginas invertida terá informações só das páginas que estão alocadas pelos processos

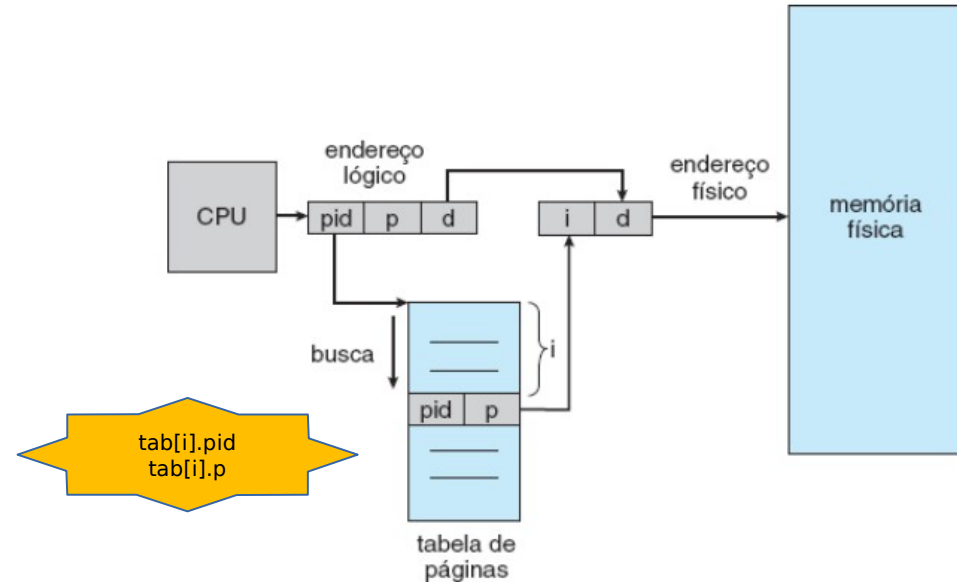


Figura 8.20 Tabela de páginas invertida.



# Paginação x Página Invertida

/\*\*\*\*\*\*

\* **UMA IDÉIA PARA PAGINAÇÃO** \*

\*\*\*\*\*/

```
typedef struct {
    long int d; //0 a 4095, 4 MB
    long int tabPagina[NUM_PAGINAS]; //≈ 4,3 trilhões para 64 bits
} Tprocesso;
```

```
Tprocesso processo[NUM_PROCESSOS];
//O SO lê em Tprocesso[pid].tabPagina[p] o número f do frame, calculando
//o endereço físico por BASE_INICIAL_DOS_FRAMES+4096*f + d
```

/\*\*\*\*\*\*

\* **UMA IDÉIA PARA PAGINAÇÃO INVERTIDA** \*

\*\*\*\*\*/

```
typedef struct {
    long int pid;
    long int pagina;
} Tframe;
```

```
Tframe tabPaginaInvertida[NUMERO_DE_QUADROS]; //Para 16 Gb, ≈ 4096 quadros de 4MB ou 4096 páginas
//O SO percorre a tabPaginaInvertida buscando encontrar pid e p.
//Se encontrar no índice i calcula o endereço físico através de BASE_INICIAL_DOS-FRAMES+4096*i+d
```

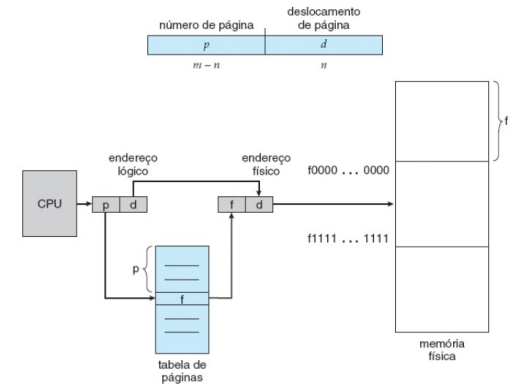


Figura 8.10 Hardware de paginação.

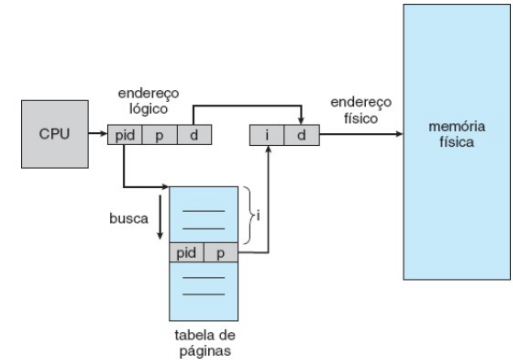


Figura 8.20 Tabela de páginas invertida.



# Paginação x Página Invertida

Uma solução para encontrar mais rapidamente o endereço físico é usar uma tabela Hash. Assim, para cada processo pid e página p teremos um local único para alocar fisicamente na memória.

Se ocorrerem conflitos de endereço vamos verificando os próximos índices.

Caso não encontremos, a página não está na memória.

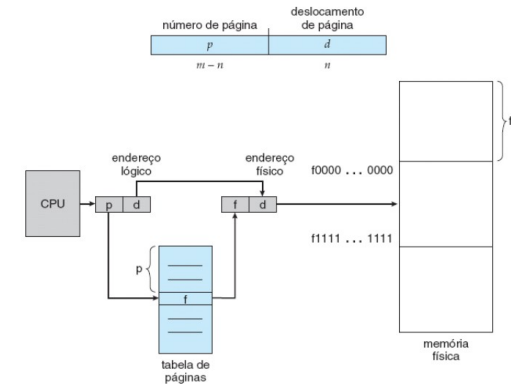


Figura 8.10 Hardware de paginação.

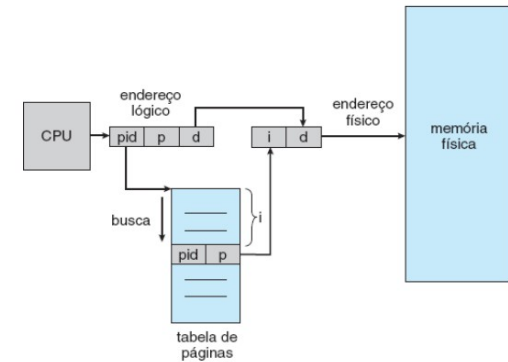


Figura 8.20 Tabela de páginas invertida.