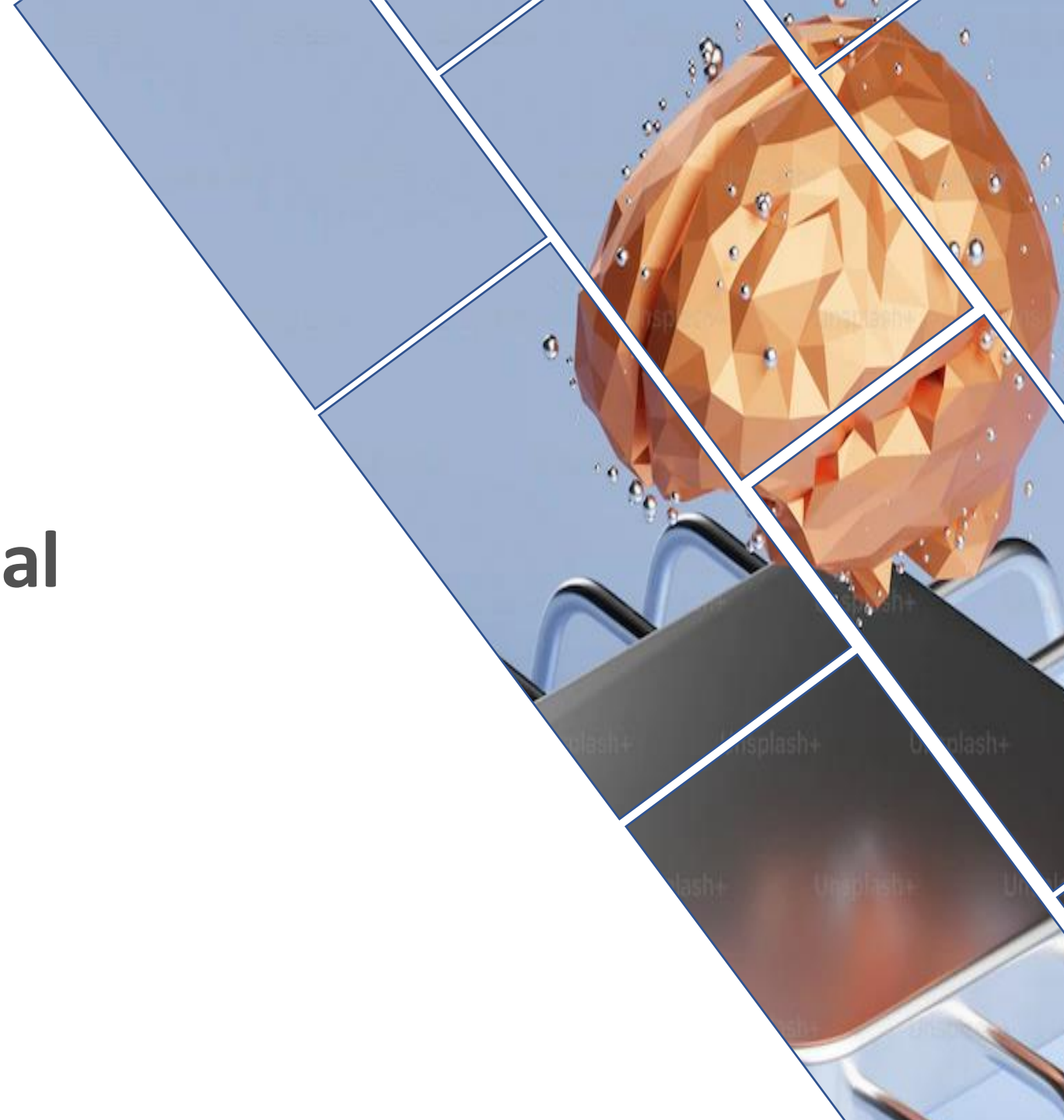
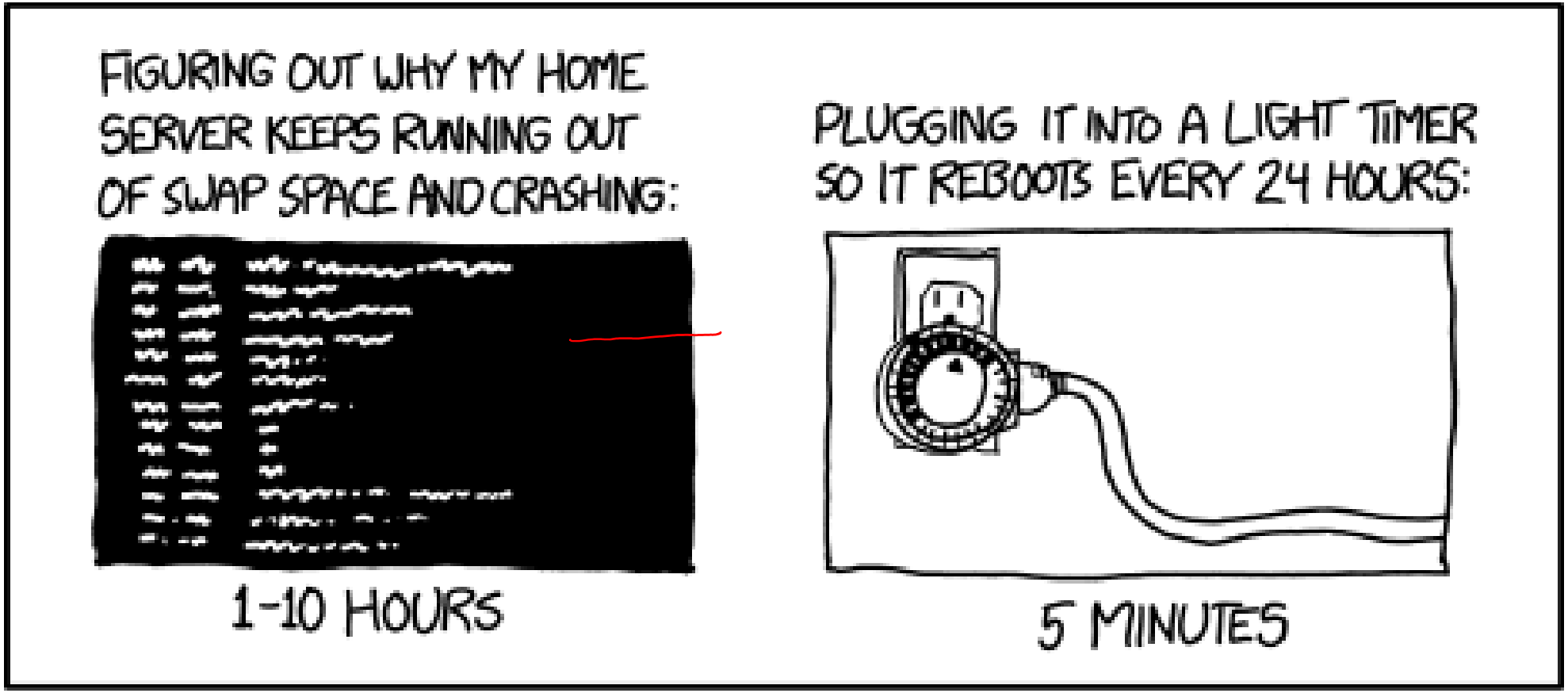
Performance em Sistemas Ciberfísicos

Aula 7 – Memória Virtual



Frank Coelho de Alcantara – 2023 -1

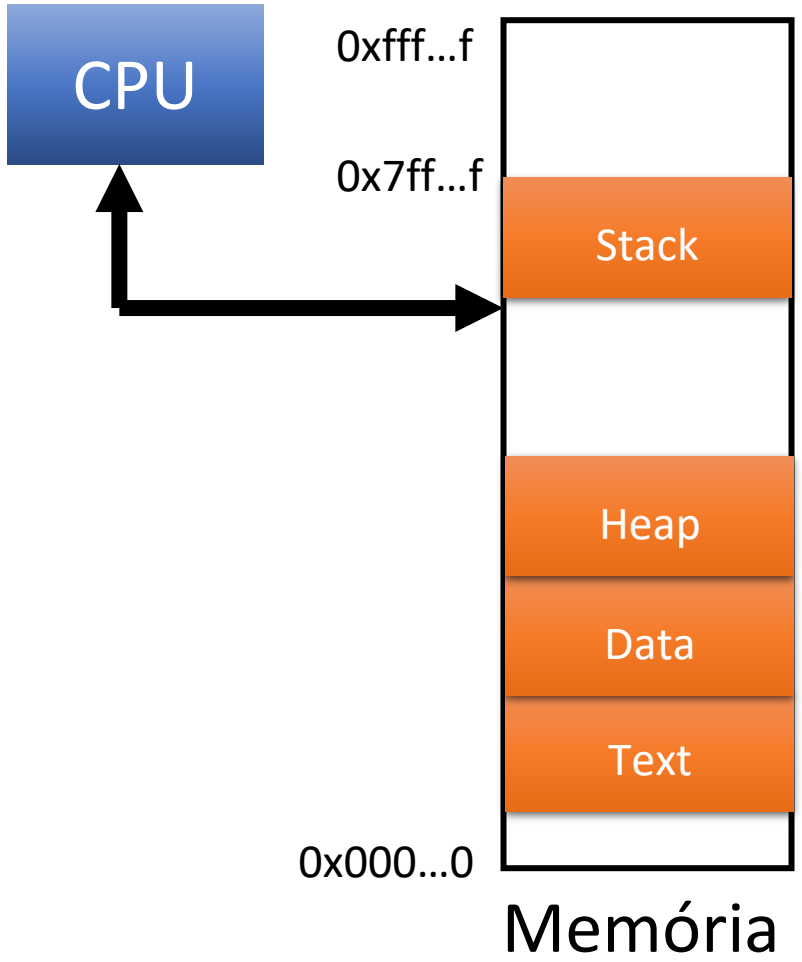
Um Problema de limites



https://xkcd.com/1495/

Sistemas Mono Processados

Vamos chamar de sistemas mono processados, para esta disciplina, todo e qualquer sistema que possua um núcleo de processamento e que este núcleo tenha acesso a toda a memória física.



Olhando o Quadro Geral

Quantos programas a sua máquina está rodando ao mesmo tempo?

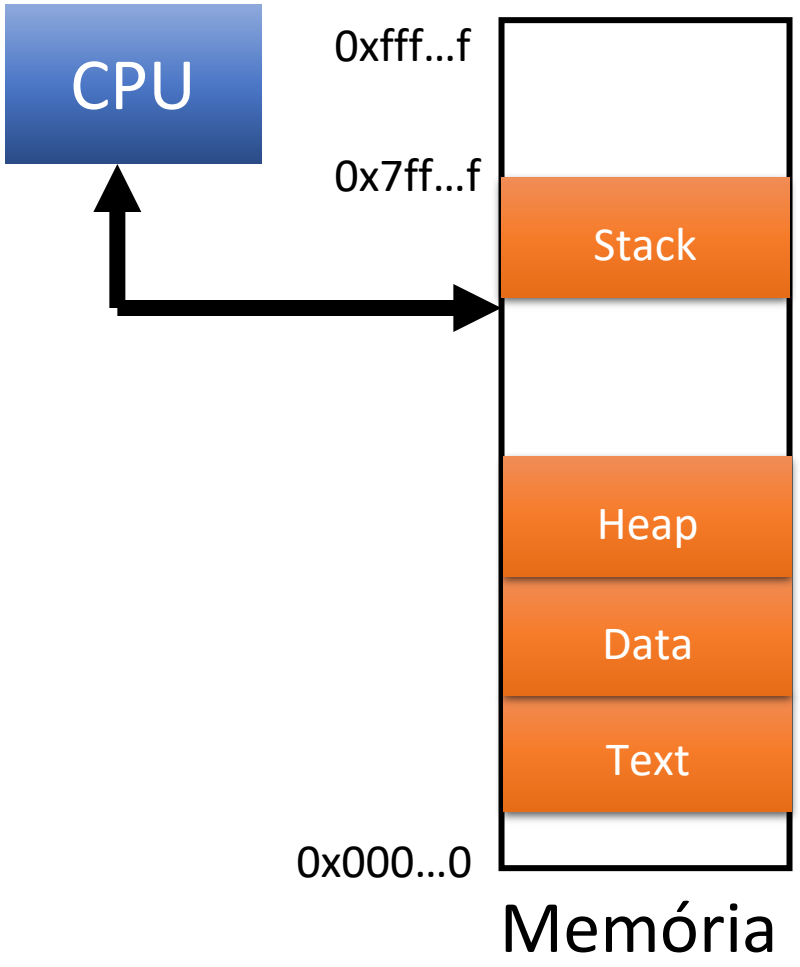
Quadro Geral – Multipolos Processos

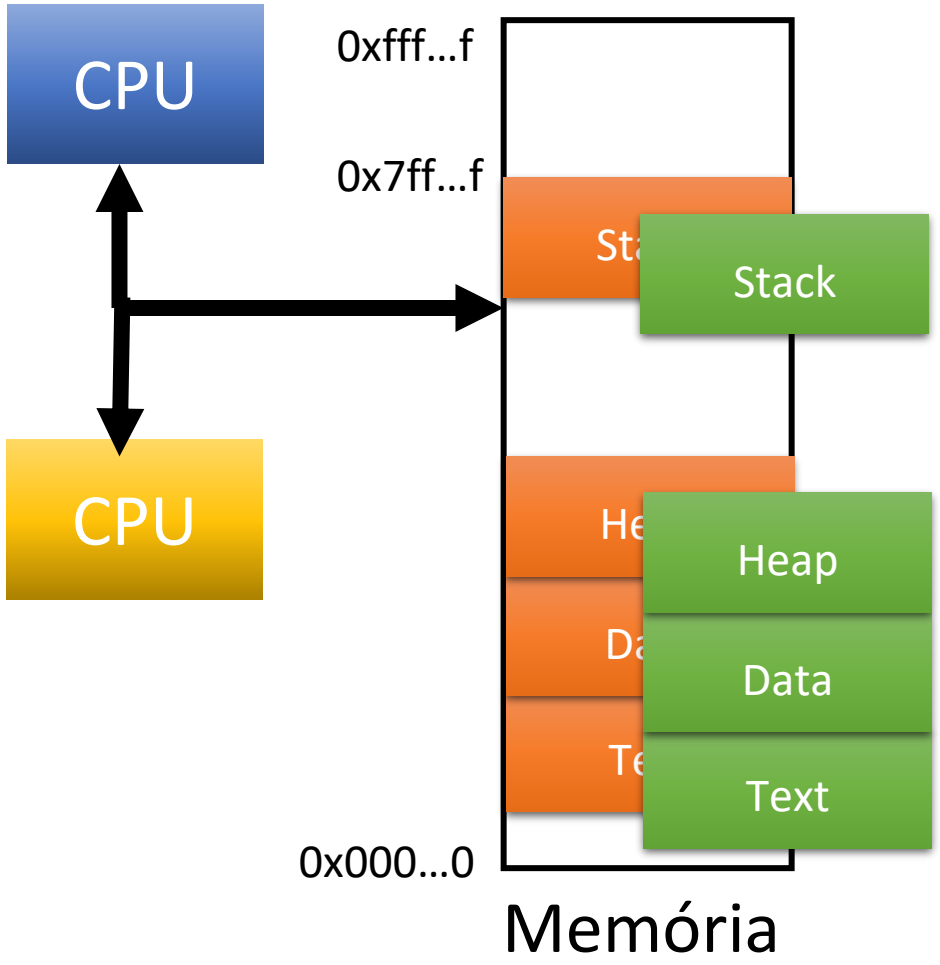
Como rodar múltiplos processos em uma máquina?

Time Multiplexing: o tempo de execução é dividido para cada processo. Isso é muito rápido!

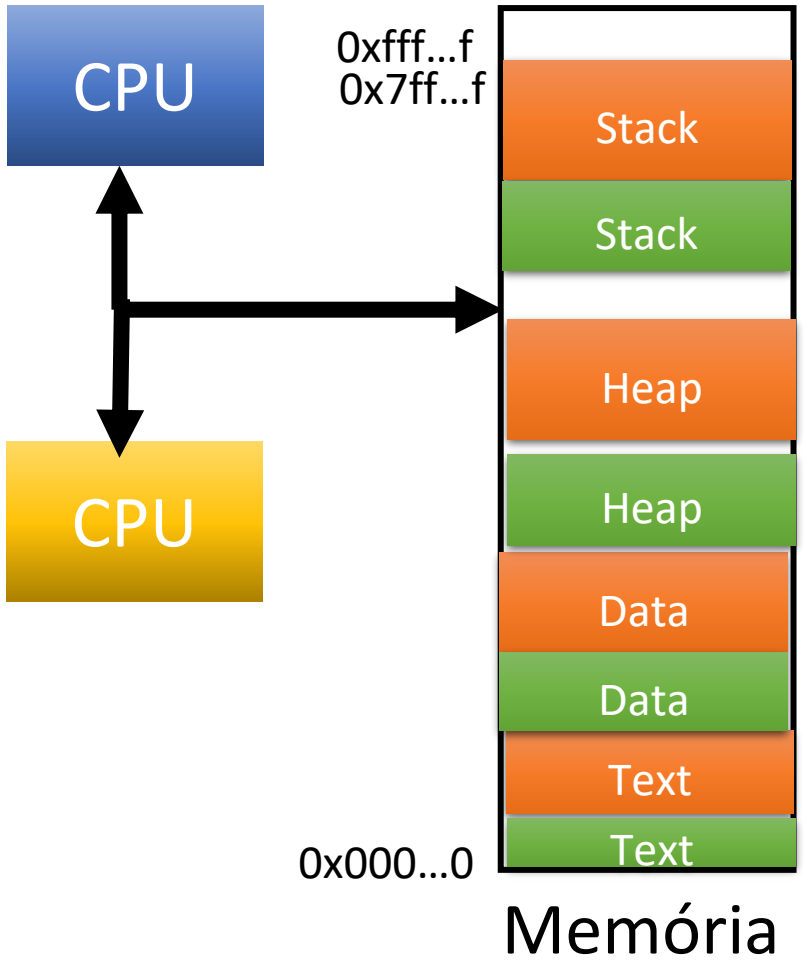
- Um processador: *multi-tasking;*
- Muitos Núcleos: *multi-core*;
- Muitos processadores: *multi-processors*;

O Processador e a Memória



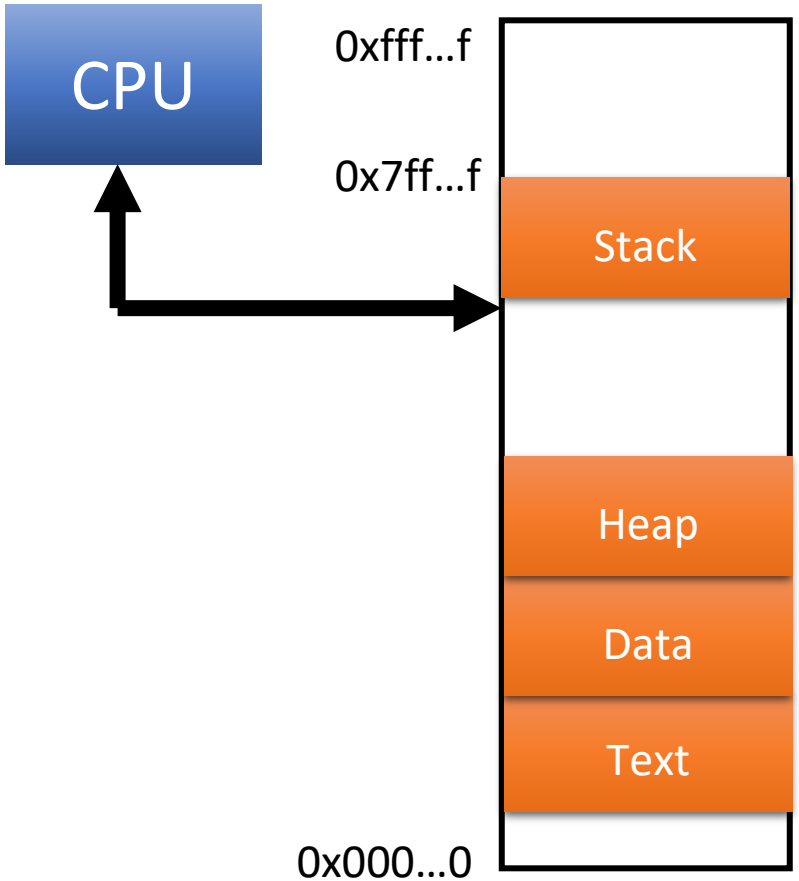


O Processador e a Memória



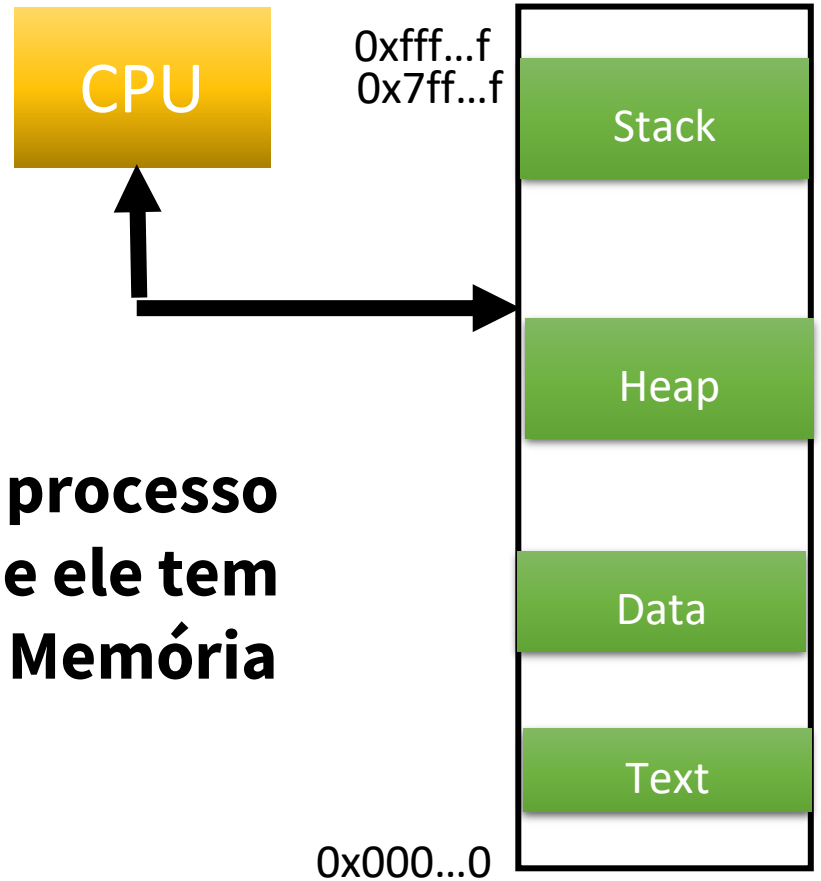
- E se não couber no espaço disponível?
- E se os dados e instruções não ficarem em endereços sequenciais?
- Ainda tem risco de sobreposição de endereços!

O Processador e a Memória



Memória Virtual

Dando a cada processo a Ilusão de que ele tem acesso a toda Memória Física



Memória Virtual

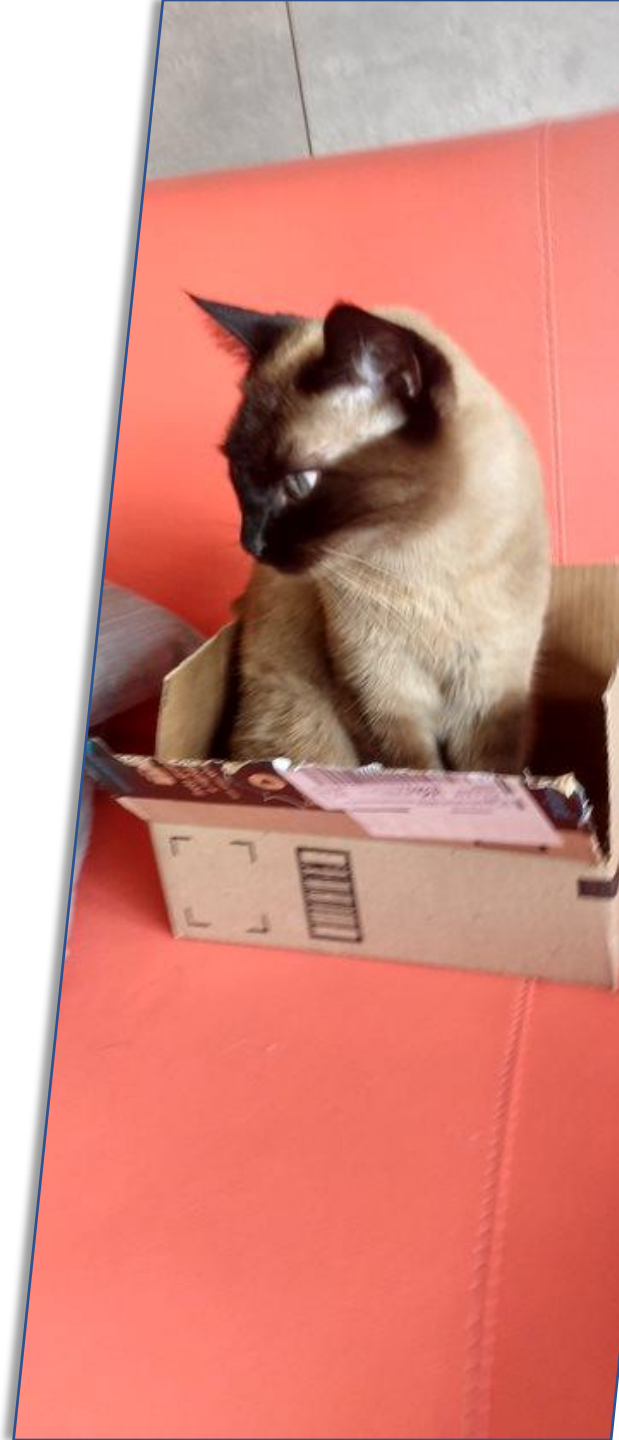
Para Começar a Fazer Sentido

- **Heap:** inicia-se em um endereço relativamente baixo após os segmentos de dados e BSS do programa. Ele cresce para cima (em direção a endereços mais altos) conforme a memória é alocada dinamicamente.
- O endereço inicial exato pode variar entre sistemas operacionais e também pode mudar durante a execução devido à randomização do layout do espaço de endereçãmento (ASLR).

Para Começar a Fazer Sentido

- **Stack:** está localizado nos endereços mais altos do espaço de endereçamento virtual e cresce para baixo (em direção a endereços mais baixos).
- Cada thread possui seu próprio stack, e o endereço inicial exato pode variar devido ao ASLR. Nos sistemas Linux x86-64, o tamanho padrão da pilha é de 8 MB, nos sistemas Windows o tamanho padrão do stack é de 1 MB. Nos dois casos, podemos alterar este tamanho por *thread* ou processo.

Memória Virtual



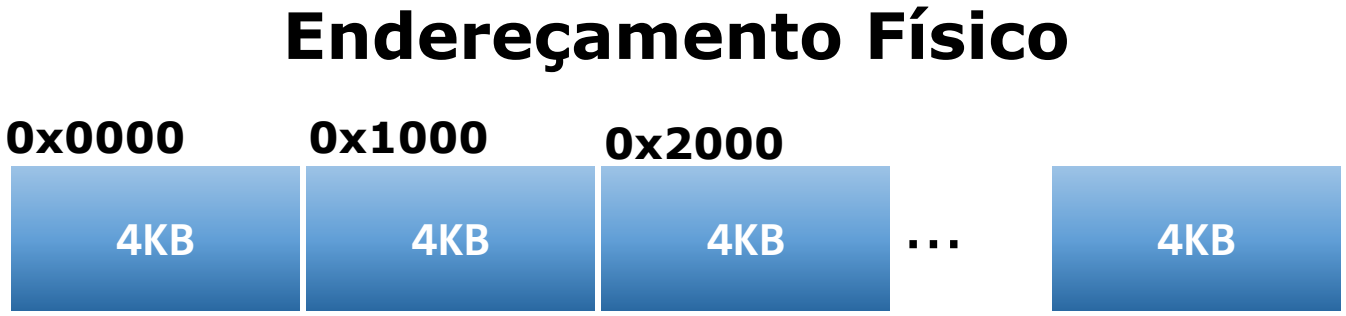
Memória Virtual – Qual o Propósito?

- Dar a cada processo a ilusão de que está rodando com acesso total a memória disponível no sistema.
    - Cada processo terá um *espaço de endereçamento virtual* único e independente de todos os outros processos;
- Preservar os recursos físicos da máquina até que eles sejam realmente necessários.
    - Muitos programas não precisam de todo código e dados na memória o tempo todo;
- Manter uma abstração para unificar todos os dispositivos de memória físicos em um conceito:

Memory Pages: as páginas da memória

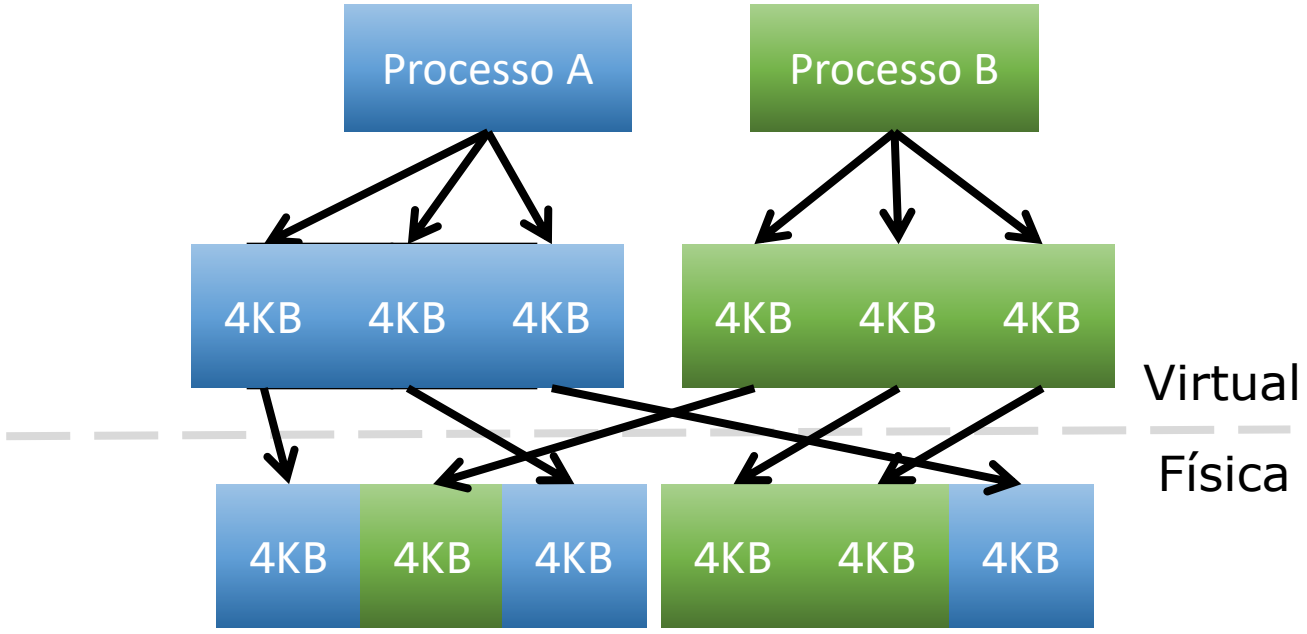
Toda a memória física é dividida em páginas. O tamanho da página depende da arquitetura. Ex. nas arquiteturas X86-64 e ARM as páginas são de 4KBytes (0x1000).

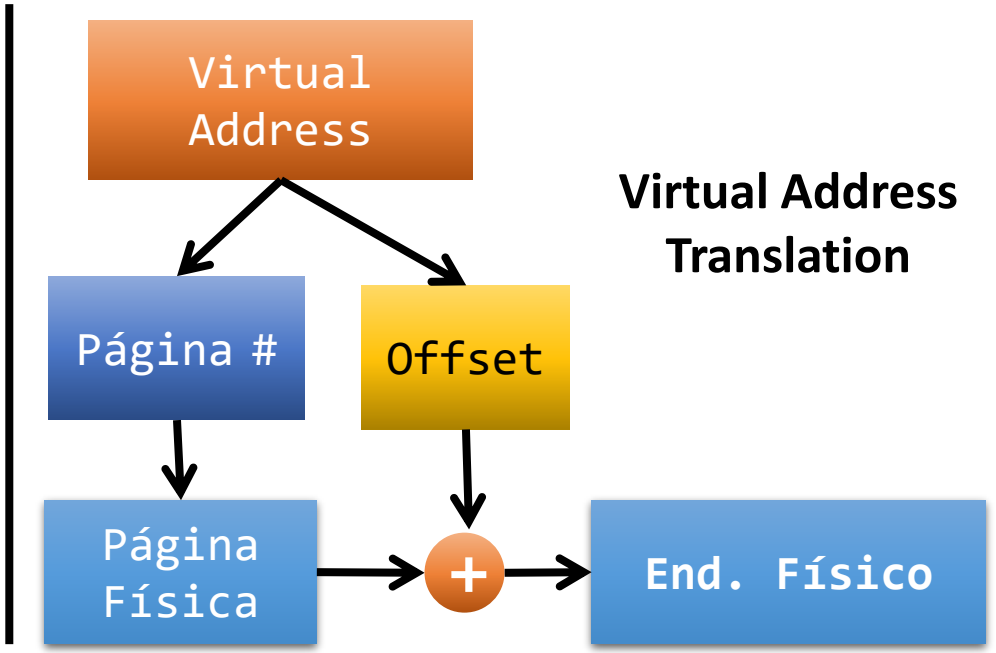
Se o total de memória é $X\ Bytes$ então o número de páginas é $X/4k\ Bytes$.



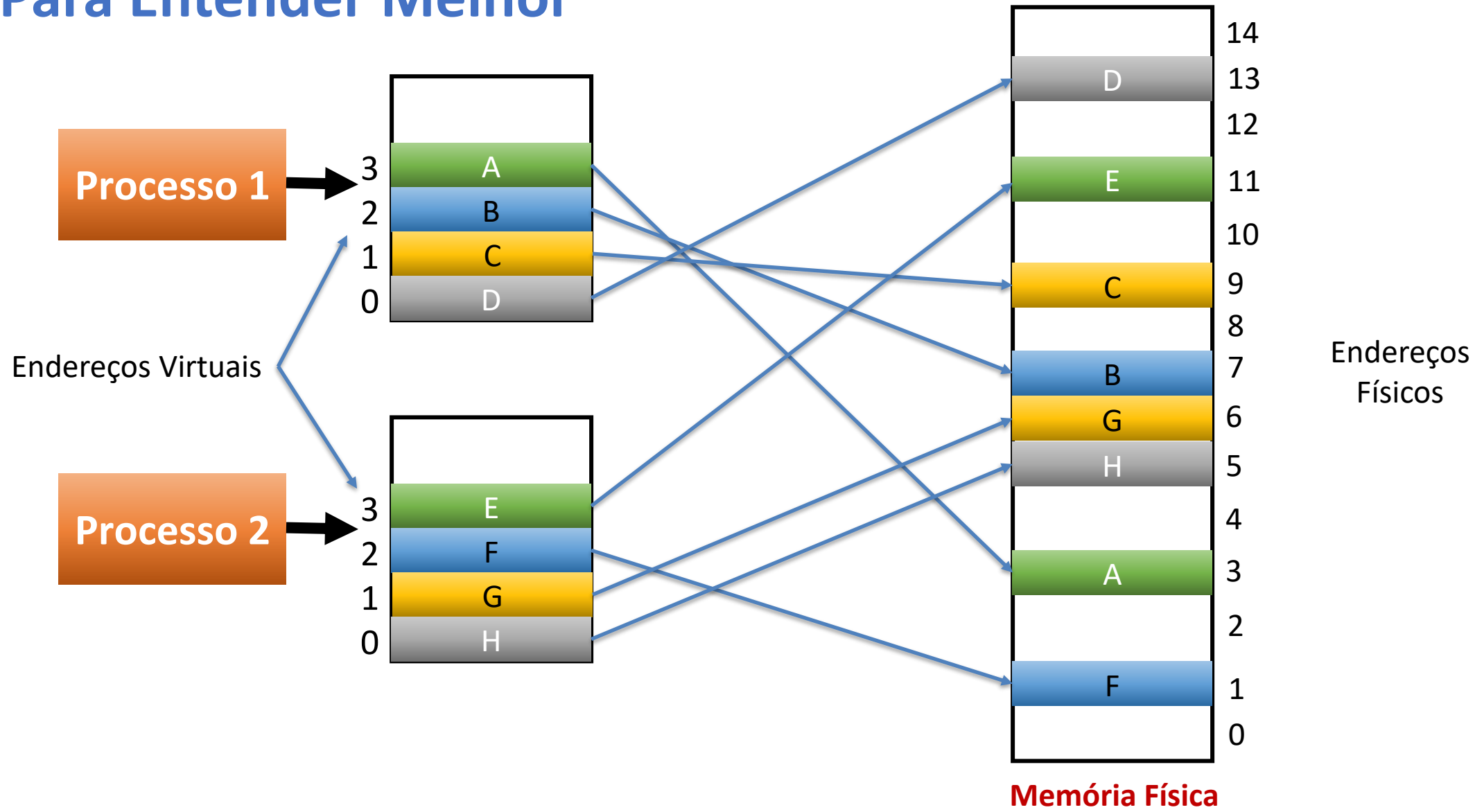
Endereçamento Virtual

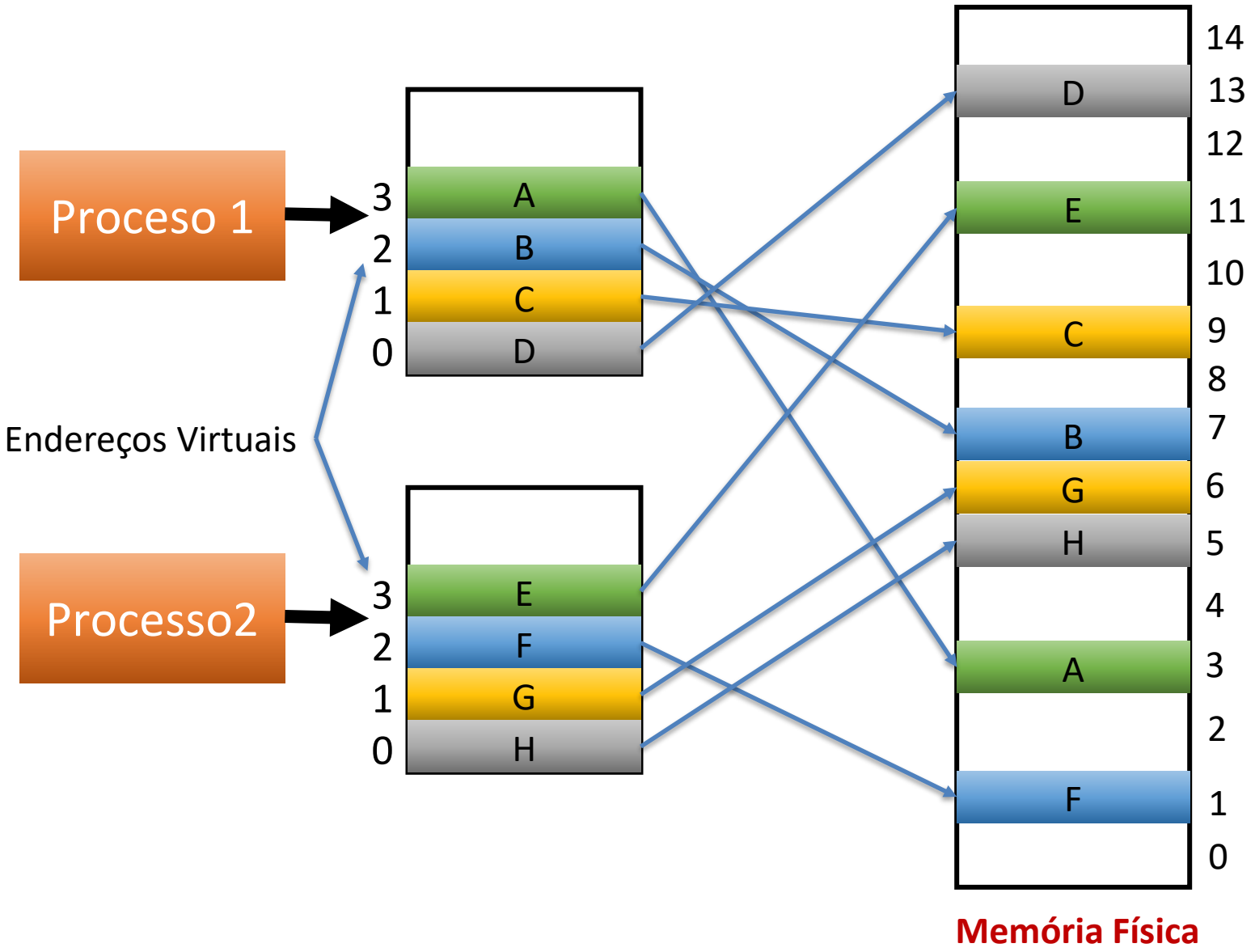
- Em todas as arquiteturas modernas a memória e acessada virtualmente por meio de um endereco virtual. Endereços virtuais requerem tradução.





Para Entender Melhor





Processo 1 precisa do dado em C

Para o Processo C esta dado está no End. 1

Logo a CPU solicita o End. 1

Esta solicitação é interceptada pelo MMU

O MMU sabe que este é um End. Virtual

O MMU olha a tabela de Mapeamento

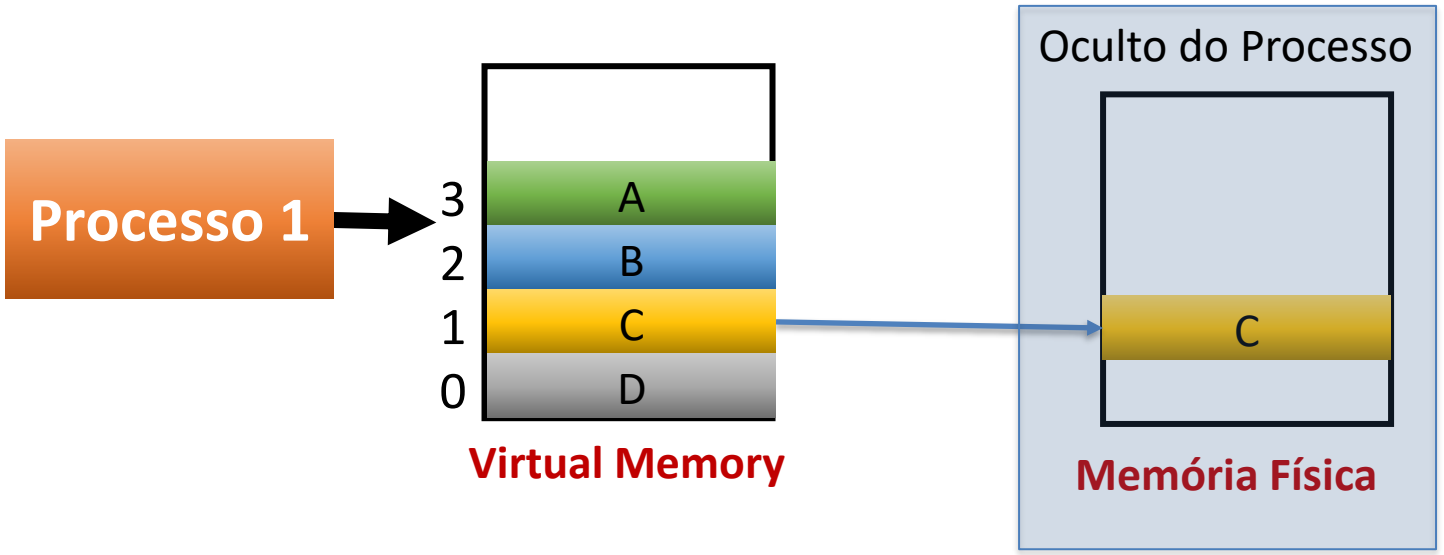
End. Virtual 1 -> End Físico 9

O Dado do Endereço 9 é enviado a CPU

E este dado é o dado C

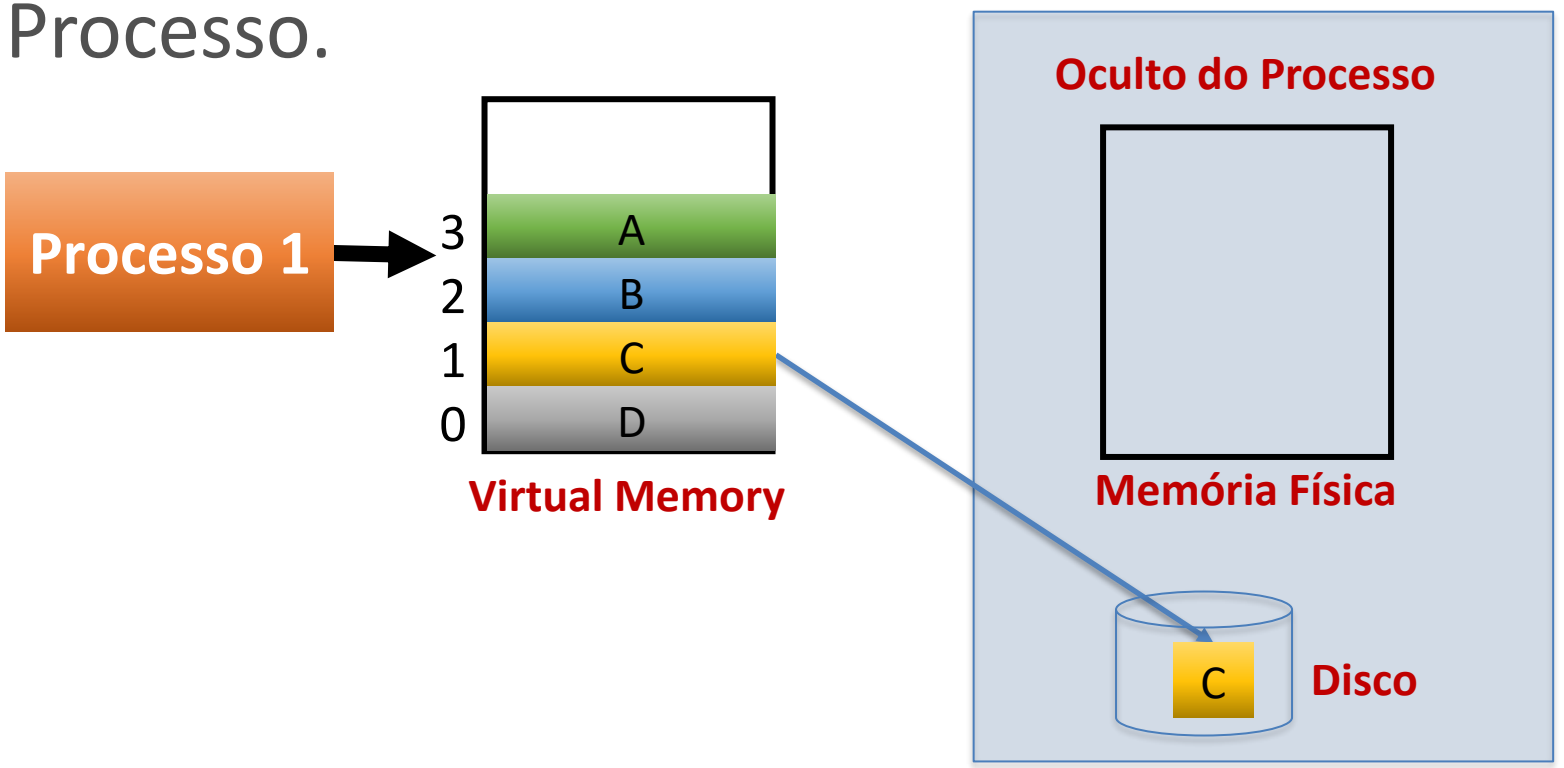
Do ponto de vista do Processo

- O processo só acessa a memória virtual. Contínua, não precisamos recompilar nada só atualizar o mapeamento.

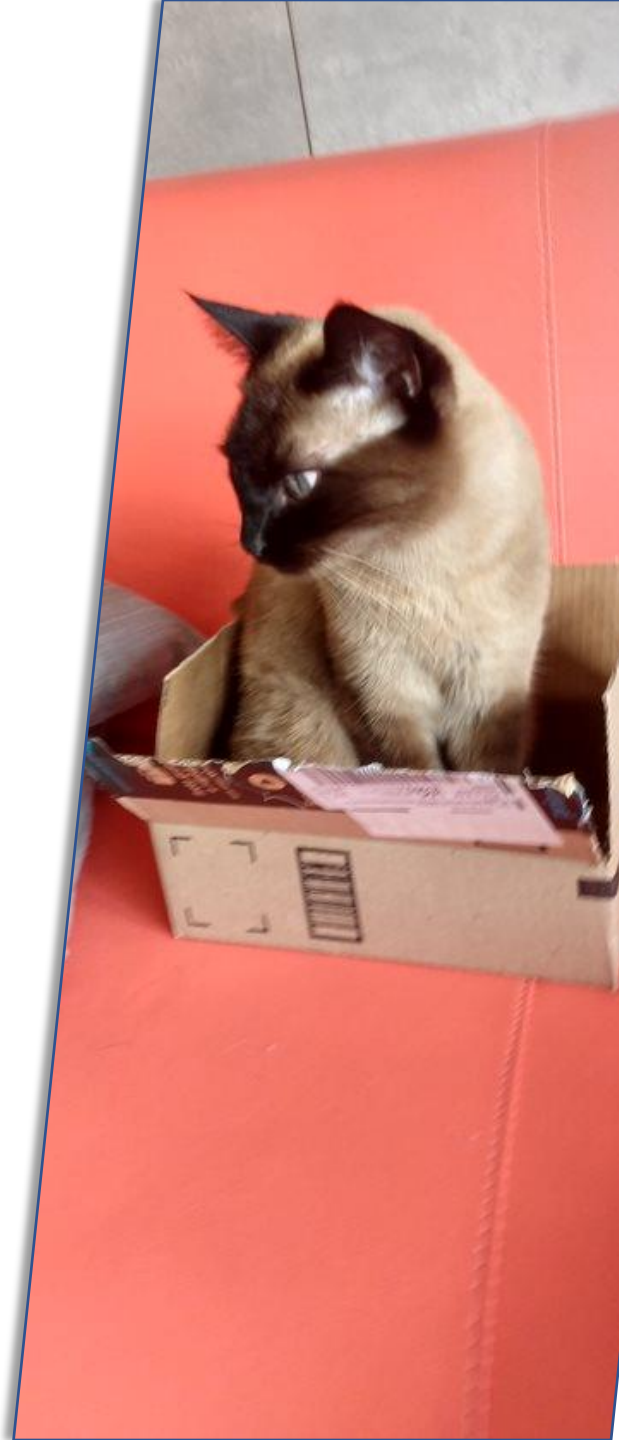


Do ponto de vista do Processo

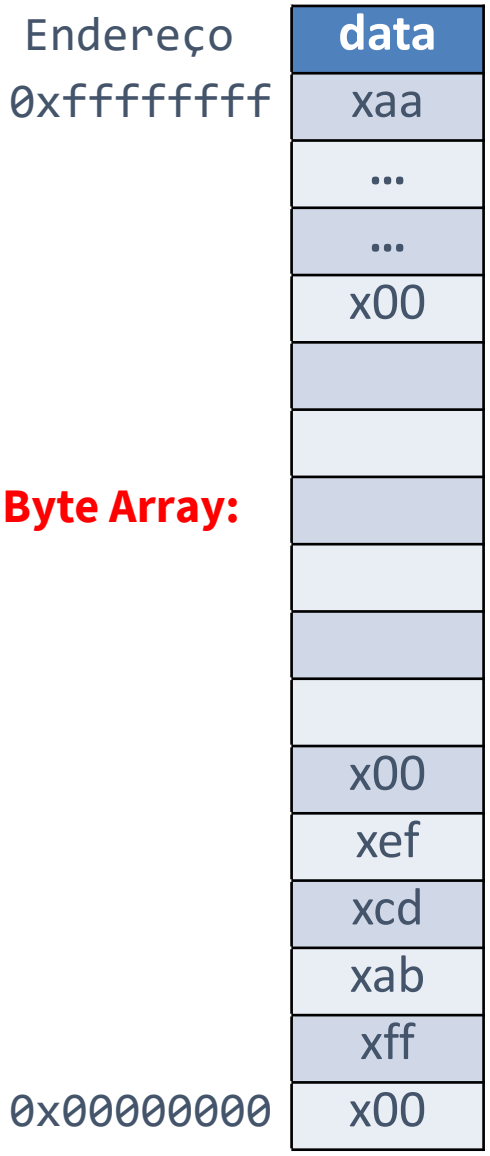
- Se o espaço físico acabar o MMU fará o mapeamento diretamente no disco. O que continua sendo transparente para o Processo.

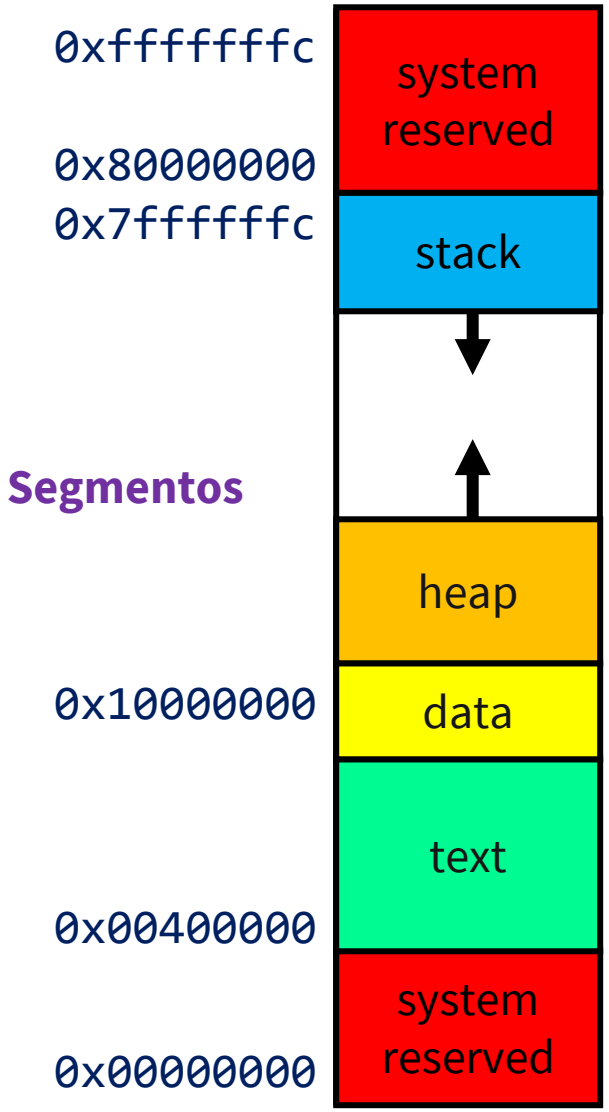


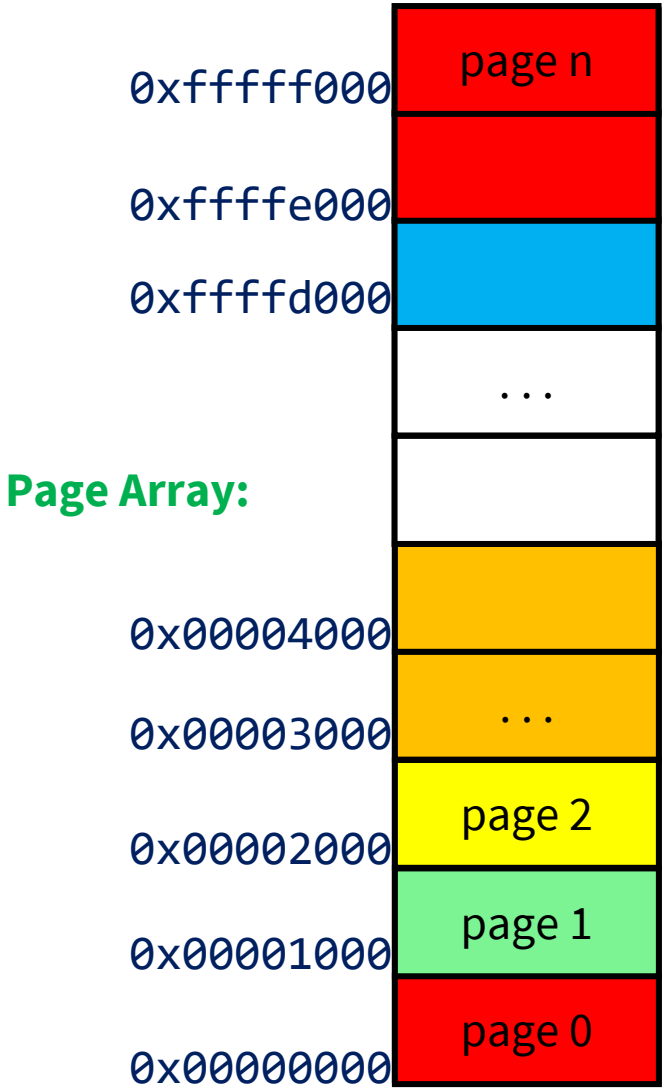
Olhares Quantitativos



Uma Memória. Muitas faces

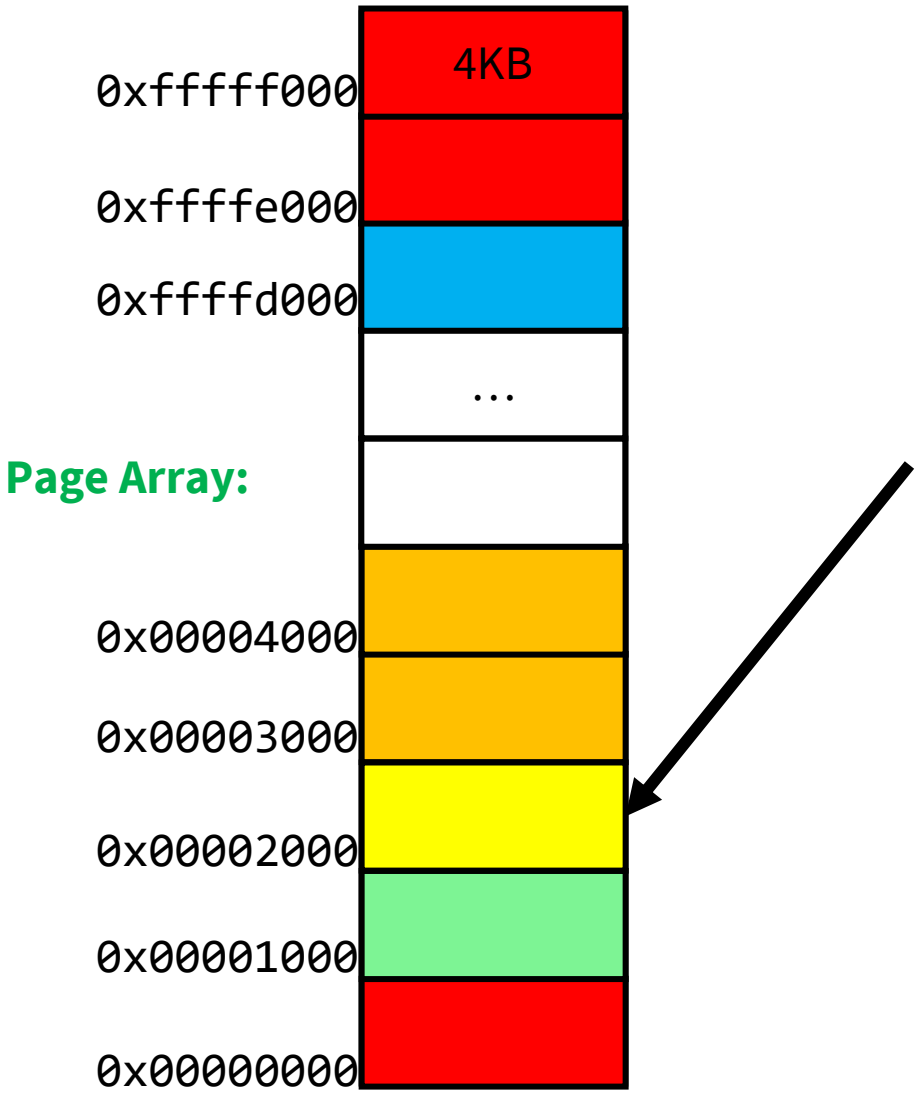






Cada segment usa o mesmo número de páginas

Olhando as Páginas com mais Cuidado



Tamanho da memória = depende do Sistema (4GB)

Page size = 4KB (por padrão)

Logo, número de páginas = 2^{20}

Qualquer dado na página # 2 tem o endereco na forma: `0x00002xxx`

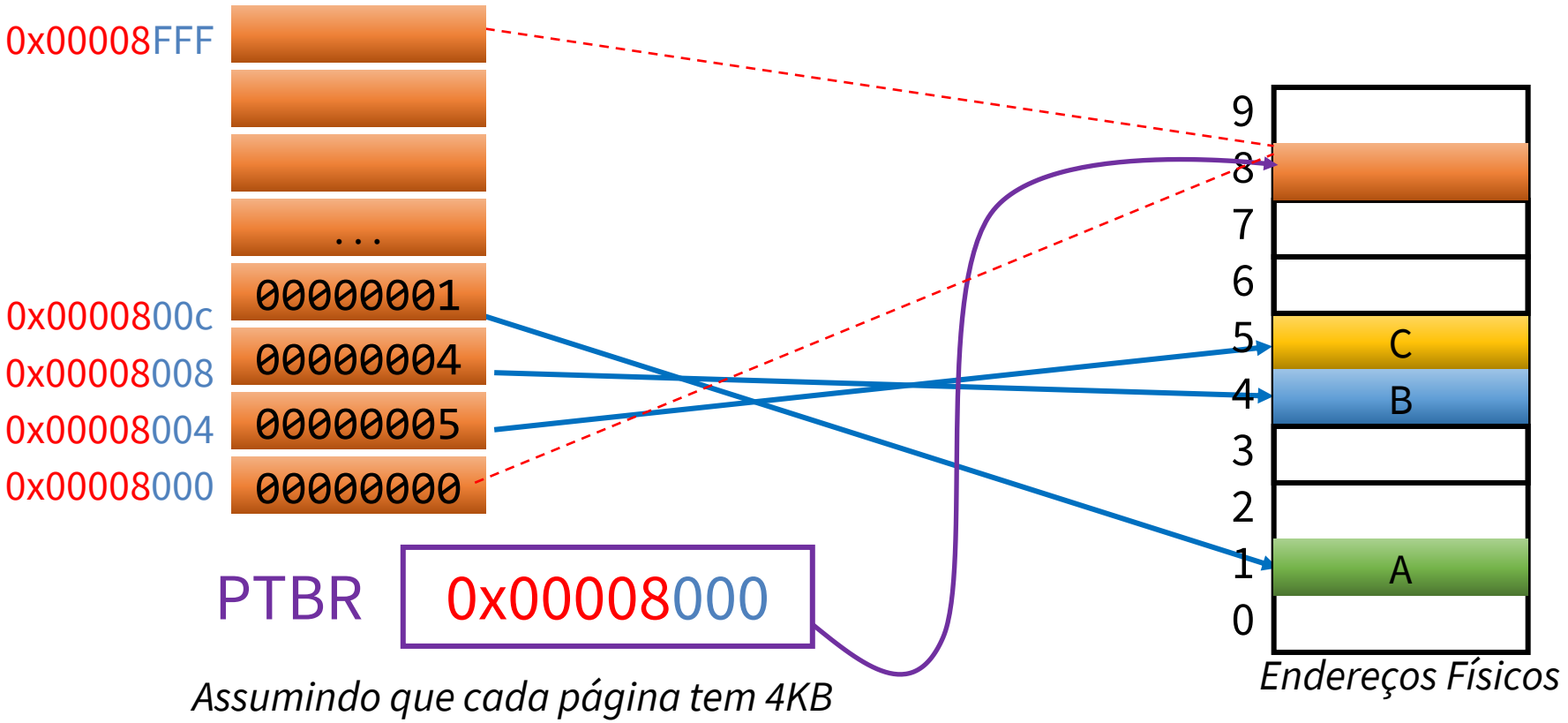
Os 12 bits menos significativos especificam em que Byte estamos nesta página:

`0x00002200` = `0010 0000 0000`
= byte 512

Os bits mais significativos: número da página (PPN)
Os bits menos significativos = offset

Mapeando Endereços

- Uma *Page Table* por processo: **Page Table Base Register (PTBR)**



Assumindo que cada página tem 4KB

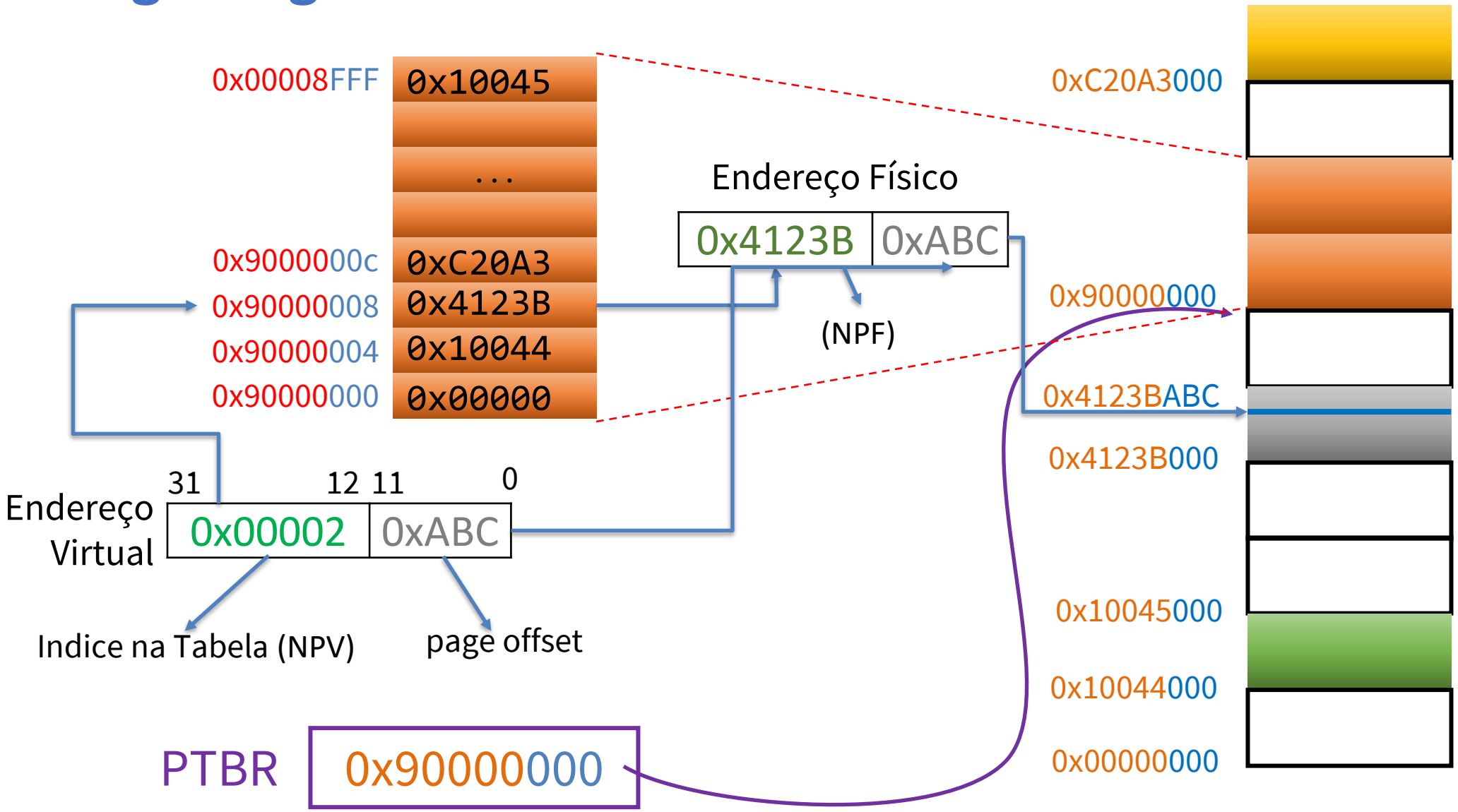
Single Page Translation

- **Divisão do endereço virtual**: O endereço virtual é dividido em duas partes: o **número da página virtual (NPV)** e o deslocamento (offset) dentro da página. Com um tamanho de página de 4KB (2^{12}Bytes), o deslocamento consiste nos 12 bits menos significativos do endereço virtual.
- **Consulta à tabela de páginas**: O Sistema Operacional mantém uma tabela de páginas para cada processo, que mapeia os NPVs para os números de página física correspondentes (NPFs). O hardware (geralmente a MMU) procura o NPV na tabela de páginas do processo para encontrar o NPF associado.

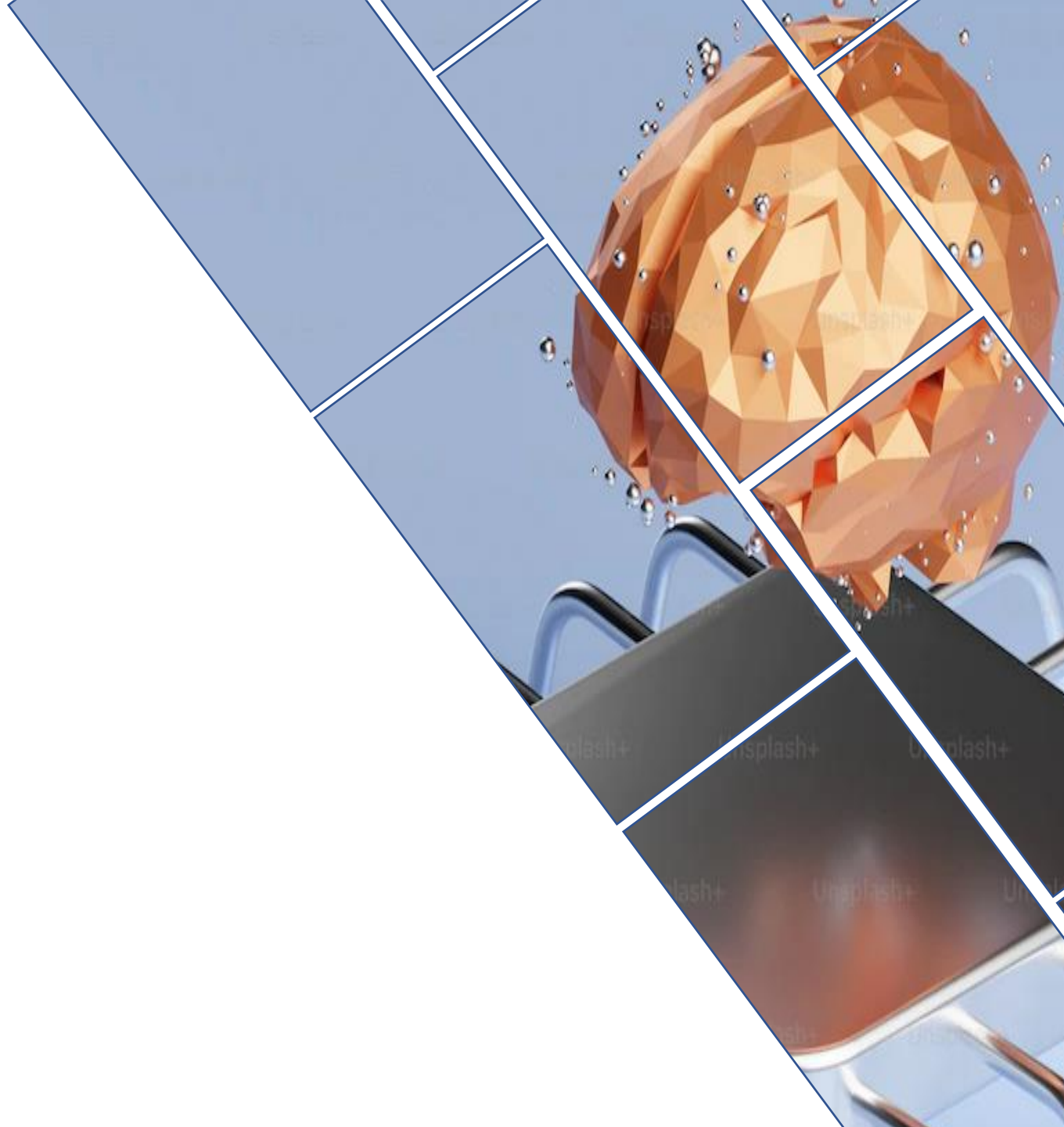
Single Page Translation

- **Construção do endereço**: O endereço físico traduzido é construído substituindo a parte do NPV do endereço virtual pelo NPF obtido na consulta à tabela de páginas e preservando a parte do deslocamento.
- **Acesso à memória física**: O sistema acessa a memória física usando o endereço físico traduzido para realizar a operação de leitura ou gravação solicitada.

Single Page Translation



Obrigado!



Frank Coelho de Alcantara – 2023 -1