



**ARQUITETURA E ORGANIZAÇÃO DE
COMPUTADORES**

**SISTEMAS OPERACIONAIS:
GERENCIAMENTO DE PROCESSOS
E DE MEMÓRIA**

Prof. Dr. Daniel Caetano

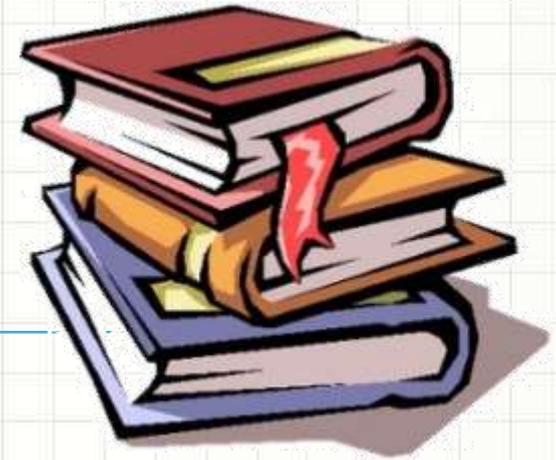
2012 - 1

Lembretes

- Apresentar o gerenciamento de processos
- Compreender a importância do timer e da interrupção
- Apresentar o gerenciamento de memória
- Compreender os recursos básicos da MMU
- **Lembrete:**
 - Lista 2 Online!



Material de Estudo



Material

Acesso ao Material

Notas de Aula

<http://www.caetano.eng.br/aulas/aoc/>
(Aula 9)

Apresentação

<http://www.caetano.eng.br/aulas/aoc/>
(Aula 9)

Material Didático

Arquitetura e Organização de Computadores, páginas
239 a 285

Biblioteca Virtual

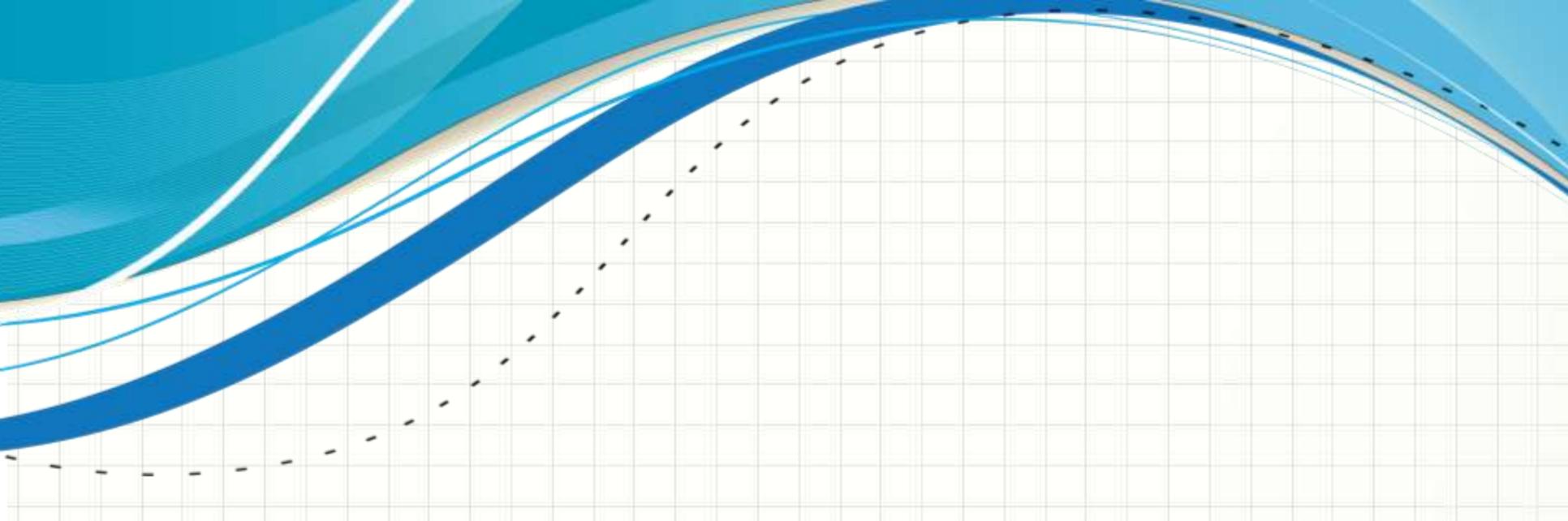
-



INTRODUÇÃO

Introdução

- Aula Passada: Função Geral do S.O.
- Gerenciador de Processos?
 - Tem algo a ver com hardware?
- Gerenciador de Memória?
 - Tem algo a ver com hardware?



GERENCIAMENTO DE PROCESSOS

Gerenciamento de Processos

- Início: 1 computador = 1 programa
- Acesso direto aos recursos
- Vários nomes: programas, processos, *jobs*...



Gerenciamento de Processos

- Problema: 1 computador = **N** programas
- Convivência de programas... novos e antigos
- Programas antigos convivendo?



Gerenciamento de Processos

- Problemas com programas antigos
- Comandos antigos
- Programas antigos

Briga!



Gerenciamento de Processos

- O S.O. tem que “rebolar”
 - Enganar cada programa...
 - para que ele pense que está sozinho!



Gerenciamento de Processos

- Para ser enganado...
 - O computador precisa estar configurado...
 - ...de maneira específica para cada programa
- Analogia:
 - República de Estudantes com um “gerente”
 - Situação:
 - Estudante 1 vendo futebol na TV
 - Estudante 2 ouvindo AC/DC no último volume no som
 - Como o “gerente” compatibiliza?

Gerenciamento de Processos

- Situação:
 - Estudante A vendo futebol na TV
 - Estudante B ouvindo AC/DC no último volume no som
- Como o “gerente” compatibiliza?
- Dividir em “um tempo” para cada estudante
- Momento 1: Estudante A
 - Liga TV no canal do futebol
 - E desliga o som do AC/DC
- Momento 2: Estudante B
 - Desliga TV
 - Liga o som do AC/DC

Gerenciamento de Processos

- Assim...
 - Não basta saber quem é o estudante...
 - ...é preciso saber o que ele estava fazendo...
 - ...e a configuração do ambiente
- Da mesma forma...
 - Não basta saber qual programa está rodando...
 - ...é preciso saber em que ponto ele está...
 - (em qual instrução)
 - ...e a configuração do computador
 - (quais partes da memória ele pode usar, por exemplo)

Gerenciamento de Processos

- S.O.: Manter controle sobre
 - Programas
 - **Estado** do computador para cada programa
- Conjunto: programa + estado = **processo**
- Programa → está no disco, sem executar
- Processo → na memória, em execução

Processo	MAIN.EXE	TEST.EXE	FULL.EXE	OUTRO.EXE
PID	0	1	2	3
Endereço	0x2000	0x4000	0x3000	0x4500
Som?	0	0	1	0

Gerenciamento de Processos

- S.O.: ...

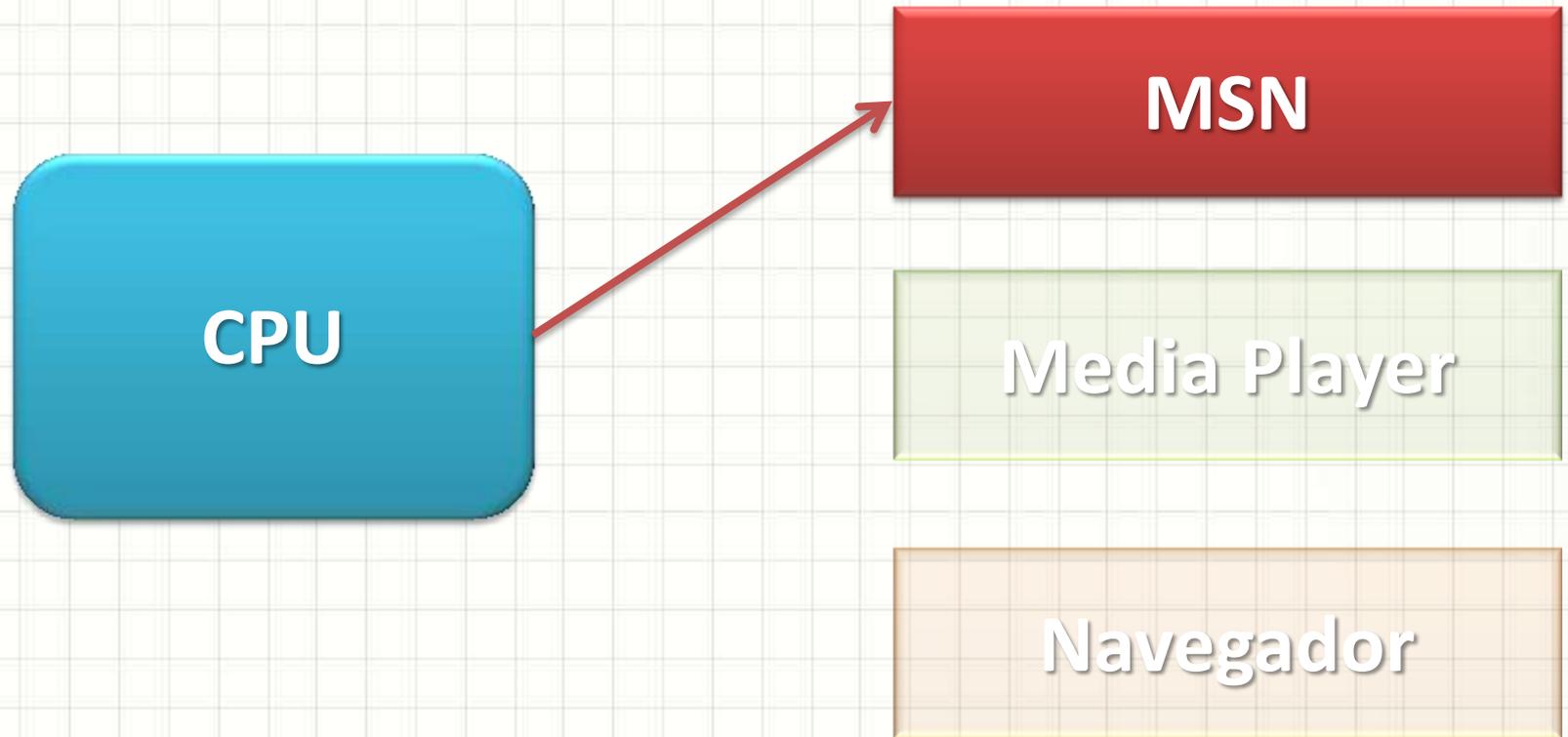
PCB: Process Control Block

- Conjunto de dados = processo
- Programa → est no disco, sem executar
- Processo → na memória, em execução

Processo	MAIN.EXE	TEST.EXE	FULL.EXE	OUTRO.EXE
PID	0	1	2	3
Endereço	0x2000	0x4000	0x3000	0x4500
Som?	0	0	1	0

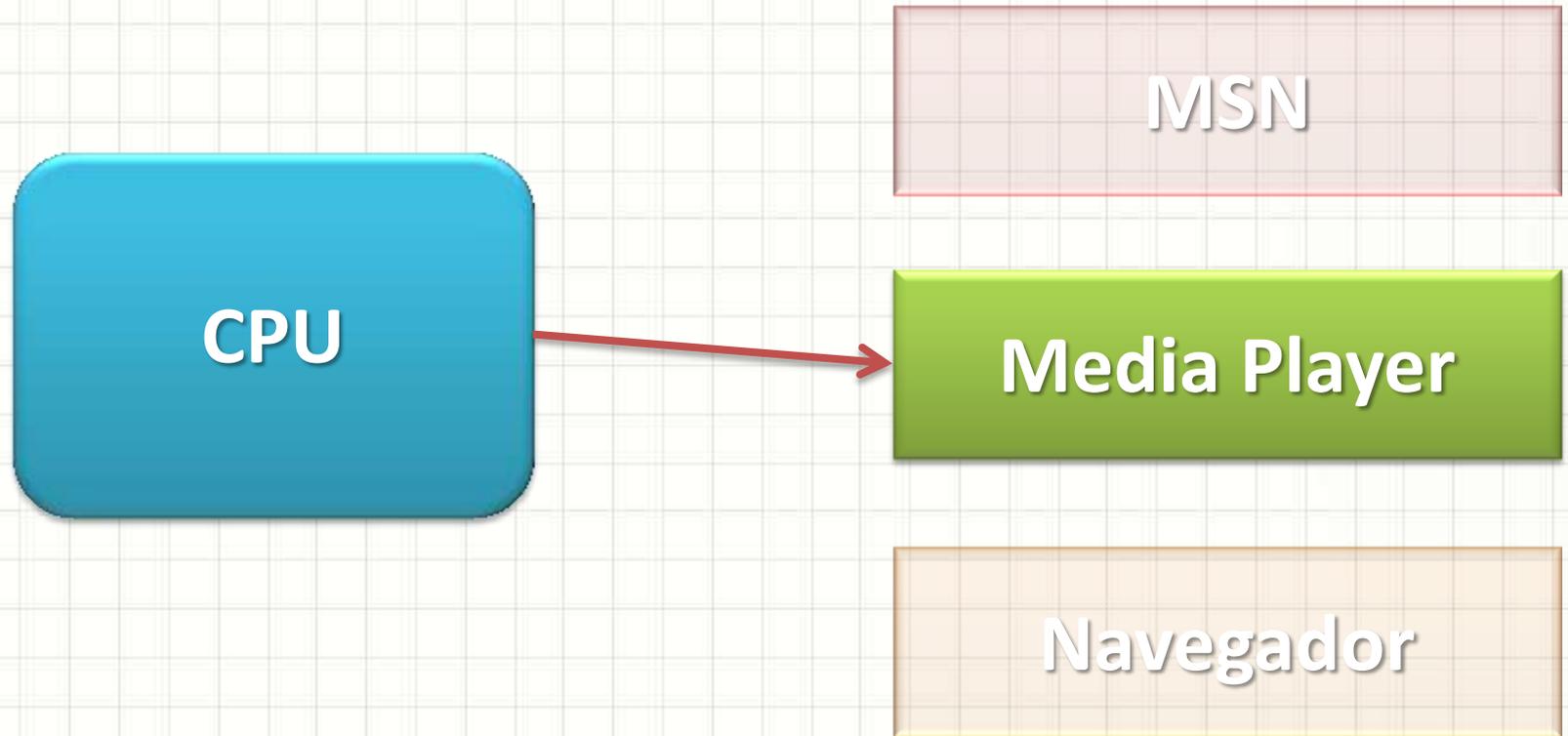
Gerenciamento de Processos

- Uma CPU: Um Processo em Execução por vez



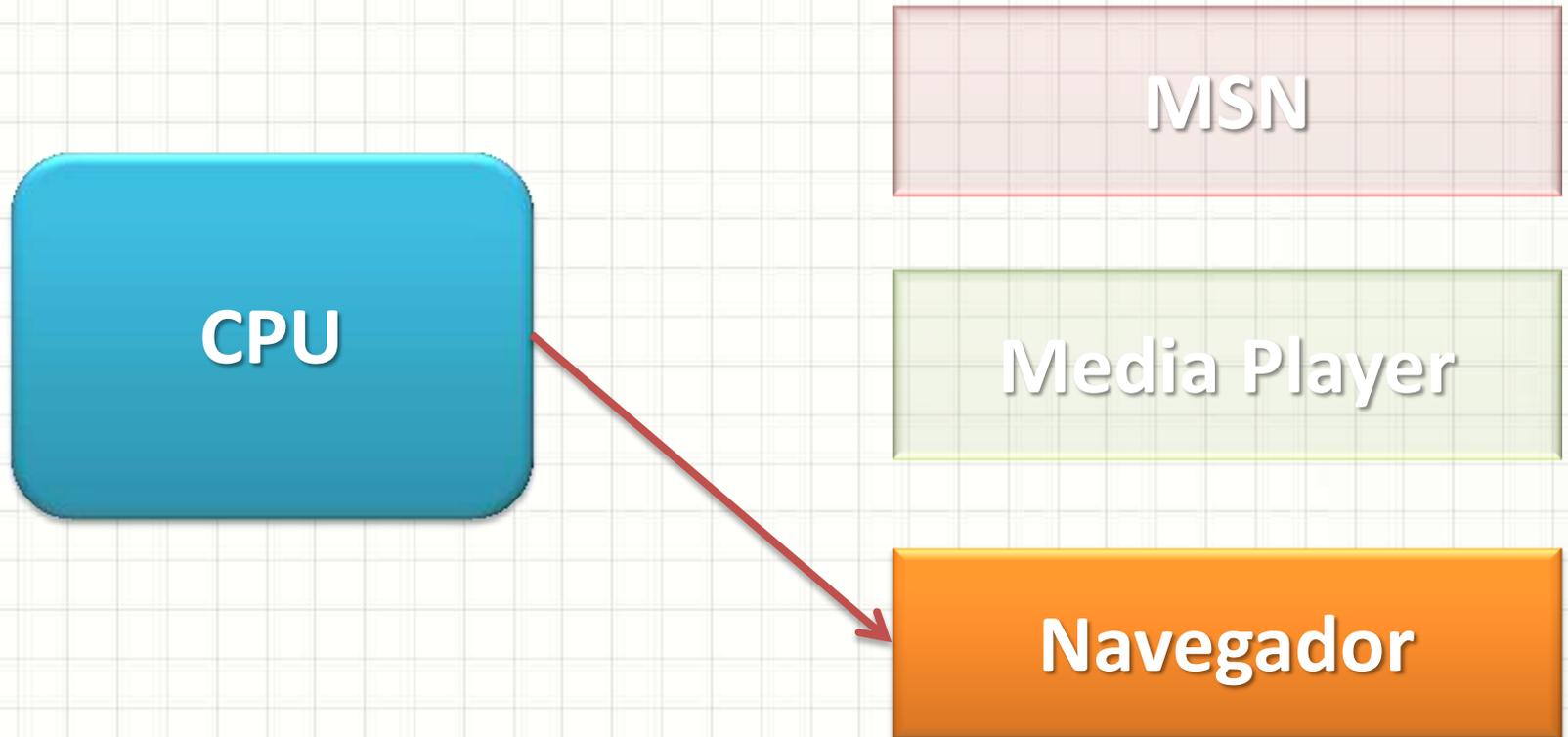
Gerenciamento de Processos

- Uma CPU: Um Processo em Execução por vez



Gerenciamento de Processos

- Uma CPU: Um Processo em Execução por vez

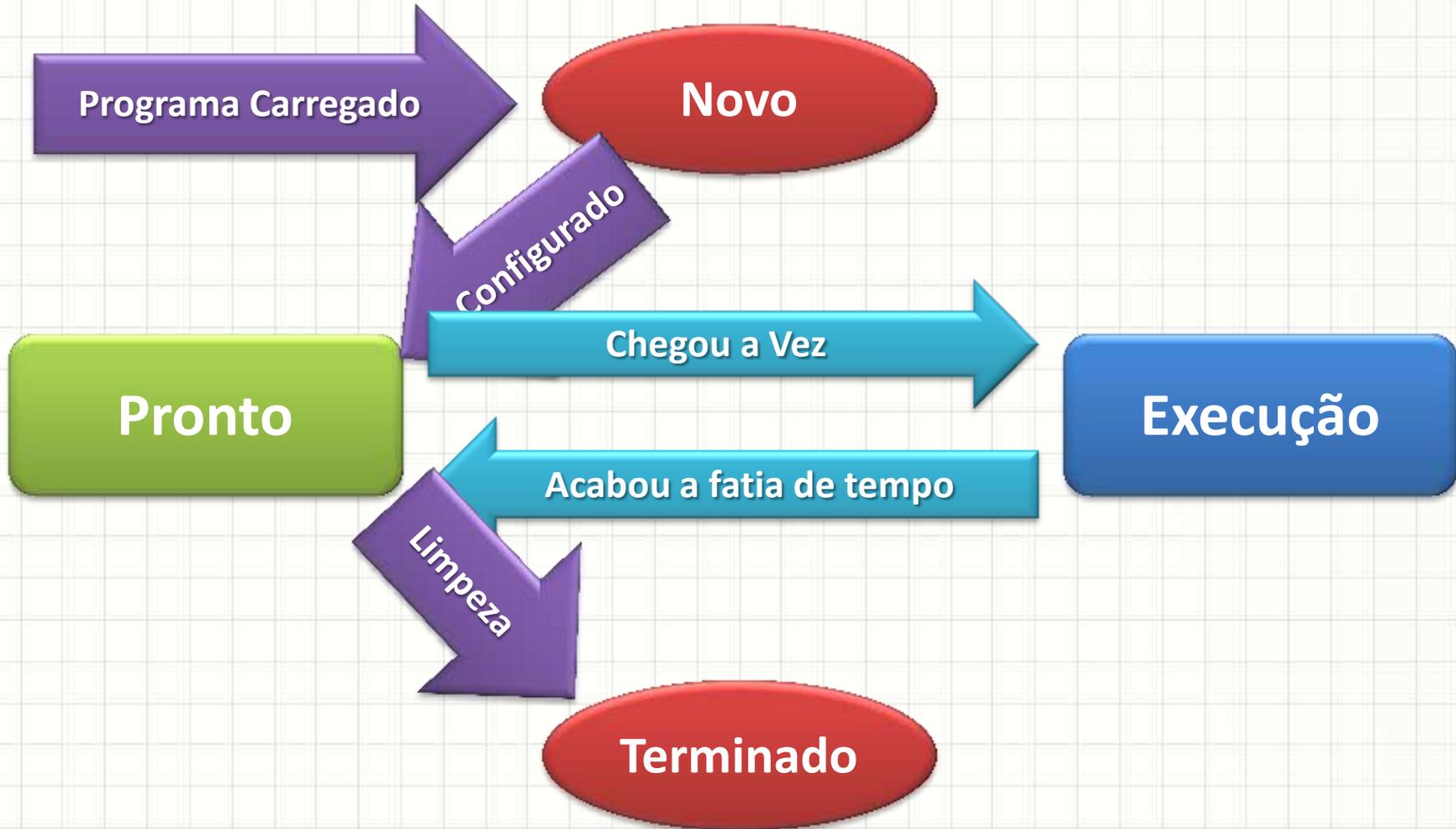


Estados dos Processos

- Estado de Execução: 5 básicos
 - Em **Execução**: está processando no momento
 - **Pronto** para executar: aguarda sua vez de processar
 - **Espera / Bloqueado**: esperando recurso
 - **Novo**: ainda está sendo configurado
 - **Terminado**: recursos sendo liberados
- Os processos **mudam** de estado

Estados dos Processos

- Ciclo de Vida Básico



Estados dos Processos

- Ciclo de Vida

**Como o S.O.
sabe que
“chegou a vez”?**

Pronto

Execução

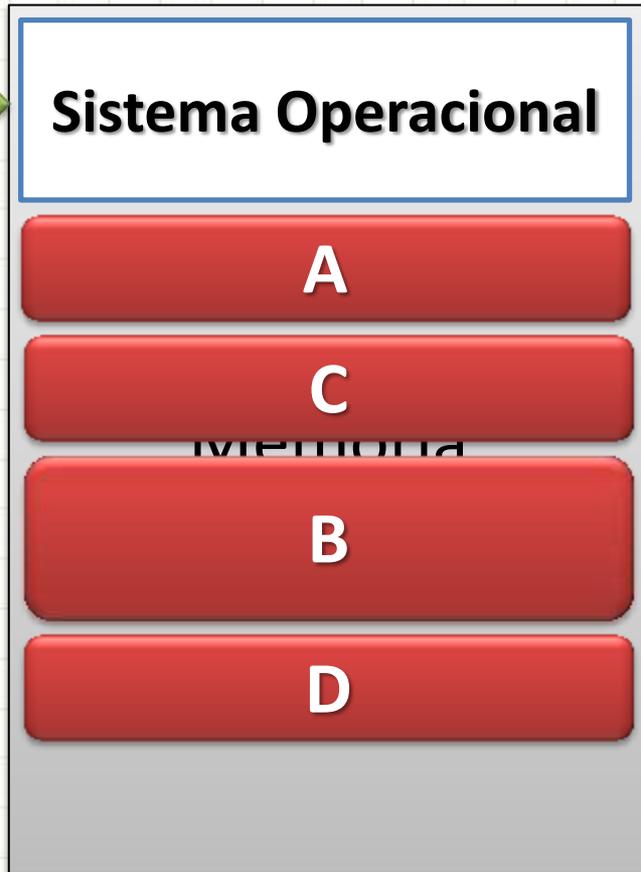
ou a fatia de tempo

Limpeza

Terminado

Escalonamento de Processos

- PCB: qual processo está em execução
- “Fila de Execução”



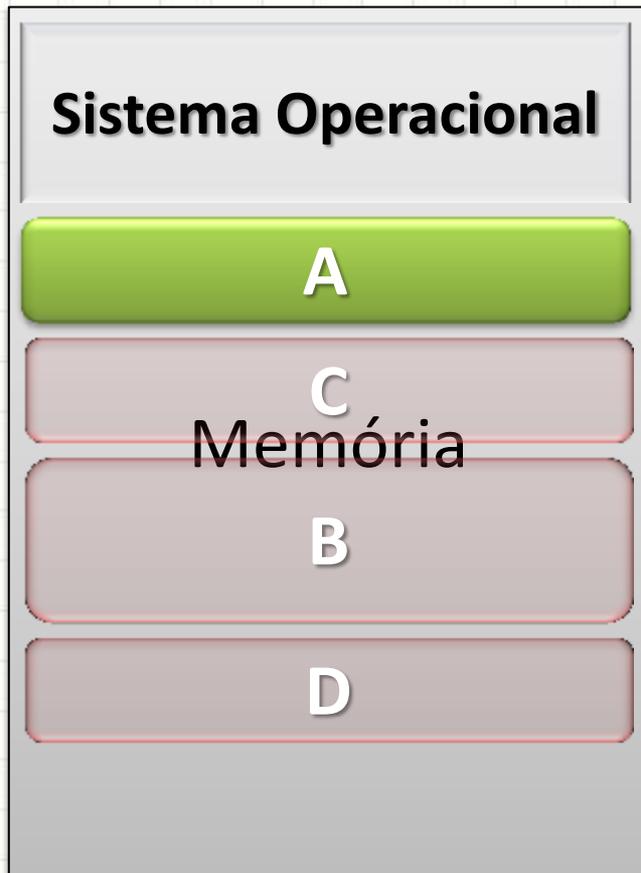
Executando...!



Processo	A.EXE	B.EXE	C.EXE	D.EXE
PID	0	1	2	3
Endereço	0x2000	0x4000	0x3000	0x5500
Som?	0	0	1	0

Escalonamento de Processos

- PCB: qual processo está em execução
- “Fila de Execução”



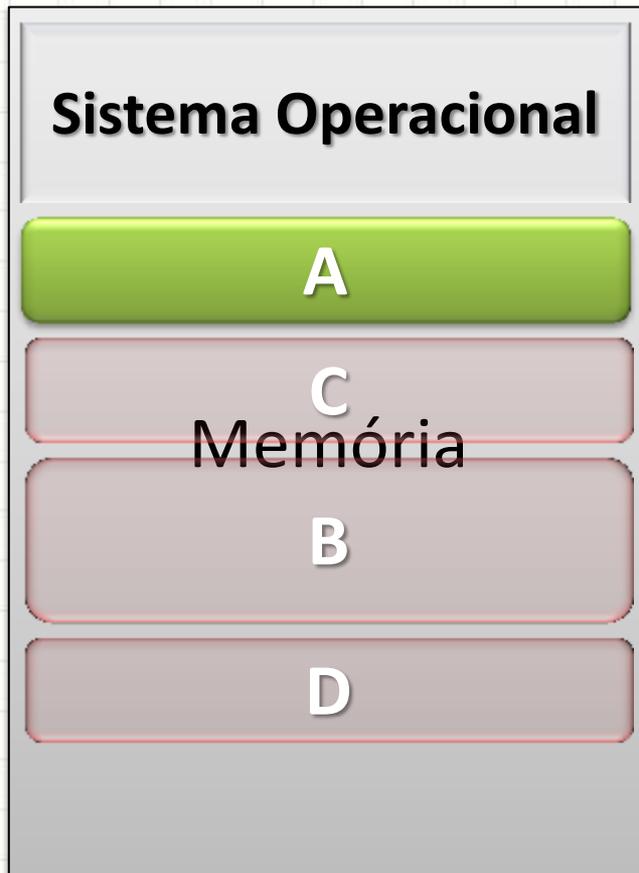
Executando...!



Processo	A.EXE	B.EXE	C.EXE	D.EXE
PID	0	1	2	3
Endereço	0x2000	0x4000	0x3000	0x5500
Som?	0	0	1	0

Escalonamento de Processos

- PCB: qual processo está em execução
- “Fila de Execução”



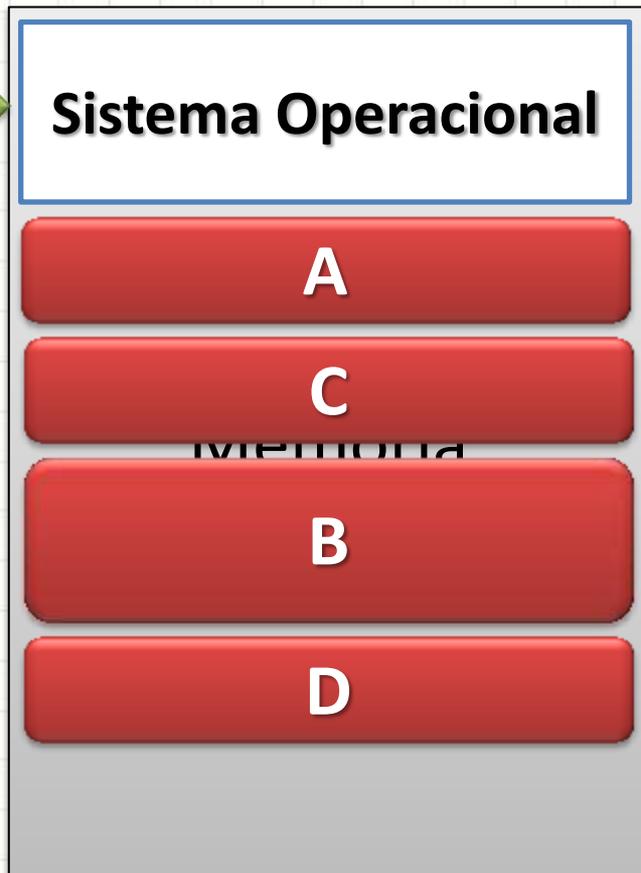
Acabou a fatia de tempo!



Processo	A.EXE	B.EXE	C.EXE	D.EXE
PID	0	1	2	3
Endereço	0x2000	0x4000	0x3000	0x5500
Som?	0	0	1	0

Escalonamento de Processos

- PCB: qual processo está em execução
- “Fila de Execução”



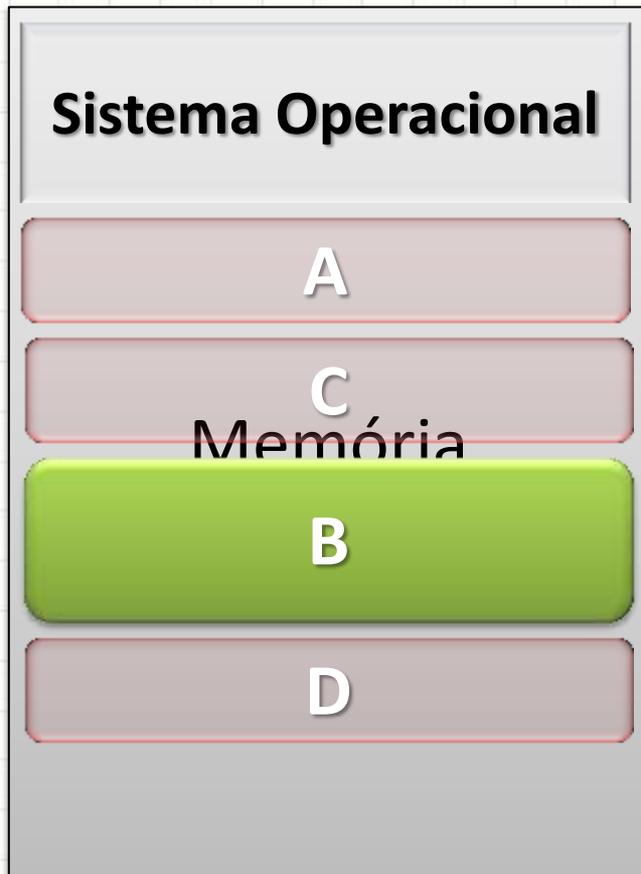
Executando...!



Processo	A.EXE	B.EXE	C.EXE	D.EXE
PID	0	1	2	3
Endereço	0x2000	0x4000	0x3000	0x5500
Som?	0	0	1	0

Escalonamento de Processos

- PCB: qual processo está em execução
- “Fila de Execução”



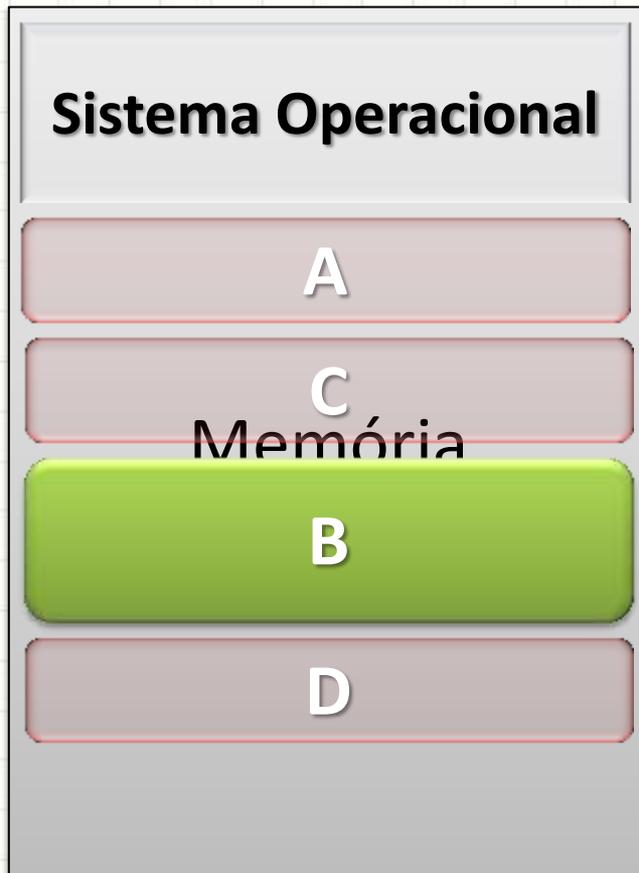
Executando...!



Processo	A.EXE	B.EXE	C.EXE	D.EXE
PID	0	1	2	3
Endereço	0x2000	0x4000	0x3000	0x5500
Som?	0	0	1	0

Escalonamento de Processos

- PCB: qual processo está em execução
- “Fila de Execução”



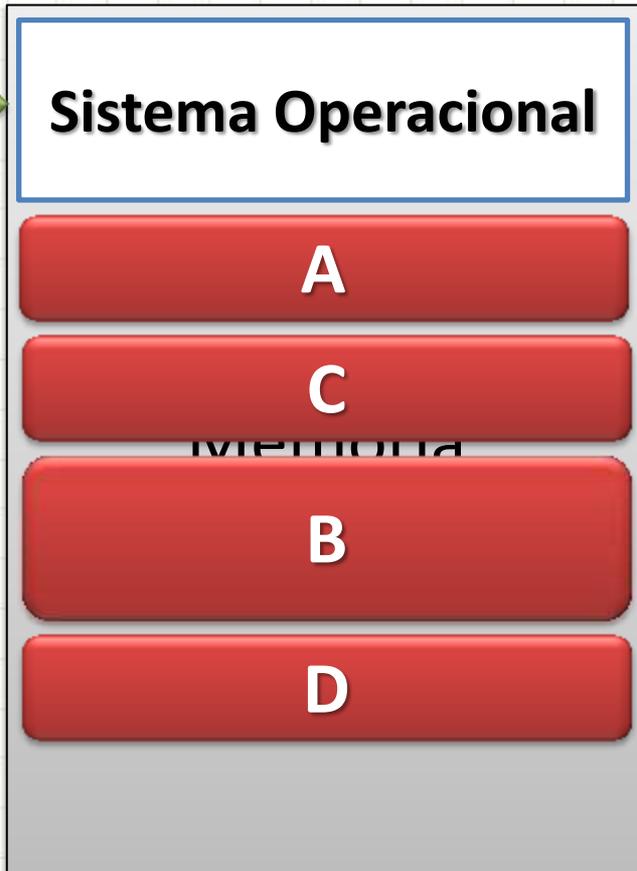
Acabou a fatia de tempo!



Processo	A.EXE	B.EXE	C.EXE	D.EXE
PID	0	1	2	3
Endereço	0x2000	0x4000	0x3000	0x5500
Som?	0	0	1	0

Escalonamento de Processos

- PCB: qual processo está em execução
- “Fila de Execução”



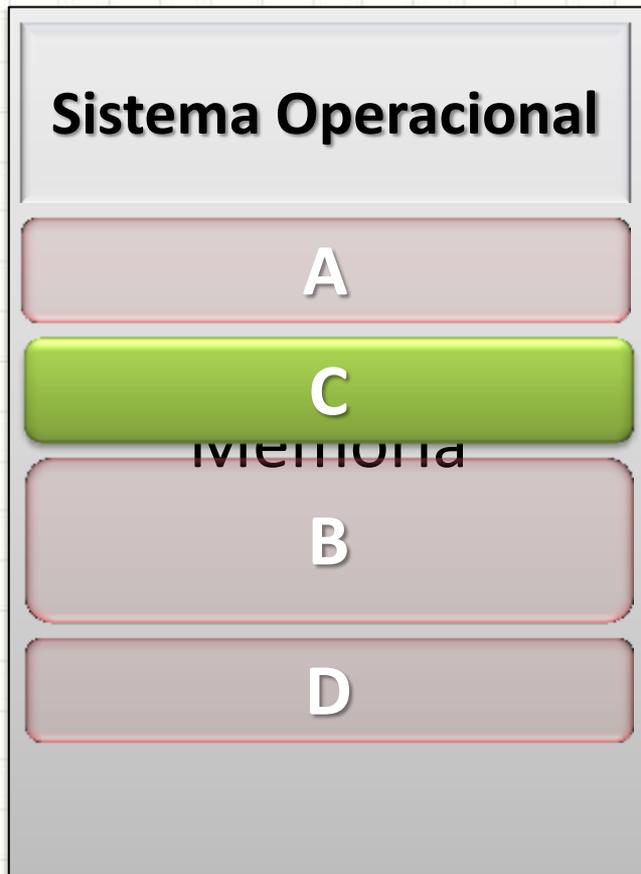
Executando...!



Processo	A.EXE	B.EXE	C.EXE	D.EXE
PID	0	1	2	3
Endereço	0x2000	0x4000	0x3000	0x5500
Som?	0	0	1	0

Escalonamento de Processos

- PCB: qual processo está em execução
- “Fila de Execução”



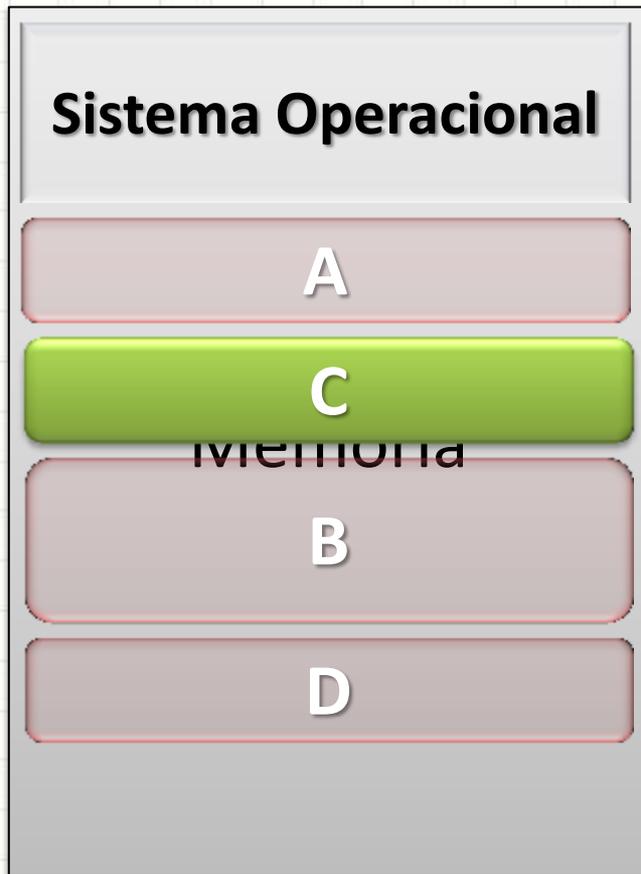
Executando...!



Processo	A.EXE	B.EXE	C.EXE	D.EXE
PID	0	1	2	3
Endereço	0x2000	0x4000	0x3000	0x5500
Som?	0	0	1	0

Escalonamento de Processos

- PCB: qual processo está em execução
- “Fila de Execução”



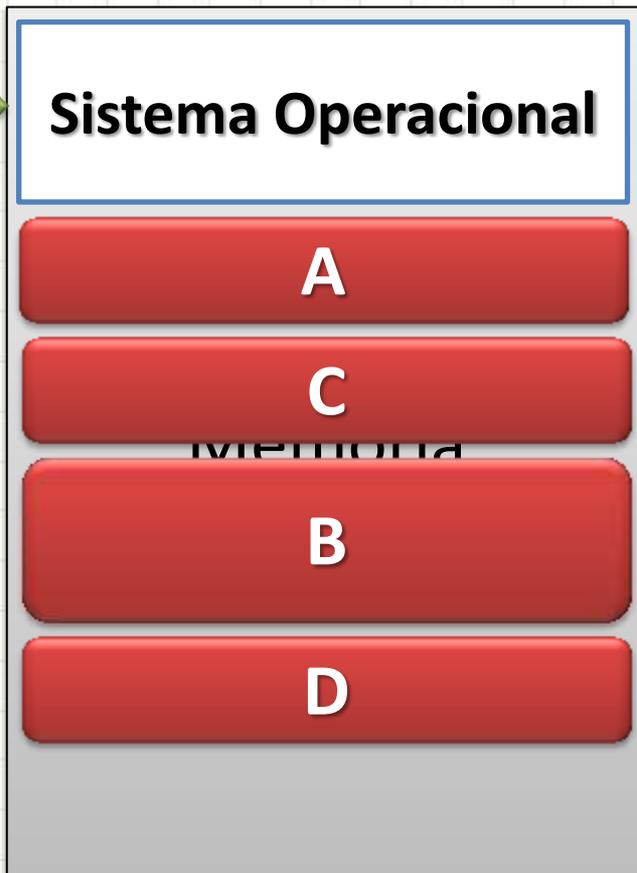
Acabou a fatia de tempo!



Processo	A.EXE	B.EXE	C.EXE	D.EXE
PID	0	1	2	3
Endereço	0x2000	0x4000	0x3000	0x5500
Som?	0	0	1	0

Escalonamento de Processos

- PCB: qual processo está em execução
- “Fila de Execução”



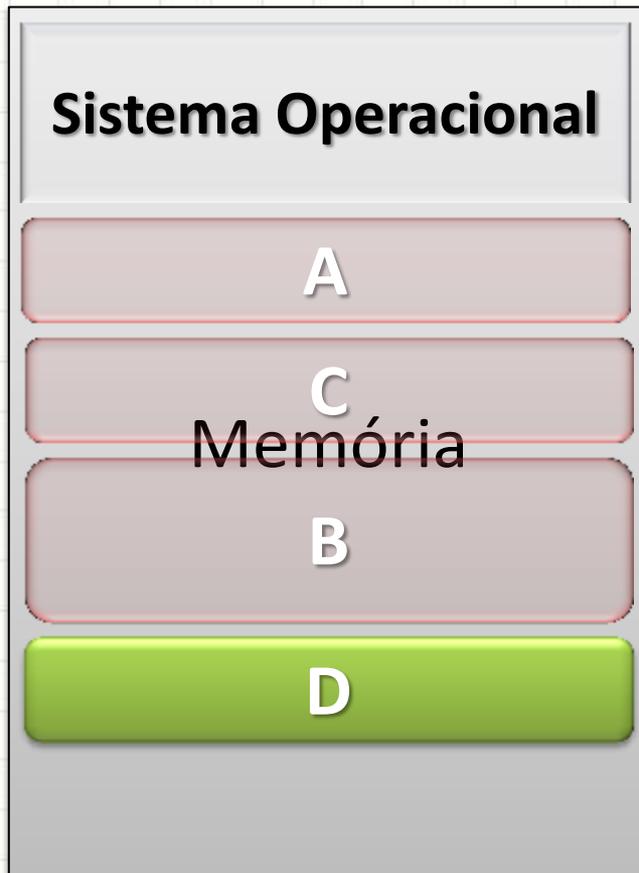
Executando...!



Processo	A.EXE	B.EXE	C.EXE	D.EXE
PID	0	1	2	3
Endereço	0x2000	0x4000	0x3000	0x5500
Som?	0	0	1	0

Escalonamento de Processos

- PCB: qual processo está em execução
- “Fila de Execução”



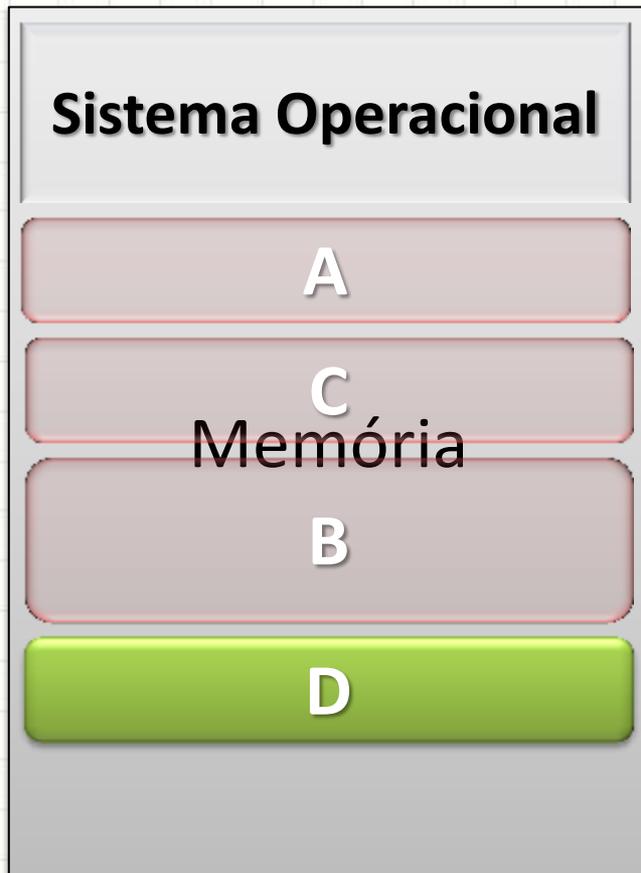
Executando...!



Processo	A.EXE	B.EXE	C.EXE	D.EXE
PID	0	1	2	3
Endereço	0x2000	0x4000	0x3000	0x5500
Som?	0	0	1	0

Escalonamento de Processos

- PCB: qual processo está em execução
- “Fila de Execução”



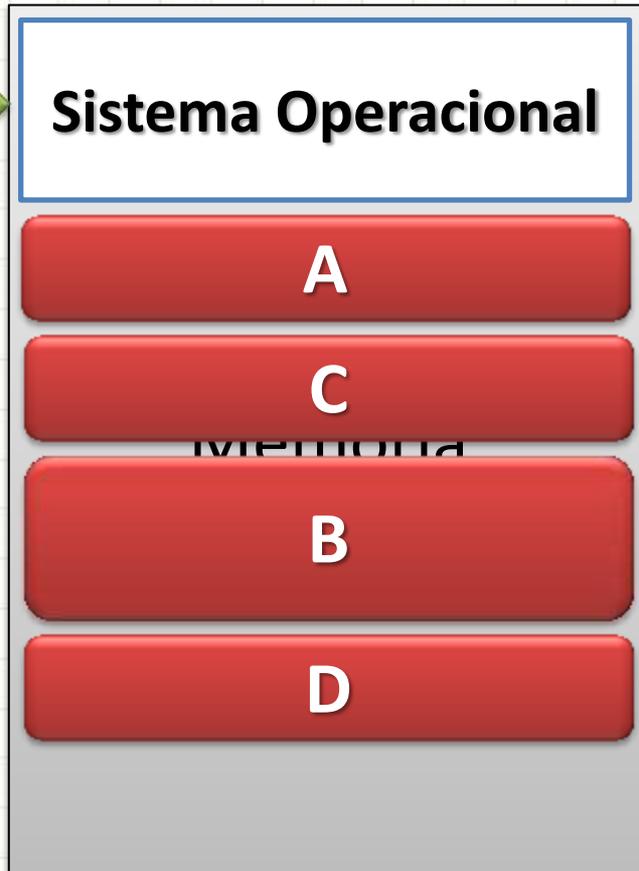
Acabou a fatia de tempo!



Processo	A.EXE	B.EXE	C.EXE	D.EXE
PID	0	1	2	3
Endereço	0x2000	0x4000	0x3000	0x5500
Som?	0	0	1	0

Escalonamento de Processos

- PCB: qual processo está em execução
- “Fila de Execução”



Executando...!



Processo	A.EXE	B.EXE	C.EXE	D.EXE
PID	0	1	2	3
Endereço	0x2000	0x4000	0x3000	0x5500
Som?	0	0	1	0

Escalonamento de Processos

- PCB: ...
- ...

**E assim por
diante!**

Sistema Oper

A

C
Memória

B

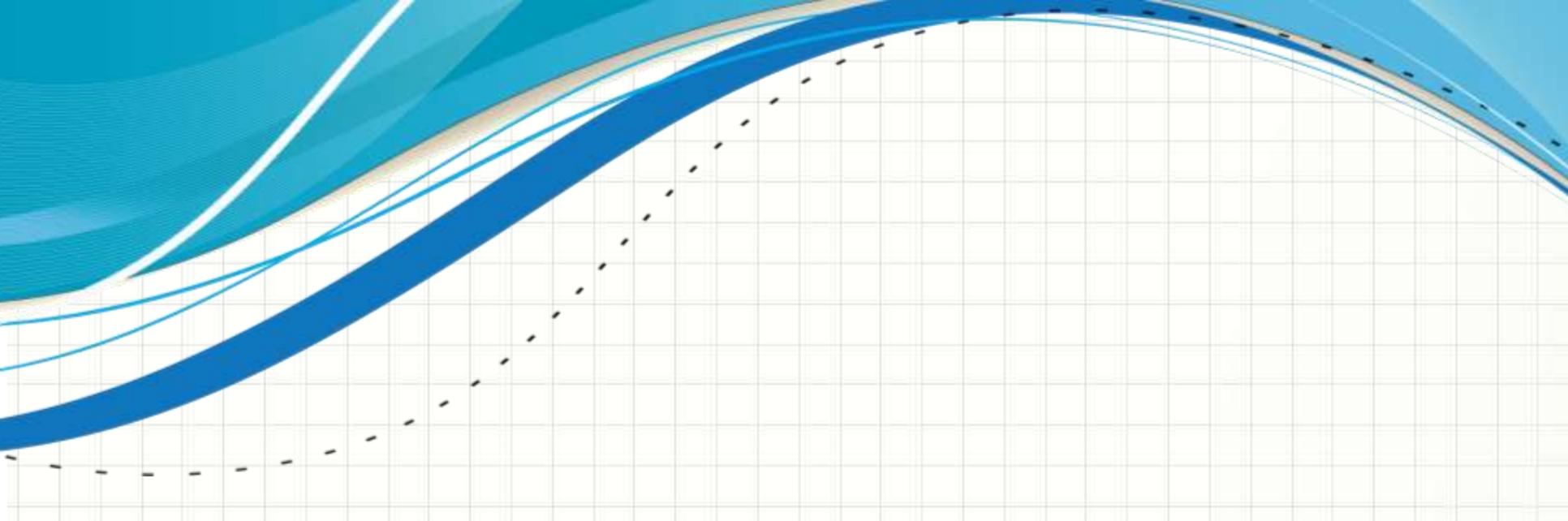
D



Processo	A.EXE	B.EXE	C.EXE	D.EXE
PID	0	1	2	3
Endereço	0x2000	0x4000	0x3000	0x5500
Som?	0	0	1	0

Fatia de Tempo (*timeslice*)

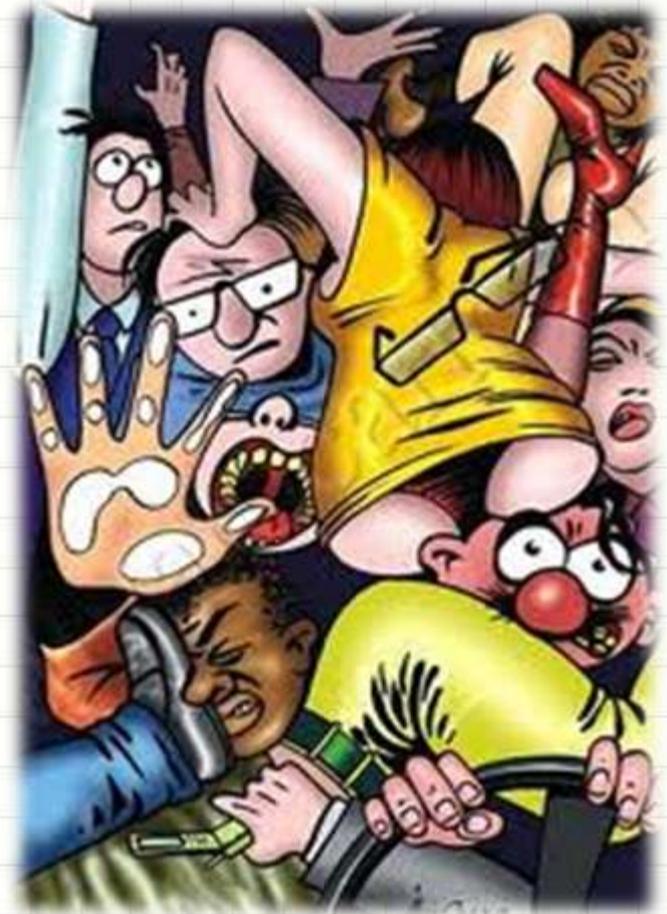
- Quando ocorre a troca de processos?
 - Quando acaba a fatia de tempo de um processo
 - Quando ele finaliza
 - Quando ele tenta ler um periférico lento
- Fatia de Tempo...?
 - Controlada por um *timer* (despertador)
 - *Timer* é configurado pelo S.O. para disparar...
 - após 32ms e coloca um processo em execução
 - Quando *timer* dispara, ocorre uma **interrupção**
 - O controle volta diretamente para o S.O.



GERENCIAMENTO DE MEMÓRIA

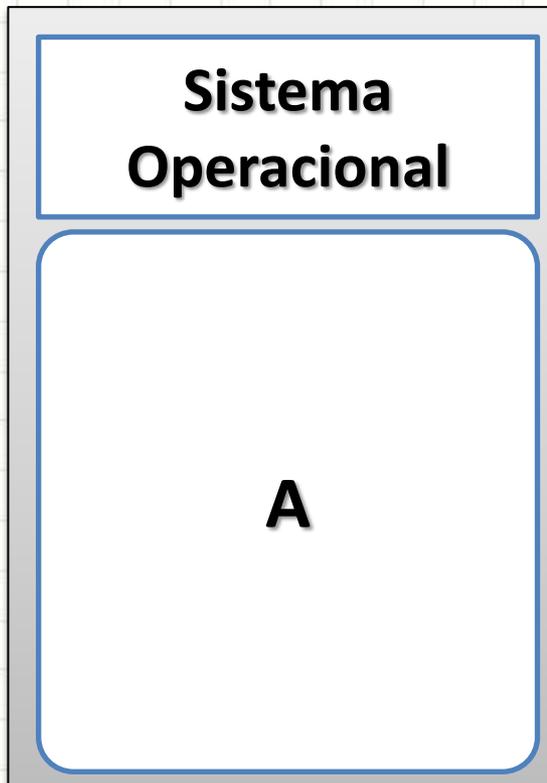
Gerenciamento de Memória

- Por que gerenciar memória?
- Recurso limitado!
- Muitos programas compartilham a mesma memória
- Um pode prejudicar outro?
- Quem cuida disso?
 - O sistema operacional
- Sempre foi assim?



Gerenciamento de Memória

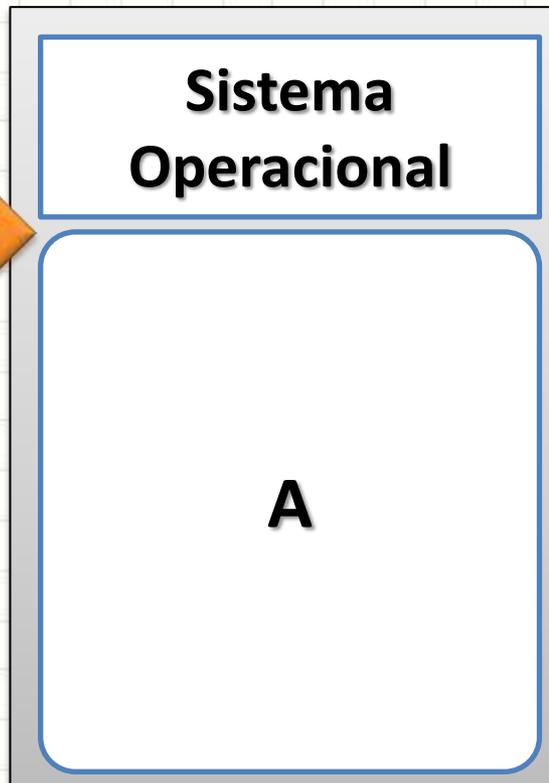
- Inicialmente, praticamente inexistente
 - Sistemas monotarefa
 - Toda a memória de um único programa



- Nada impedia, inclusive, que o programa “destruísse” o S.O.
- Exemplo: alguns videogames
- Analogia: estudante morando sozinho

Gerenciamento de Memória

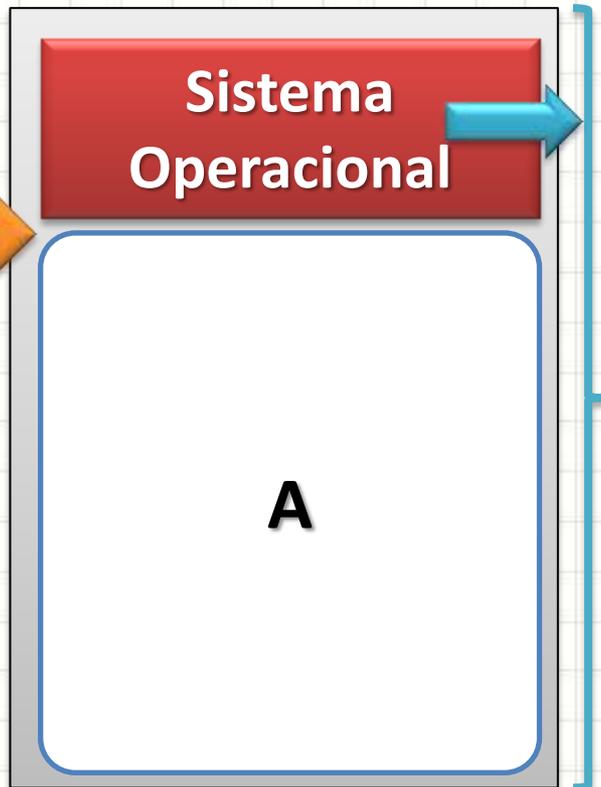
- Surgimento da MMU: Memory Management Unit
 - Sistemas monotarefa / único programa
 - Registrador de Proteção: Acesso apenas do S.O.



- Divide a memória em duas regiões

Gerenciamento de Memória

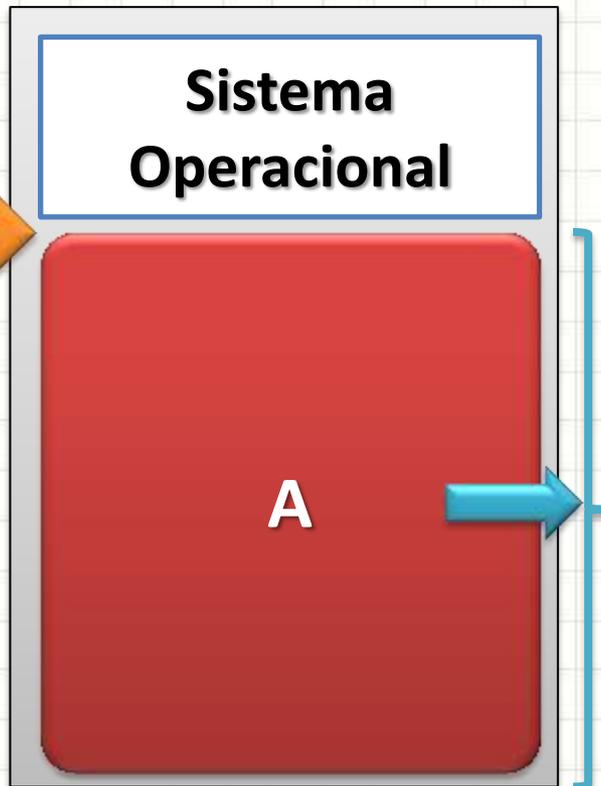
- Surgimento da MMU: Memory Management Unit
 - Sistemas monotarefa / único programa
 - Registrador de Proteção: Acesso apenas do S.O.



- O S.O. fica na primeira região...
- Acessa toda a RAM

Gerenciamento de Memória

- Surgimento da MMU: Memory Management Unit
 - Sistemas monotarefa / único programa
 - Registrador de Proteção: Acesso apenas do S.O.



- Programa na segunda região (processo)...
- Acessa apenas a memória **APÓS** o endereço do registrador de proteção
- Analogia: criança que mora com os pais

Gerenciamento de Memória

- Surgimento da MMU: Memory Management Unit
 - Sistemas monotarefa / único programa
 - Registrador de Proteção



- Programa na segunda região (processo)...
- Se tentar acessar a primeira região...

Gerenciamento de Memória

test.exe - Application Error



The instruction at "0x78027470" referenced memory at "0x00000008". The memory could not be "written".

Click on OK to terminate the program
Click on CANCEL to debug the program

OK

Cancel

Sistema Operacional

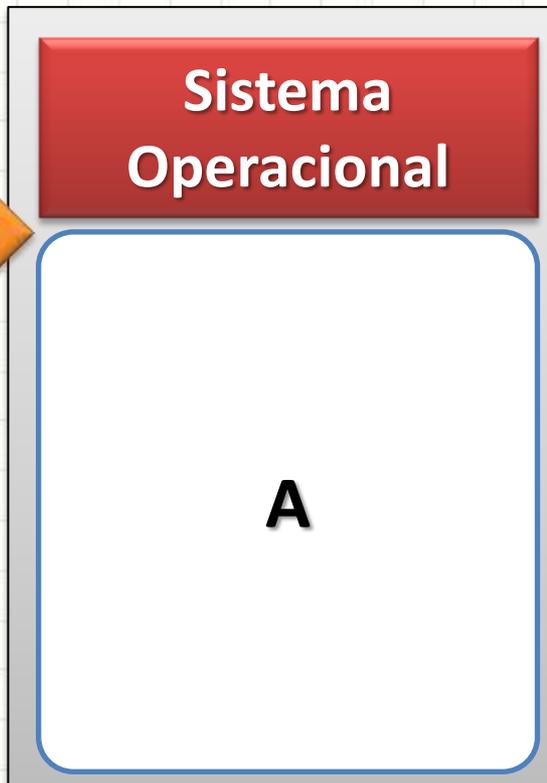
A

- Programa na segunda região (processo)...
- Se tentar acessar a primeira região...
- Ocorre uma falha de proteção...

0x2000

Gerenciamento de Memória

- Surgimento da MMU: Memory Management Unit
 - Sistemas monotarefa / único programa
 - Registrador de Proteção



- Programa na segunda região (processo)...
- Se tentar acessar a primeira região...
- Ocorre uma falha de proteção...
- Que transfere o controle para o S.O.

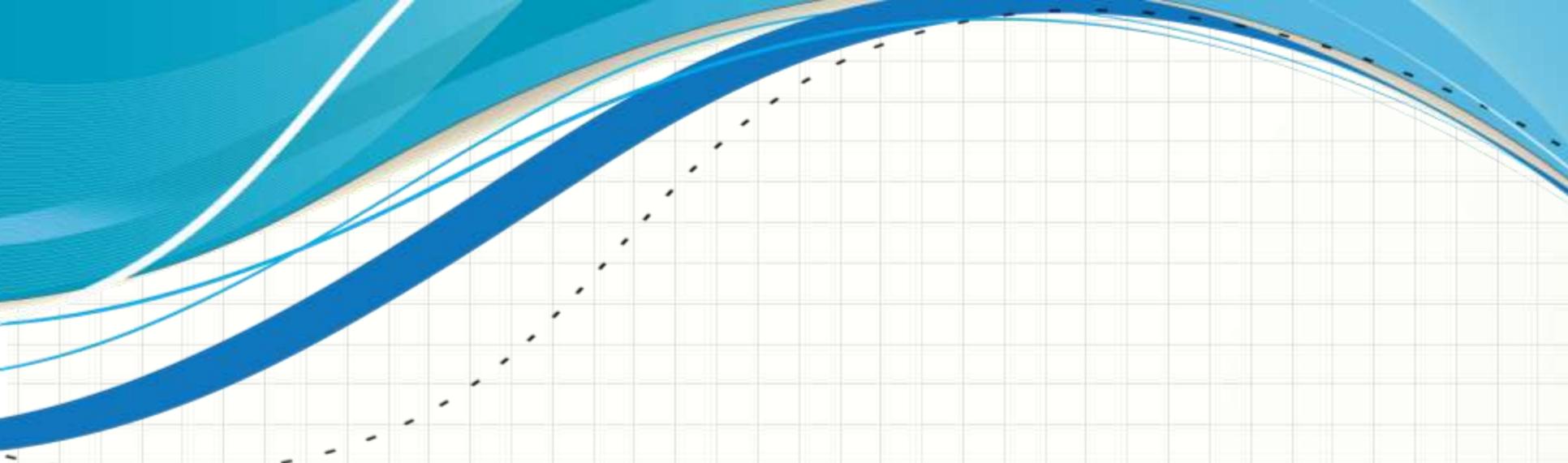
Gerenciamento de Memória

- ...

**O programa causou
uma operação ilegal
e foi finalizado!**

0x2000

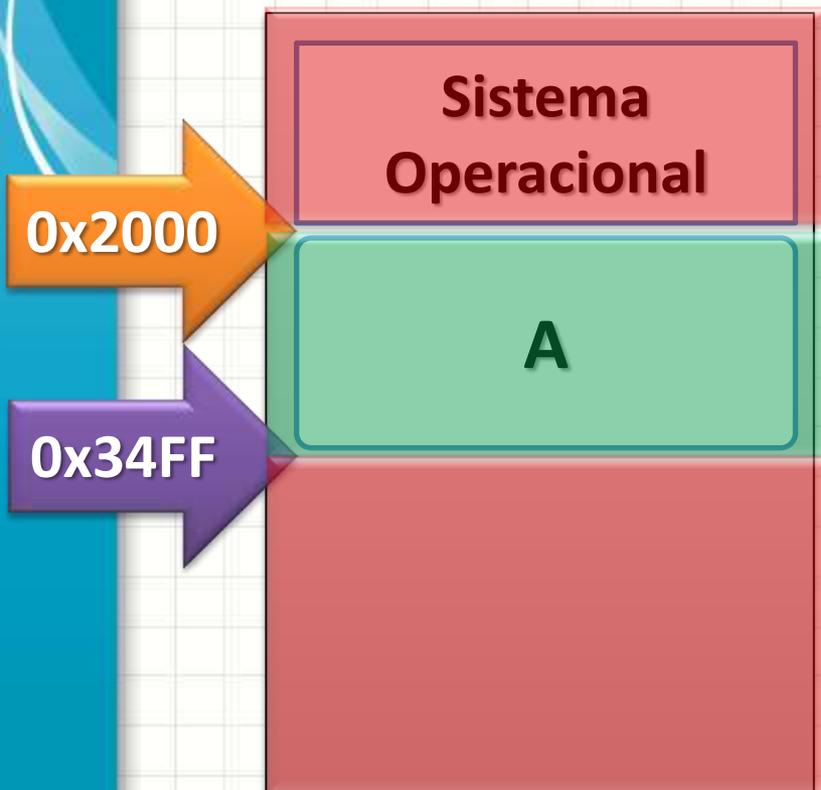
- ...
- Que transfere o controle para o S.O.



MMUs MODERNAS

Gerenciamento de Memória

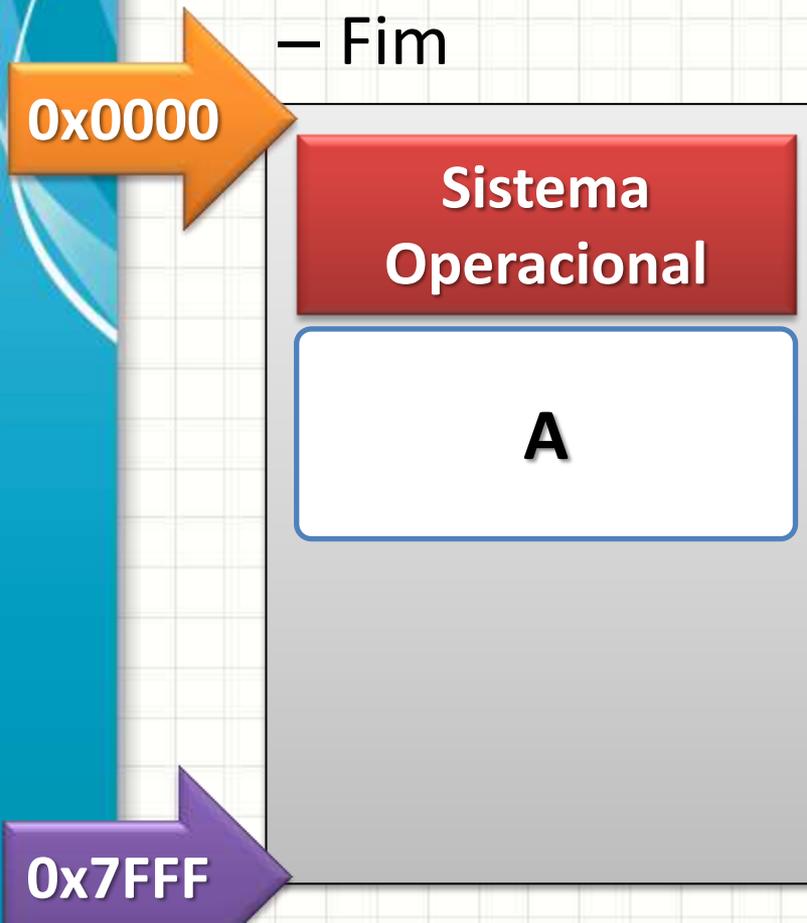
- MMUs Modernas: Dois Registradores
 - Início
 - Fim



- Dividem a memória em dois tipos de região:
- Acessível (entre os dois marcadores)
- Não Acessível (fora dos marcadores)

Gerenciamento de Memória

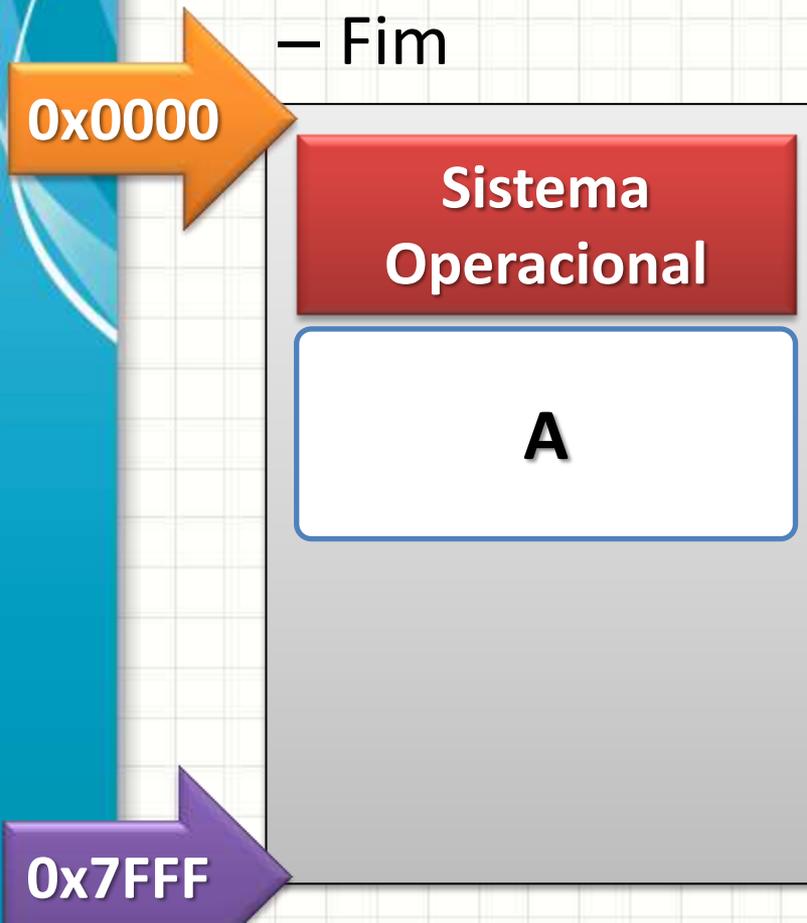
- MMUs Modernas: Dois Registradores
 - Início
 - Fim



- O sistema operacional tem acesso a tudo...

Gerenciamento de Memória

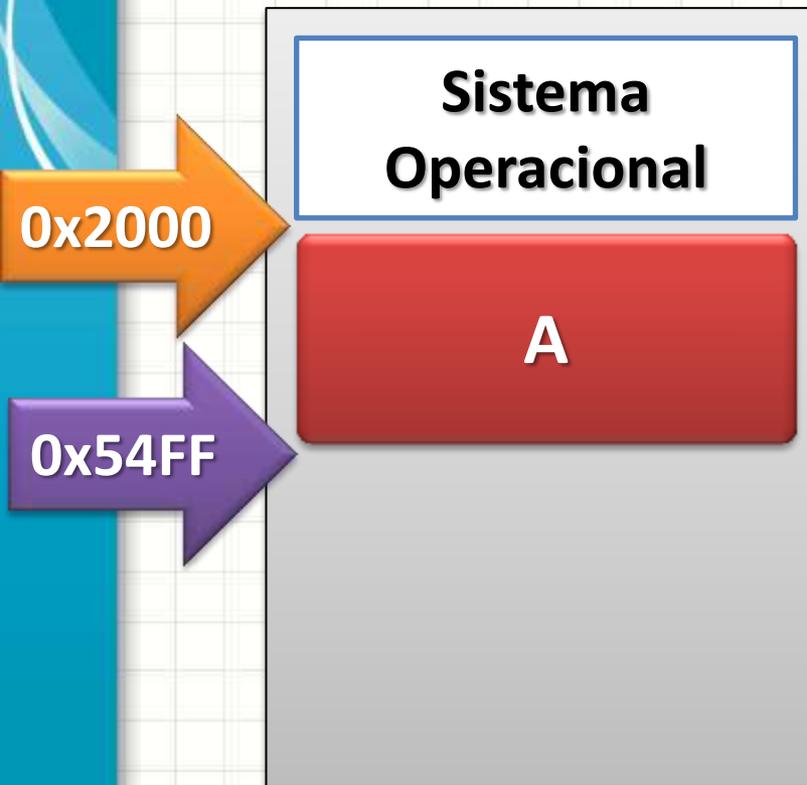
- MMUs Modernas: Dois Registradores
 - Início
 - Fim



- Quando um processo vai entrar em execução...

Gerenciamento de Memória

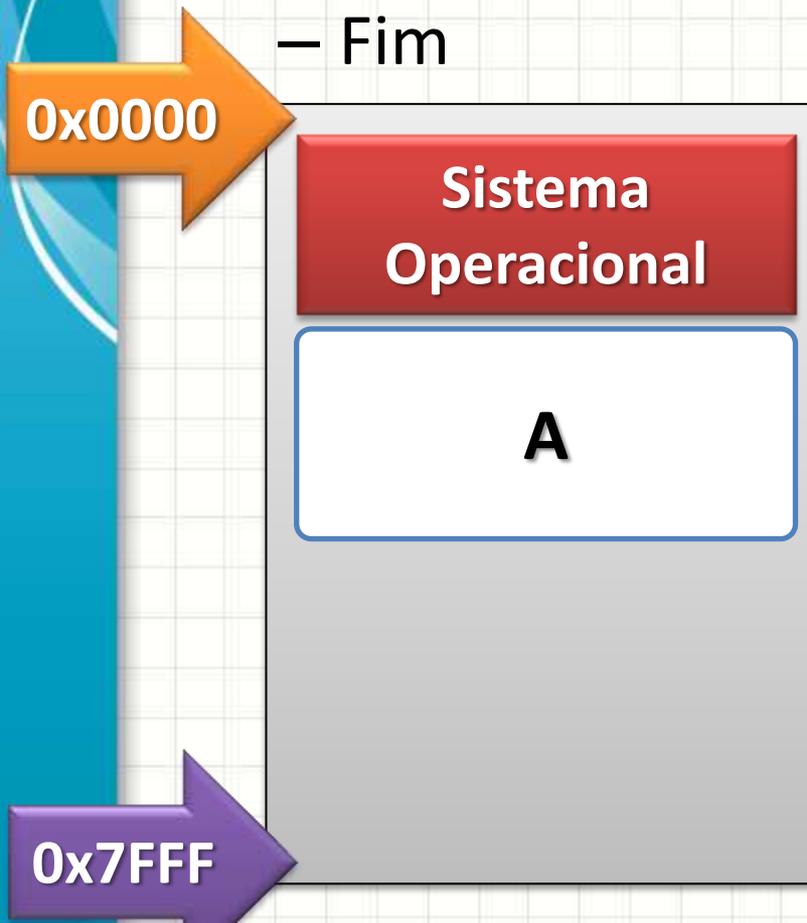
- MMUs Modernas: Dois Registradores
 - Início
 - Fim



- Quando um processo vai entrar em execução...
- A MMU é reconfigurada!
- Quando acaba a “fatia de tempo” do processo...

Gerenciamento de Memória

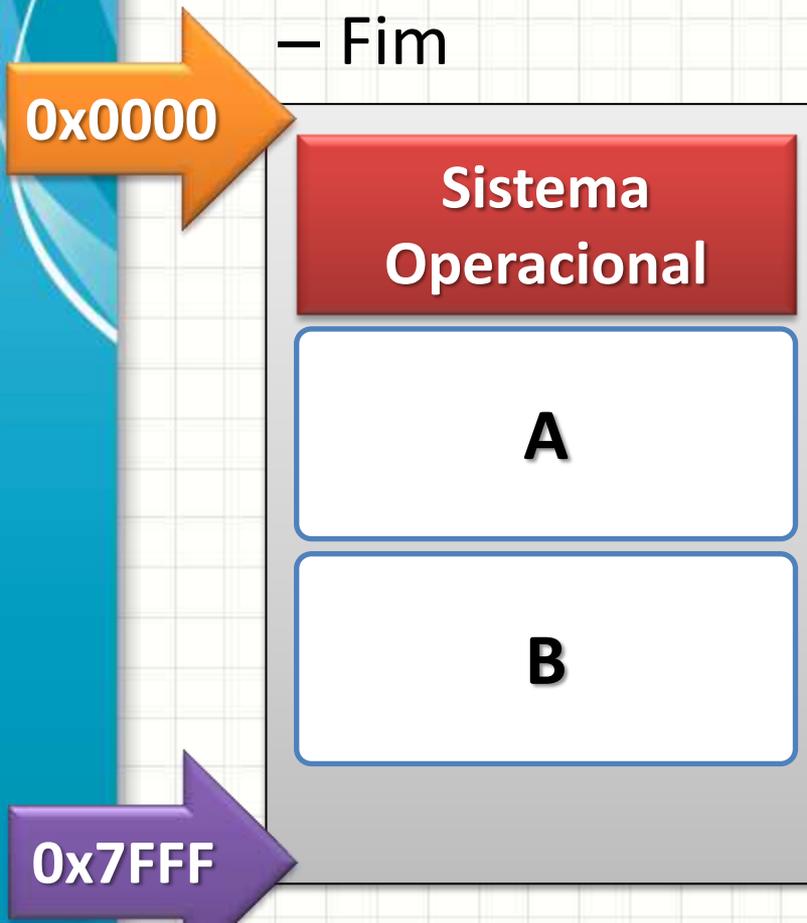
- MMUs Modernas: Dois Registradores
 - Início
 - Fim



- O controle volta ao SO...
- Com a respectiva reconfiguração da MMU!

Gerenciamento de Memória

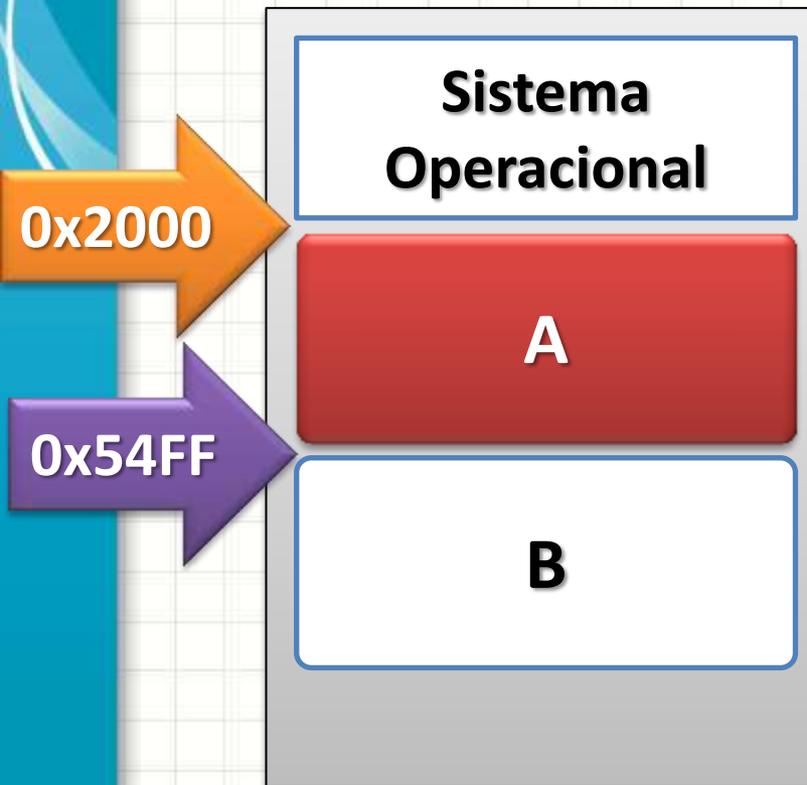
- MMUs Modernas: Dois Registradores
 - Início
 - Fim



- Vantagem...
- Podemos ter mais processos protegidos entre si

Gerenciamento de Memória

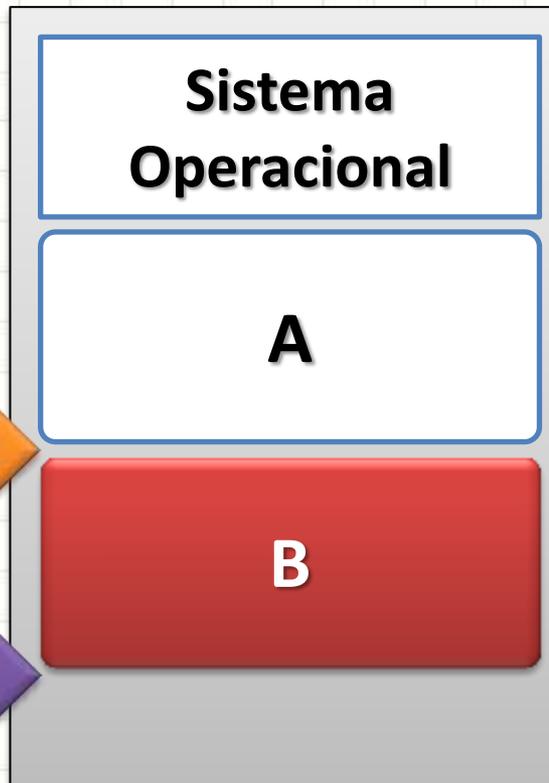
- MMUs Modernas: Dois Registradores
 - Início
 - Fim



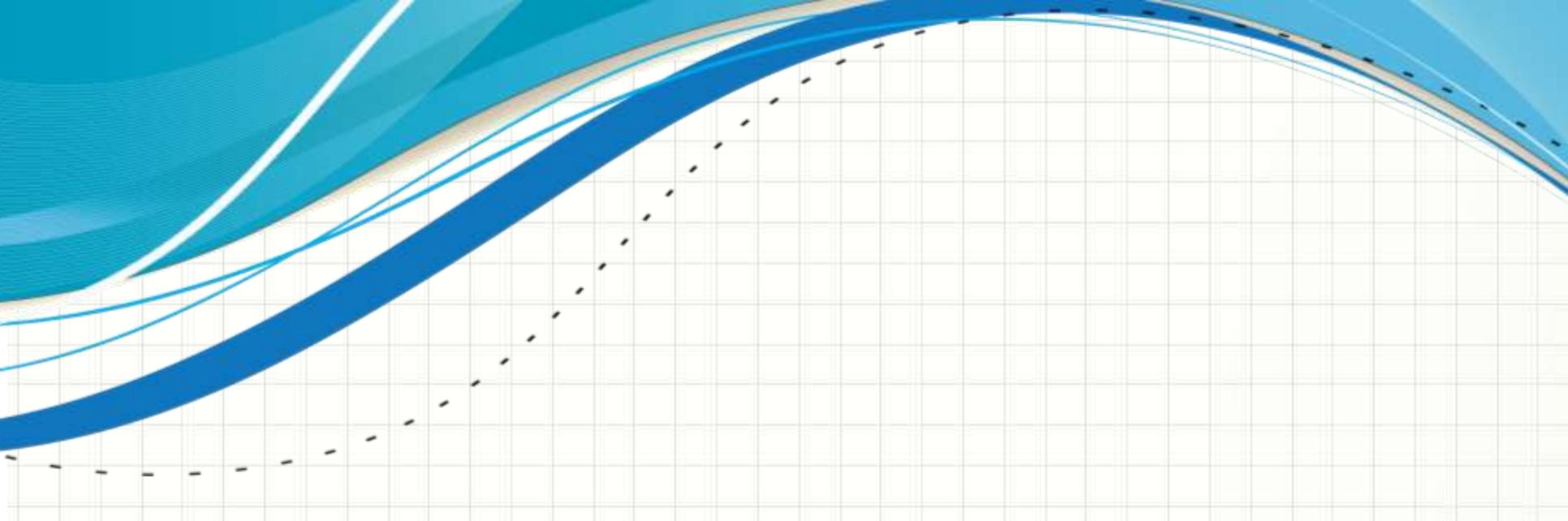
- Vantagem...
- Podemos ter mais processos protegidos entre si

Gerenciamento de Memória

- MMUs Modernas: Dois Registradores
 - Início
 - Fim



- Vantagem...
- Podemos ter mais processos protegidos entre si
- Analogia: república de estudantes com uma governanta



ALOCAÇÃO DE MEMÓRIA

Gerenciamento de Memória

- O Gerenciador também cuida da **alocação...**

**Sistema
Operacional**

A (2MB)

B (3MB)

C (1MB)

D (3MB)

Gerenciamento de Memória

- O Gerenciador também cuida da **alocação...**
- E da **desalocação...**

**Sistema
Operacional**

A (2MB)

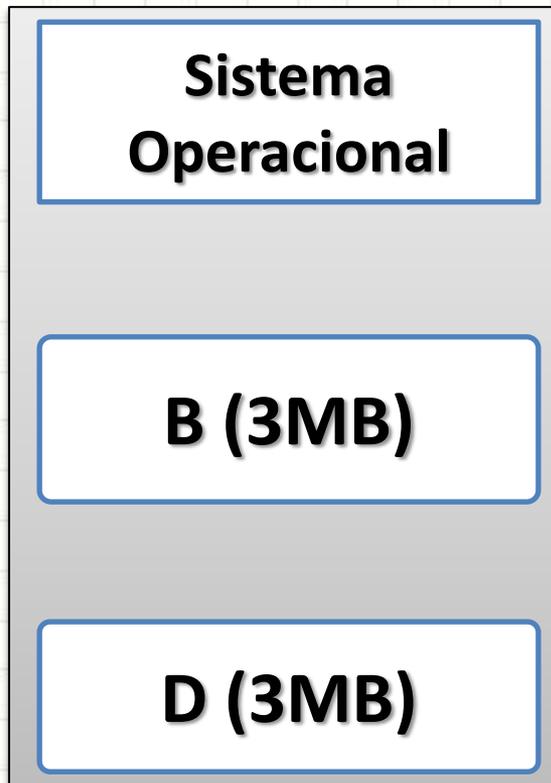
B (3MB)

C (1MB)

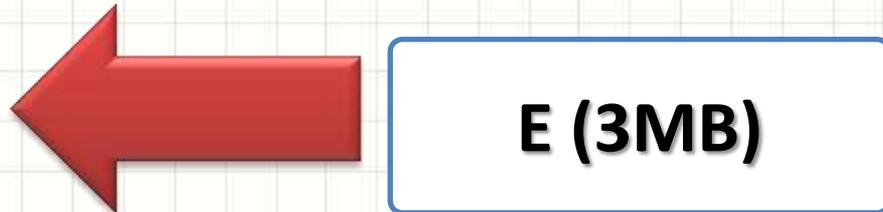
D (3MB)

Gerenciamento de Memória

- O Gerenciador também cuida da **alocação...**
- E da **desalocação...**

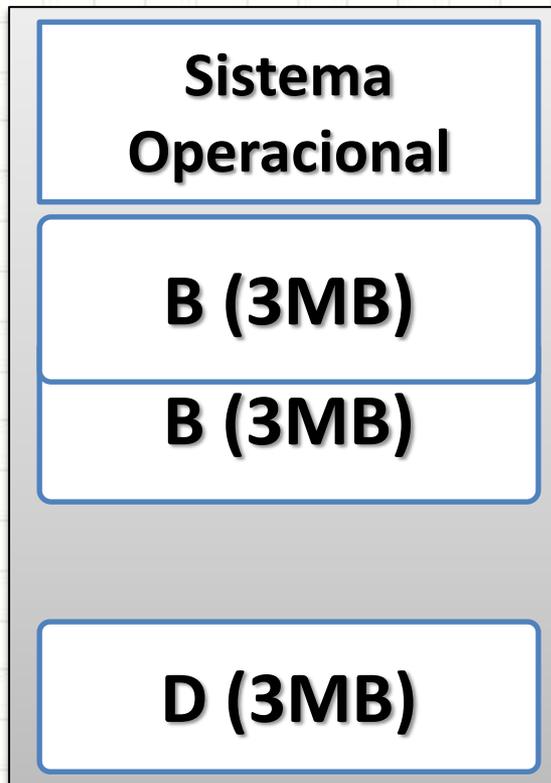


- Como carregar o processo E?

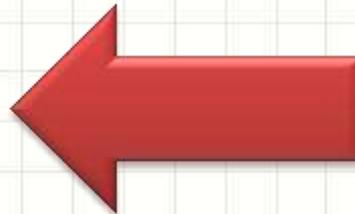


Gerenciamento de Memória

- O Gerenciador também cuida da **alocação...**
- E da **desalocação...**

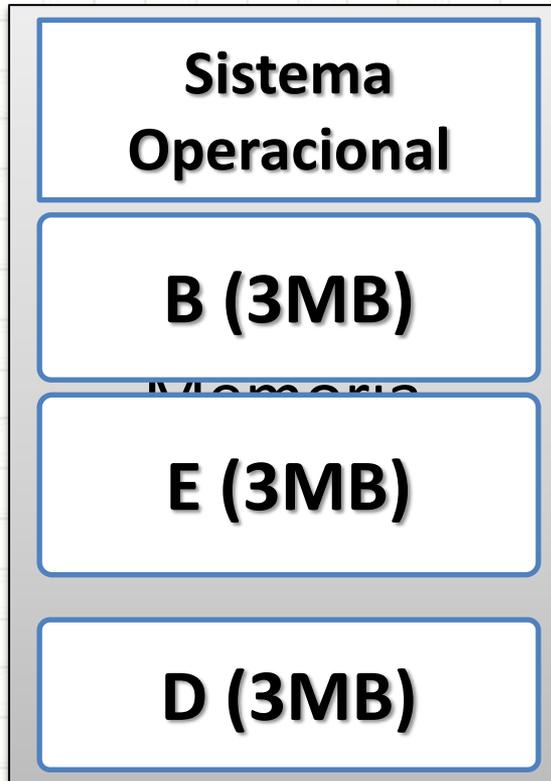


- Como carregar o processo E?
- Mover B?



Gerenciamento de Memória

- O Gerenciador também cuida da **alocação...**
- E da **desalocação...**



- Como carregar o processo E?
- Mover B?

Gerenciamento de Memória

- O Gerenciador também cuida da **alocação...**

• E da **desalocação...**

Programa B antes de ser movido

0x4000	LD A,(0x400B)
0x4003	LD B,(0x400C)
0x4006	ADD A,B
0x4007	LD A,B
0x4008	LD (0x400D),A
0x400B	DB 0x01
0x400C	DB 0x02
0x400D	DB 0x00

Programa B depois de ser movido

0x2000	LD A,(0x400B)
0x2003	LD B,(0x400C)
0x2006	ADD A,B
0x2007	LD A,B
0x2008	LD (0x400D),A
0x200B	DB 0x01
0x200C	DB 0x02
0x200D	DB 0x00



Segmentação de Memória

- O Gerenciamento de Memória não funciona...

Não funciona!

Programa B

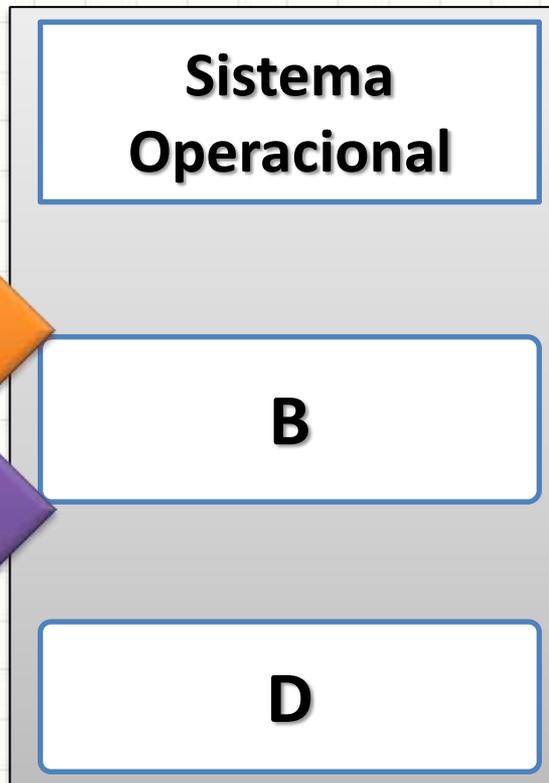
0x4000	LD	(0x400B)
0x4003	LD B,(0x400C)	
0x4006	ADD A,B	
0x4007	LD A,B	
0x4008	LD (0x400D),A	
0x400B	DB 0x01	
0x400C	DB 0x02	
0x400D	DB 0x00	

Programa A

0x2000	LD B,(0x400C)
0x2006	ADD A,B
0x2007	LD A,B
0x2008	LD (0x400D),A
0x200B	DB 0x01
0x200C	DB 0x02
0x200D	DB 0x00

Gerenciamento de Memória

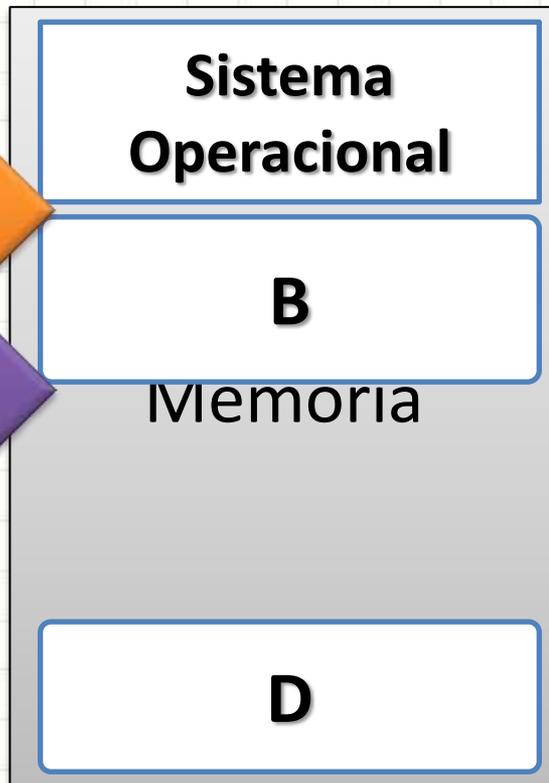
- Para resolver esse problema... usar a MMU!
- Truque: fazer o programa pensar que sempre roda a partir do endereço ZERO



MMU1	0x4000
MMU2	0x45FF
Real	0x4000
Virtual	0x0000

Gerenciamento de Memória

- Para resolver esse problema... usar a MMU!
- Truque: fazer o programa pensar que sempre roda a partir do endereço ZERO



MMU1	0x2000
MMU2	0x25FF
Real	0x2000
Virtual	0x0000

Gerenciamento de Memória

- Para resolver esse problema... usar a MMU!

• **MMU1: 0x4000**

Programa B antes de **+** ser movido

0x4000	LD A,(0x000B)
0x4003	LD B,(0x000C)
0x4006	ADD A,B
0x4007	LD A,B
0x4008	LD (0x000D),A
0x400B	DB 0x01
0x400C	DB 0x02
0x400D	DB 0x00

• **MMU1: 0x2000**

Programa B depois de **+** ser movido

0x2000	LD A,(0x000B)
0x2003	LD B,(0x000C)
0x2006	ADD A,B
0x2007	LD A,B
0x2008	LD (0x000D),A
0x200B	DB 0x01
0x200C	DB 0x02
0x200D	DB 0x00

Gerenciamento de Memória

- Para resolver esse problema... usar a MMU!

MMU1: 0x4000

MMU1: 0x2000

Programa B antes de ser movido

Programa B depois de ser movido

0x4000 LD A,(0x0000)

0x4003

0x4006

0x4007

0x4008 LD A,(0x000D),A

0x400B DB 0x01

0x400C DB 0x02

0x400D DB 0x00

0x2000

0x2003

0x2006

0x2007

0x2008

0x200B DB 0x01

0x200C DB 0x02

0x200D DB 0x00

**Endereçamento
Virtual**

Gerenciamento de Memória

Para resolver esse problema, vamos usar a MMU!

- Memória virtual de 400 páginas sempre

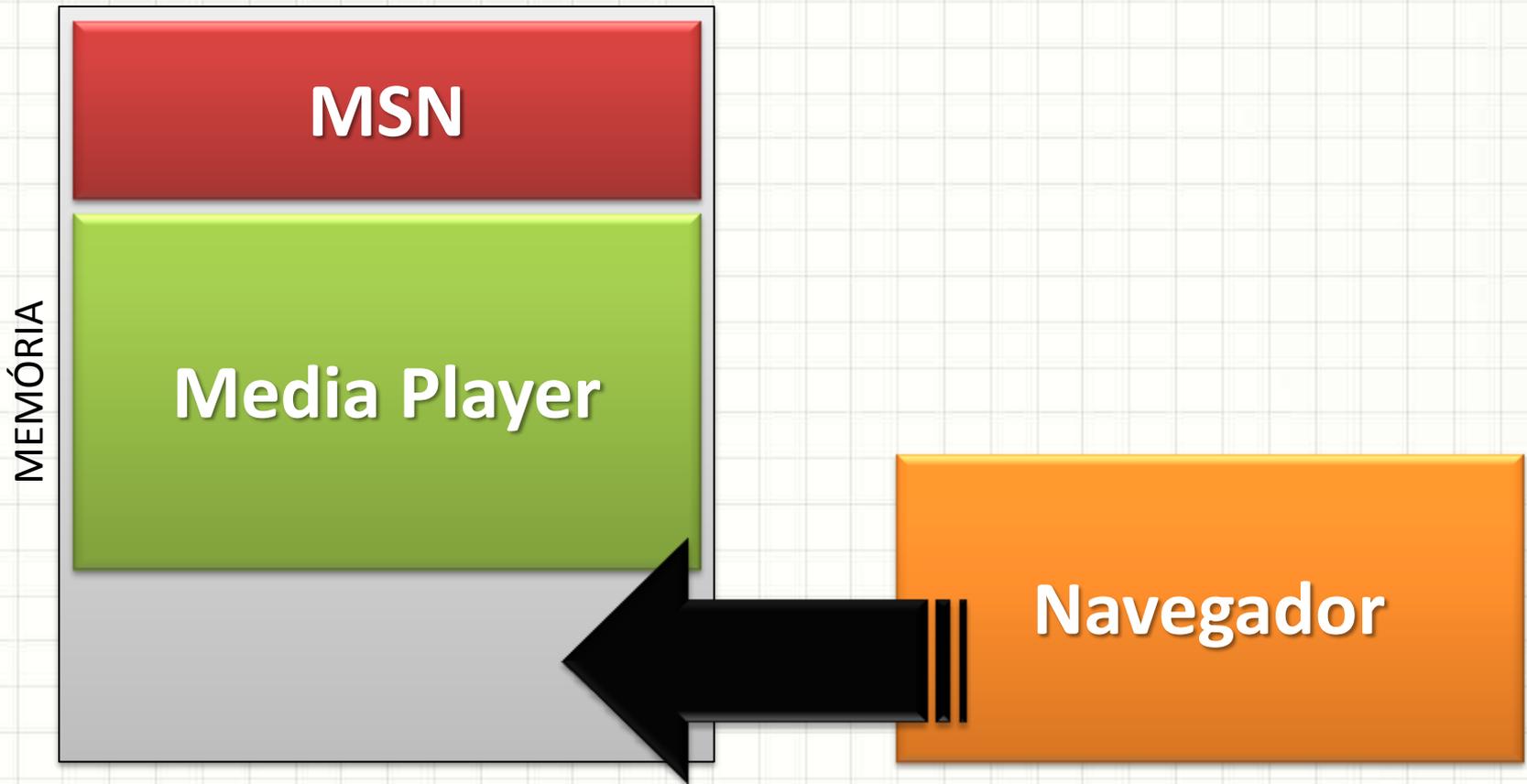
Programa B sempre ser movido

0x4000		
0x4001		(JUC)
0x4002		B
0x4007		
0x4008		(x000D),A
0x400B	DE	01
0x400C	D	02
0x400D	B 0x00	DB 0x00

Os valores da MMU de um processo ficam na PCB (estado)

Gerenciamento de Memória

- Muitos Processos x Pouca Memória
 - O que fazer quando não cabe?



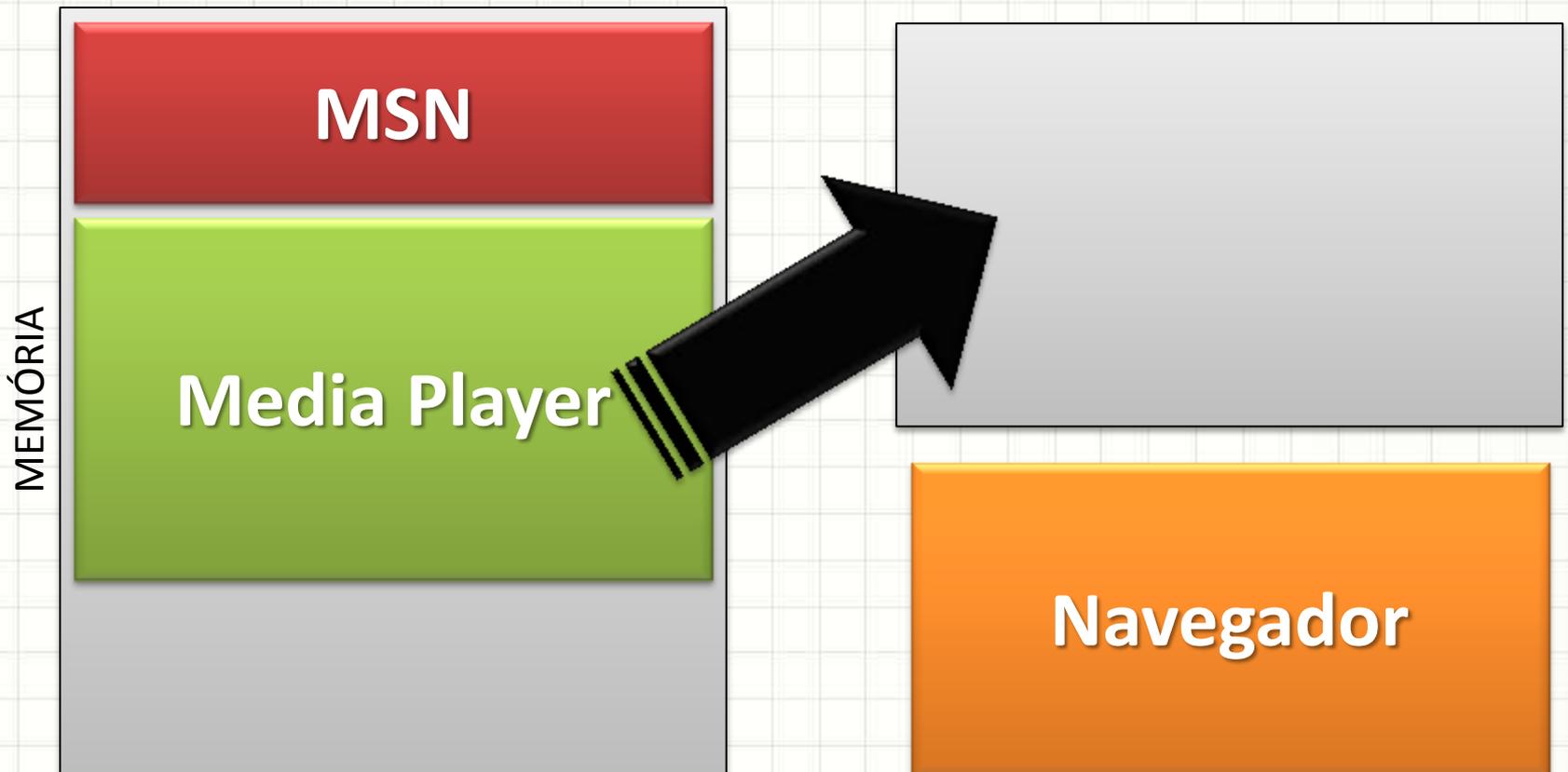
Gerenciamento de Memória

- Muitos Processos x Pouca Memória
 - Quando não cabe: Move programa menos usado



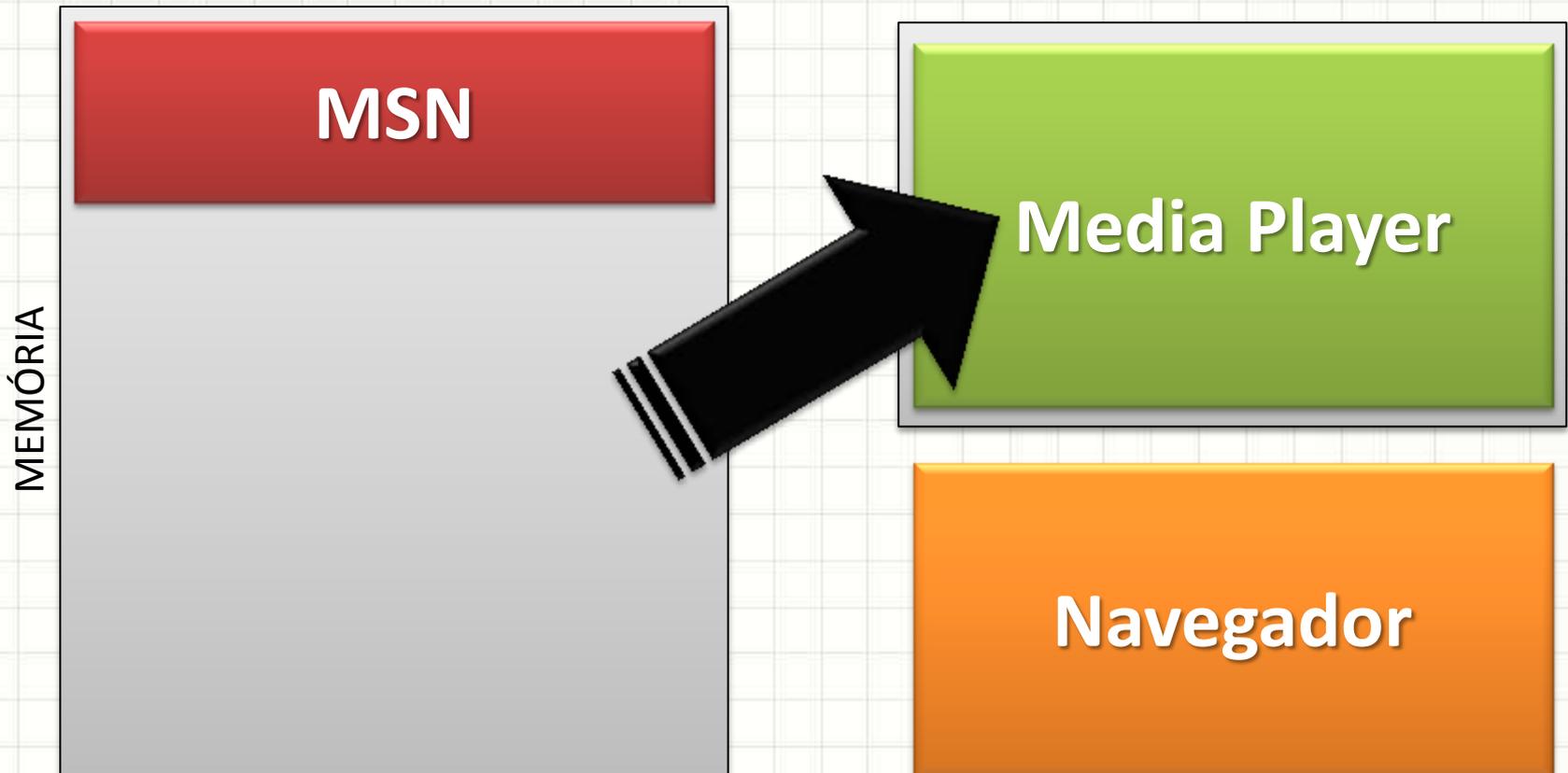
Gerenciamento de Memória

- Muitos Processos x Pouca Memória
 - Quando não cabe: Move programa menos usado



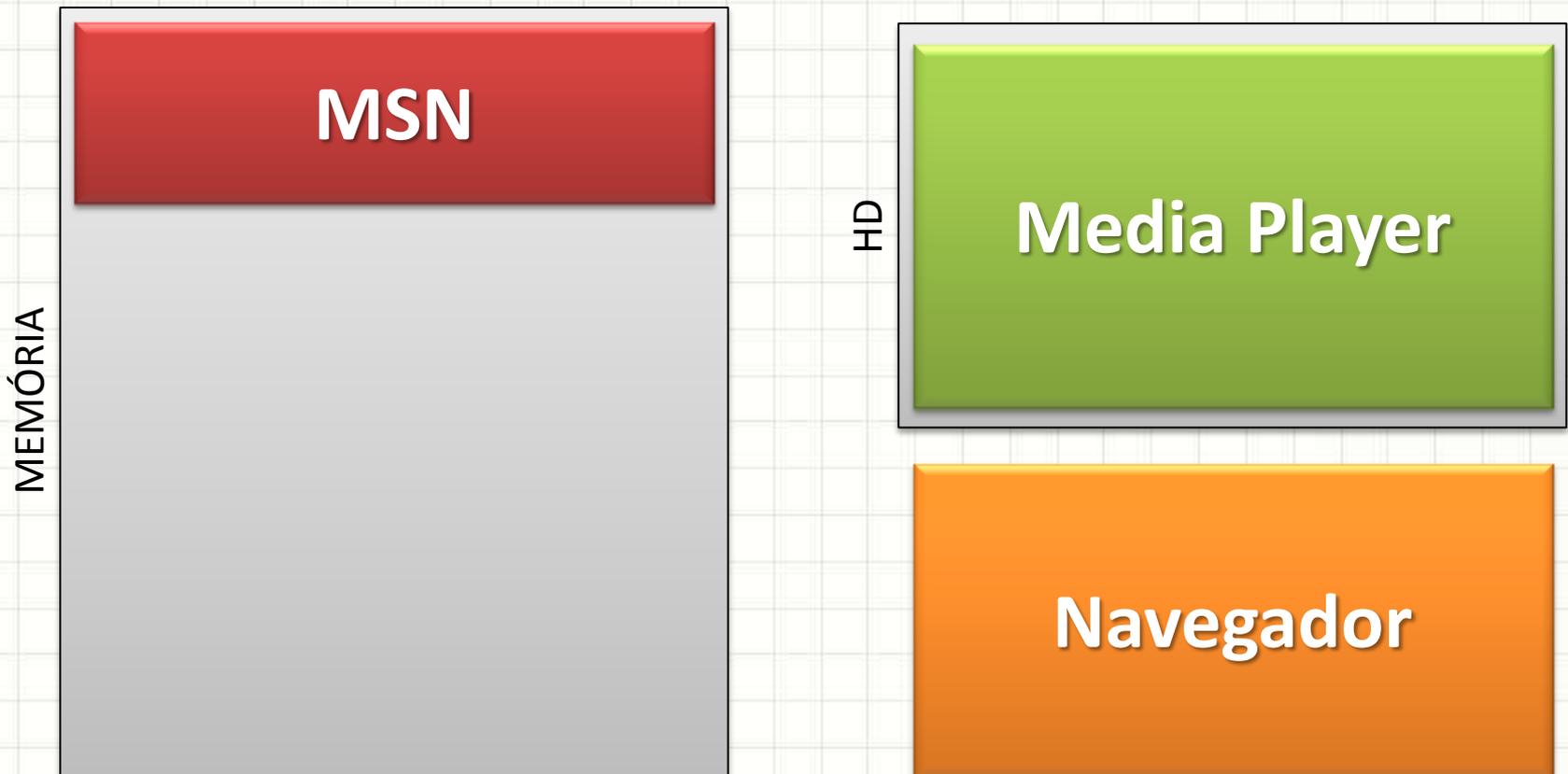
Gerenciamento de Memória

- Muitos Processos x Pouca Memória
 - Quando não cabe: Move programa menos usado



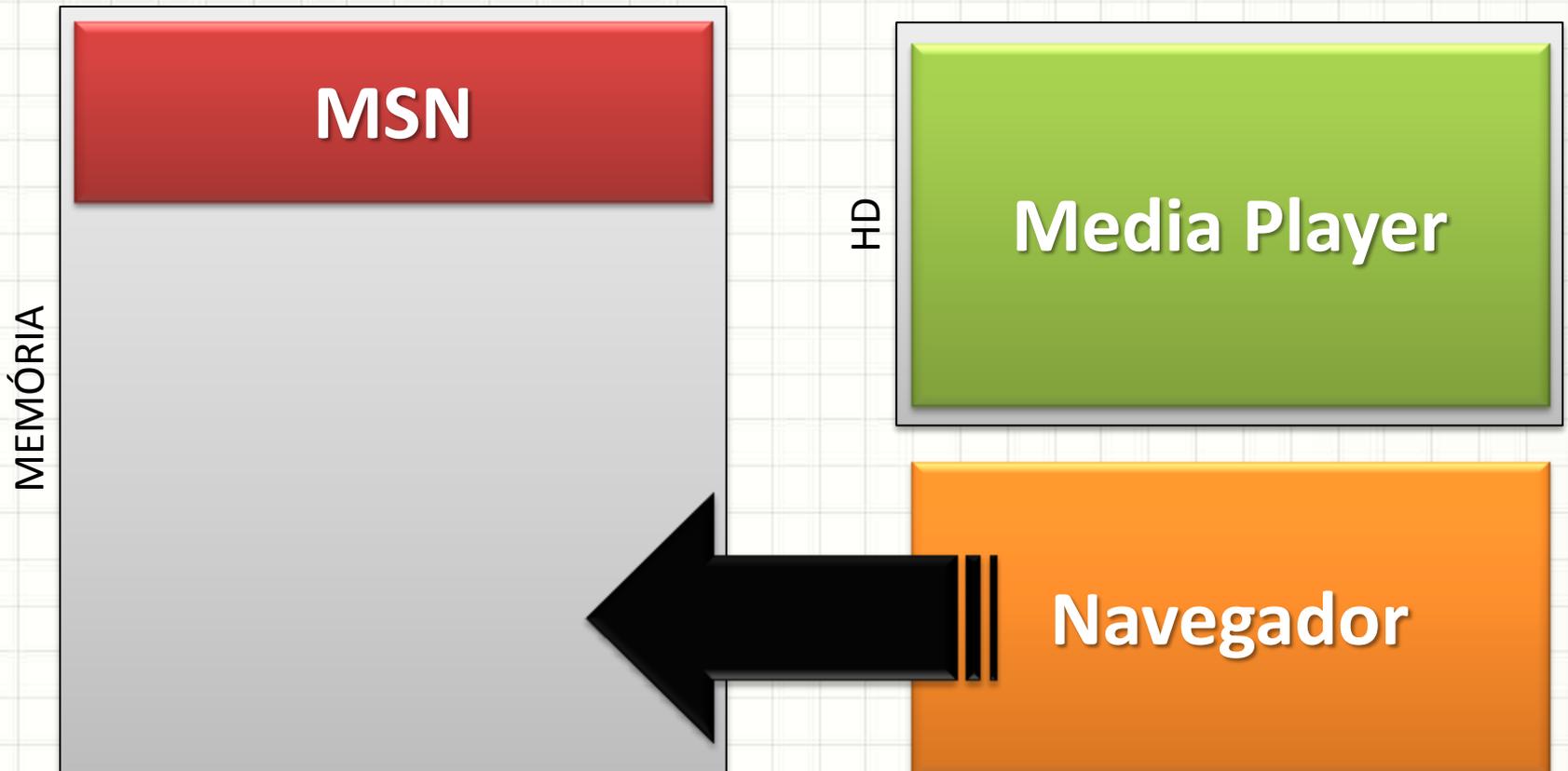
Gerenciamento de Memória

- Muitos Processos x Pouca Memória
 - Quando não cabe: Move programa menos usado



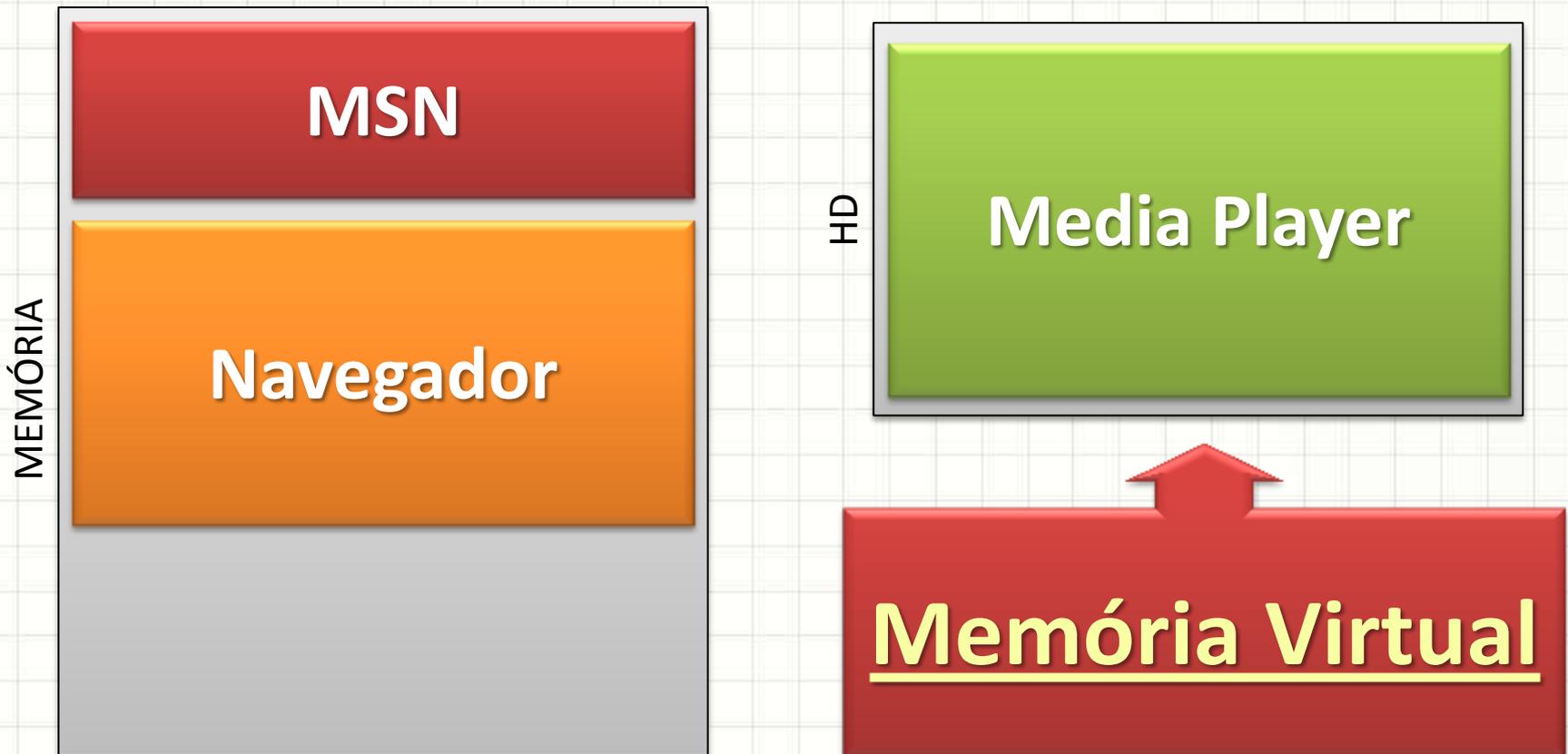
Gerenciamento de Memória

- Muitos Processos x Pouca Memória
 - Quando não cabe: Move programa menos usado



Gerenciamento de Memória

- Muitos Processos x Pouca Memória
 - Quando não cabe: Move programa menos usado



Gerenciamento de Memória

os Processos

usado

**E quando o Media
Player for rodar, como o
S.O. sabe que tem que
trazê-lo para a RAM?**

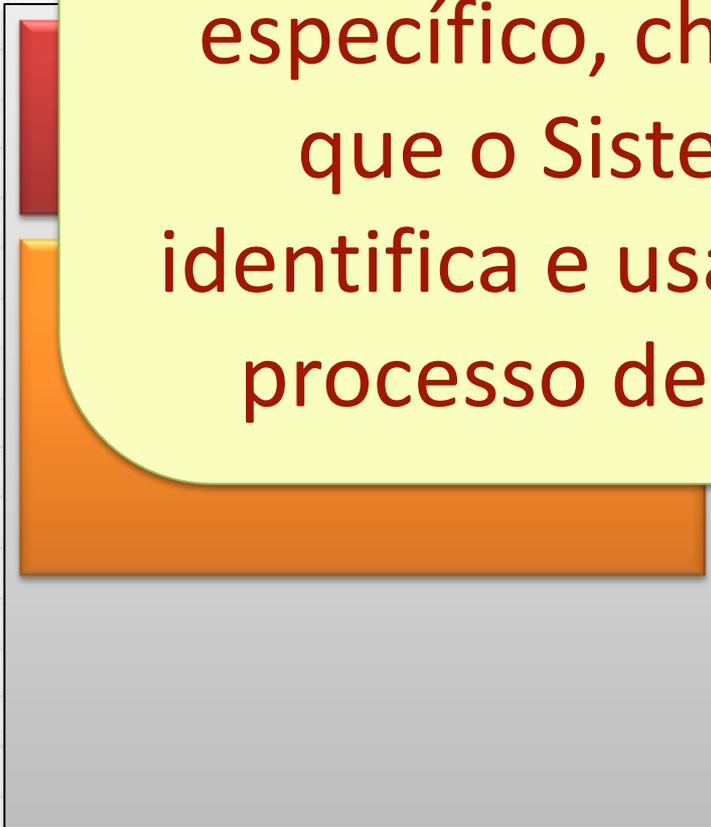
MEN

Memória Virtual

Ger

Quando a CPU tenta acessar um processo que está em um endereço **fora** da RAM, ocorre um “erro” específico, chamado **page fault**, que o Sistema Operacional identifica e usa para trazer aquele processo de volta para a RAM

MEMÓRIA



Memória Virtual



Gerenciamento de Memória

Assim, se o micro tem 2GB de RAM e o registro MMU1 indica um endereço inicial MAIOR que 2GB, ocorre um page fault!

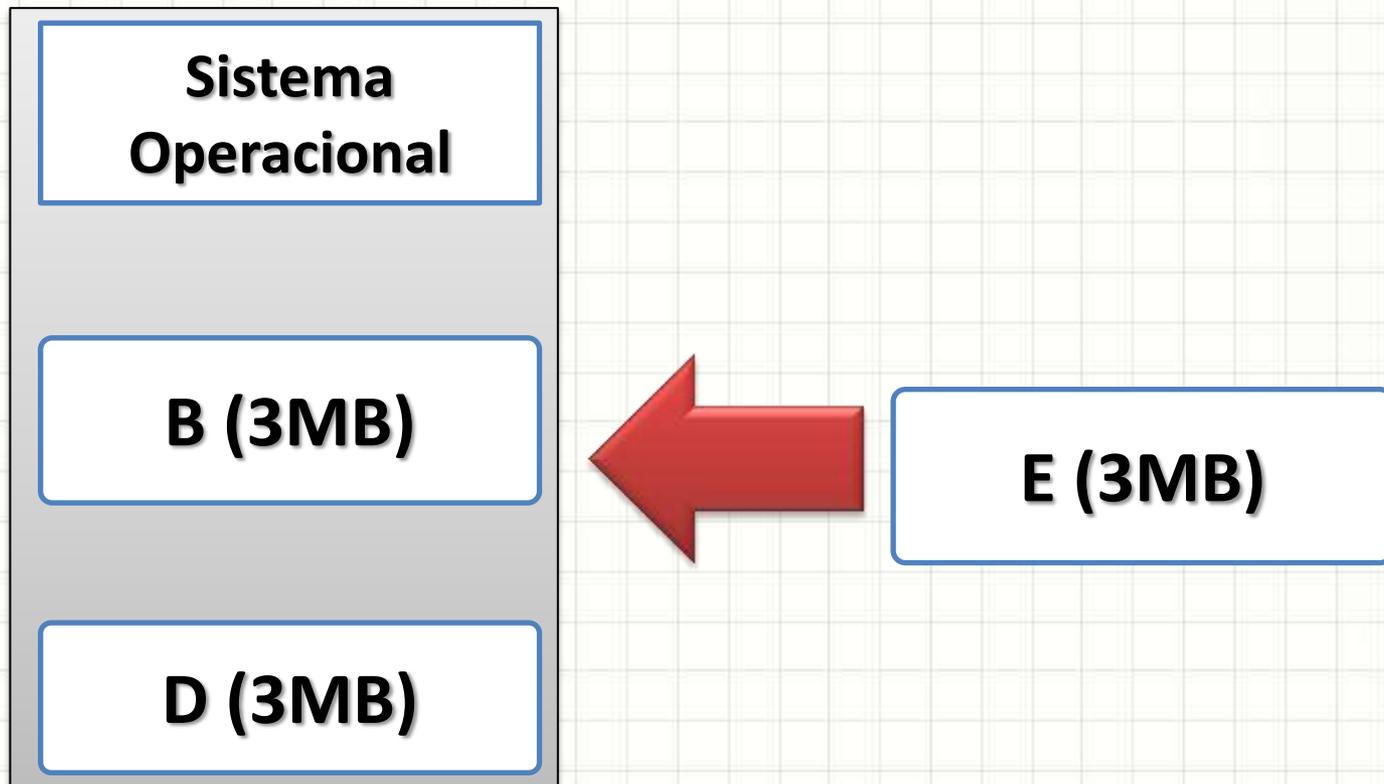
usado

MEN

Memória Virtual

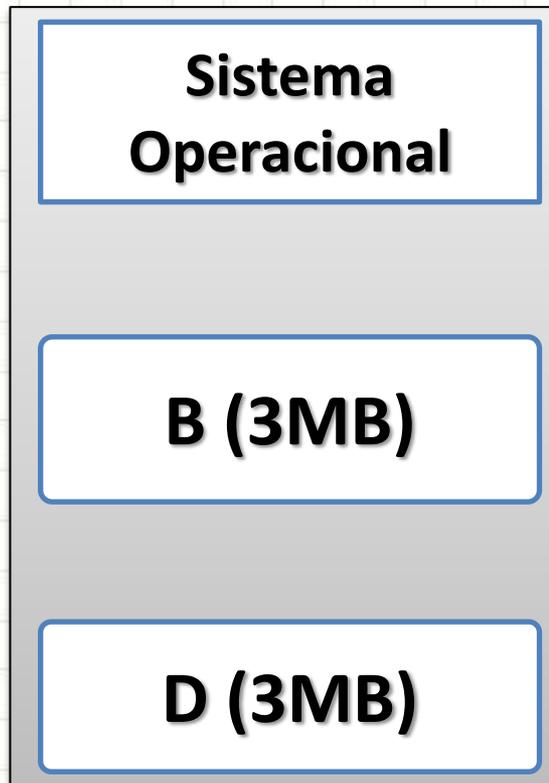
Gerenciamento de Memória

- Voltemos à situação original...
- Mover o processo B é a única solução?



Gerenciamento de Memória

- Voltemos à situação original...
- Mover o processo B é a única solução?

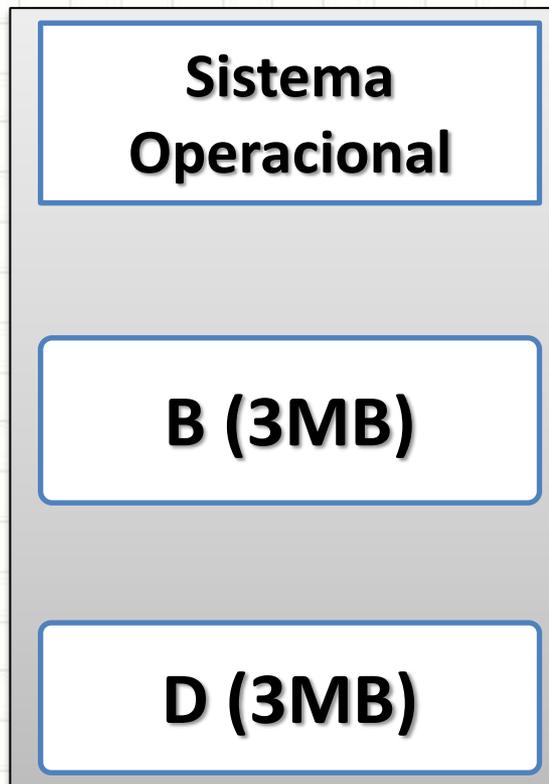


- Poderíamos “quebrar” **E** em duas partes...

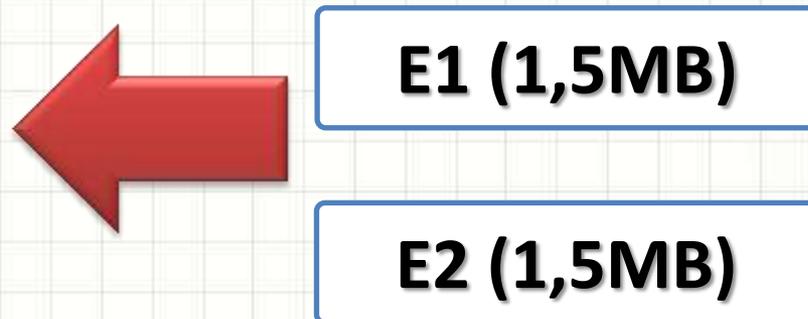


Gerenciamento de Memória

- Voltemos à situação original...
- Mover o processo B é a única solução?

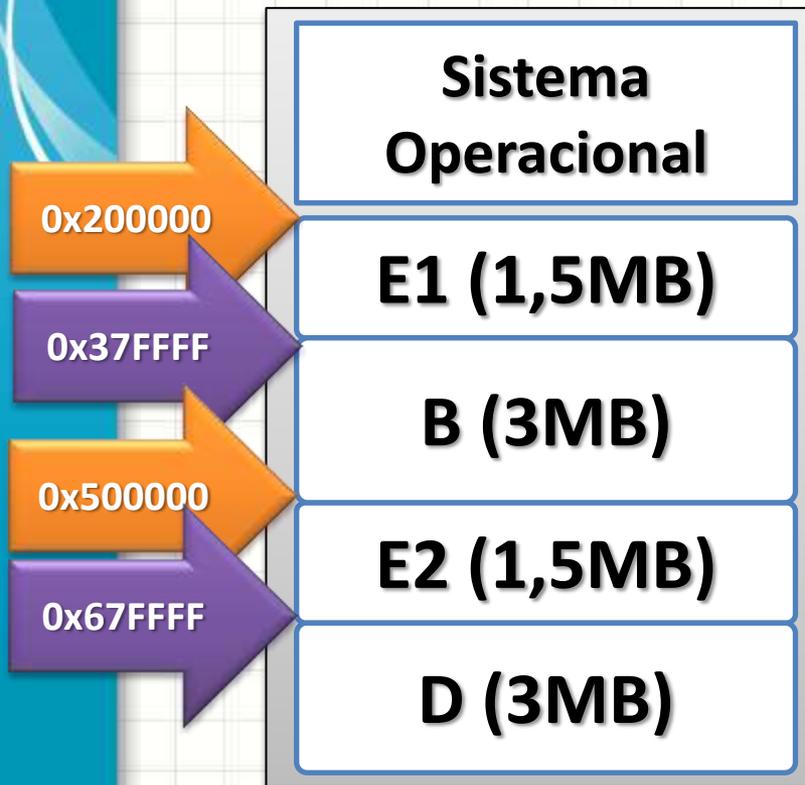


- Poderíamos “quebrar” **E** em duas partes...



Gerenciamento de Memória

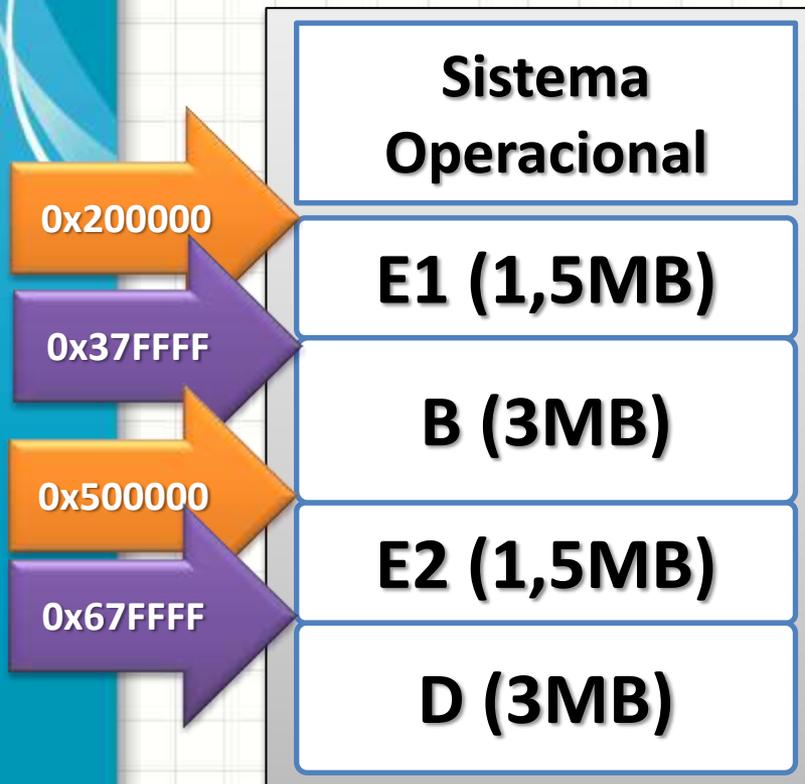
- Voltemos à situação original...
- Mover o processo B é a única solução?



- Poderíamos “quebrar” **E** em duas partes...
- Mas, para isso funcionar, precisamos de múltiplos registros de MMU...

Gerenciamento de Memória

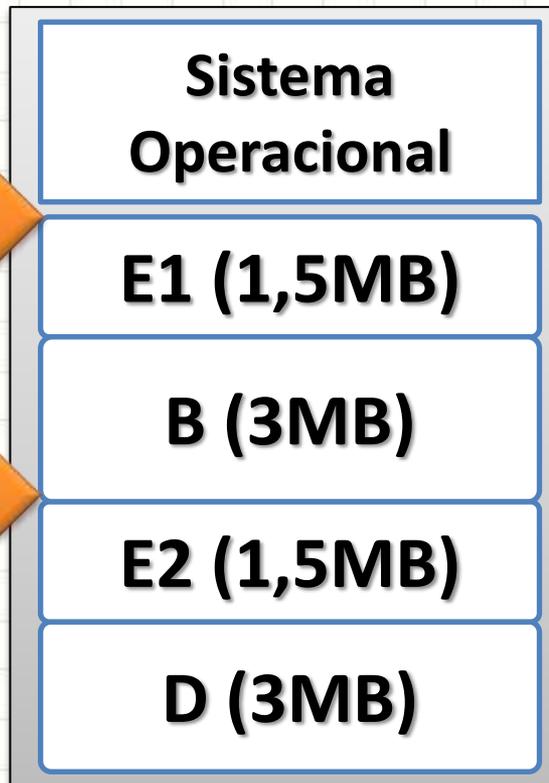
- Voltemos à situação original...
- Mover o processo B é a única solução?



- O processador faz a mágica para que o processo pense que está em um único bloco
- Como diminuir o número de registros de MMU?

Gerenciamento de Memória

- Voltemos à situação original...
- Mover o processo B é a única solução?



- Padronizando o tamanho do bloco... por exemplo, em 1,5MB
- Só são necessários registros de início de área
- Cada bloco: **página**

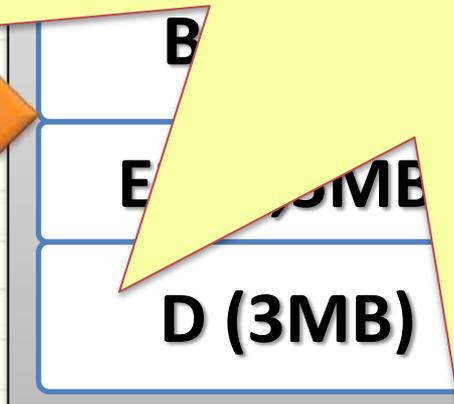
Gerenciamento de Memória

- Memórias à
- Memórias

O menor processo possível ocupa uma página. 4KB é um valor usual para uma página

0x200000

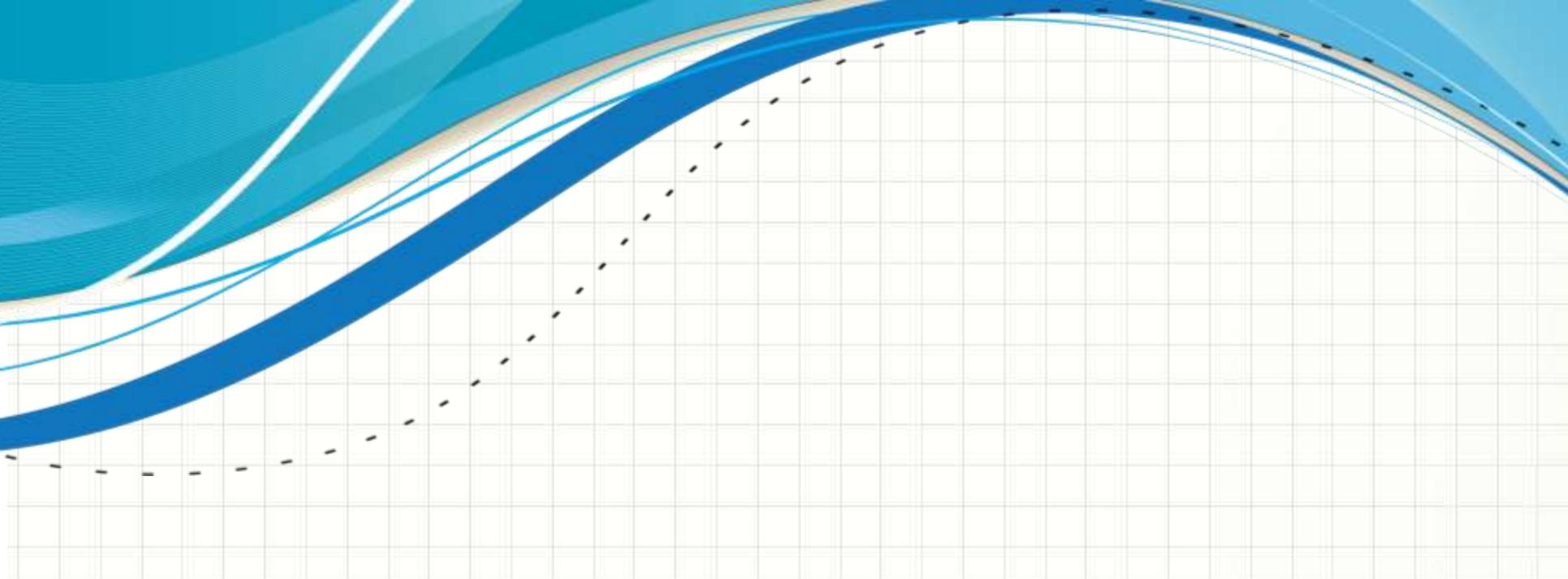
0x500000



- Cada bloco: página

Demonstração

- Gerenciamento de Memória e Processos?
- SO Sim (Simulador de Sistema Operacional)



CONCLUSÕES

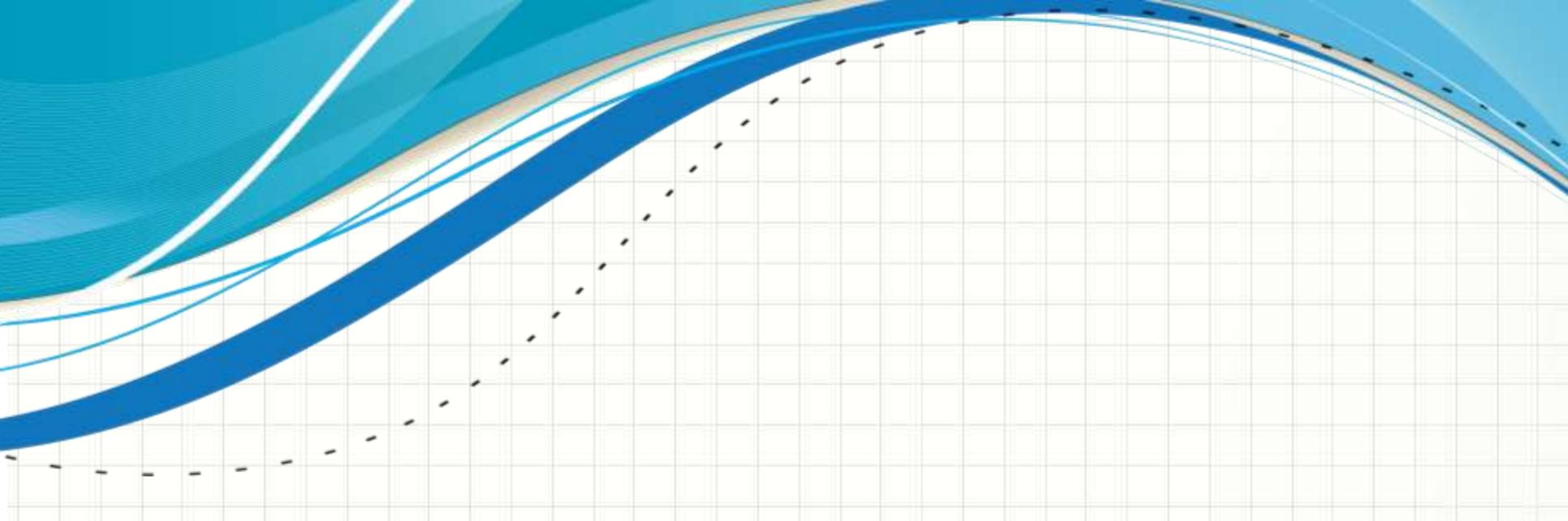
Resumo

- O S.O. tem funções extremamente complexas e importantes
- Muitas das funcionalidades dos S.O.s dependem de auxílio do hardware, como interrupções e MMUs
- Em sistemas que o possuem, o disco pode ser usado como memória virtual
- **TAREFA**
 - Lista de Exercícios 2!

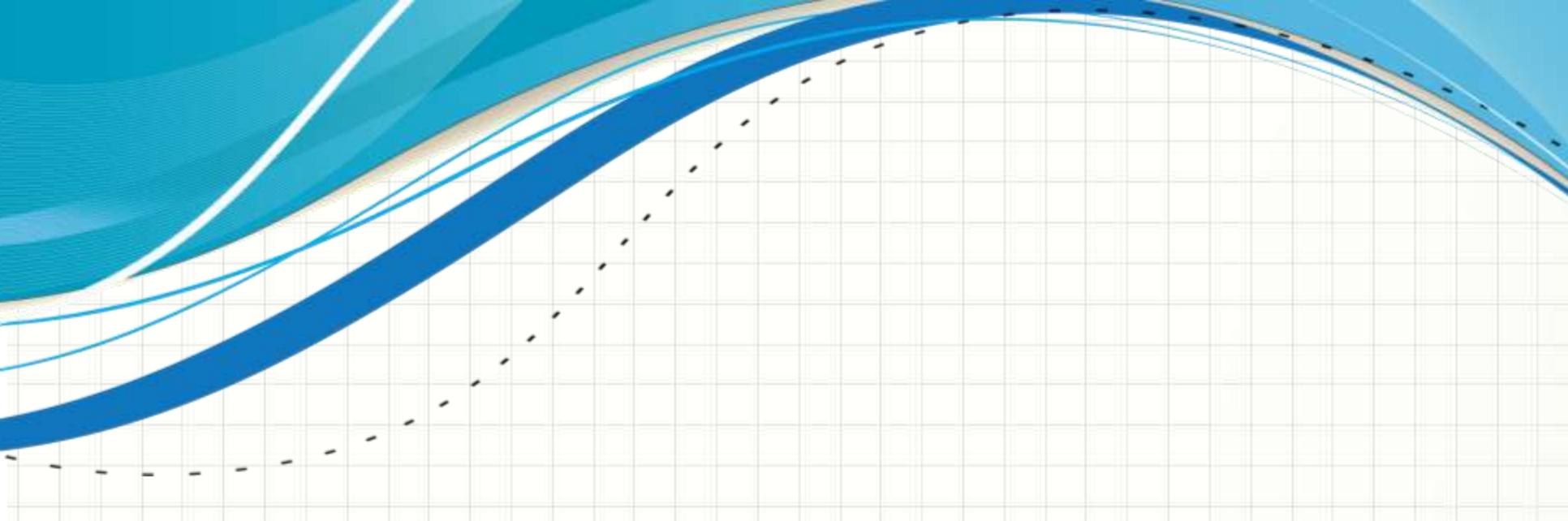
Próxima Aula



- Ok... mas o S.O. é um programa...
- Quem o executa é a CPU!
 - Como é a CPU?
 - O que ela faz?



PERGUNTAS?



**BOM DESCANSO
A TODOS!**