

# **ARQUITETURA E ORGANIZAÇÃO DE COMPUTADORES**

## **INTRODUÇÃO AOS SISTEMAS OPERACIONAIS**

Prof. Dr. Daniel Caetano

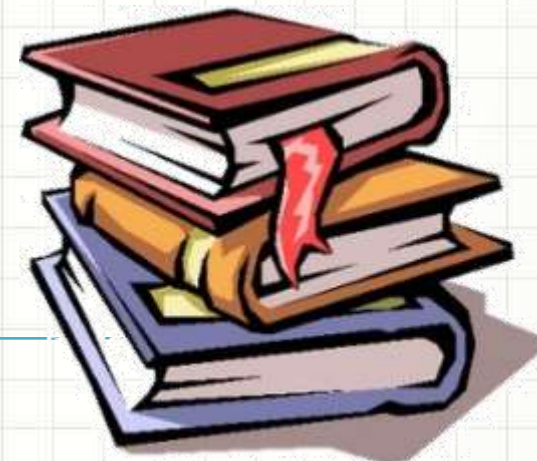
2012 - 2

# Objetivos

- Compreender o que é um Sistema Operacional
- Conhecer as principais funções de um sistema operacional
- **Lembrete:**
  - Lista 2!



# Material de Estudo



Material	Acesso ao Material
Notas de Aula	<a href="http://www.caetano.eng.br/">http://www.caetano.eng.br/</a> (Aula 9)
Apresentação	<a href="http://www.caetano.eng.br/">http://www.caetano.eng.br/</a> (Aula 9)
Material Didático	Arquitetura e Organização de Computadores, páginas 239 a 285
Biblioteca Virtual	-



# INTRODUÇÃO

# Por que estudar S.O.s?

- Engenheiro eletrônico: faz hardware
- Por que estudar software?
  - Porque todo hardware tem um software
  - Porque todo projetista de hardware entra em contato com o engenheiro de computação
  - Porque só é possível compreender alguns recursos do hardware após compreender o uso destes recursos no software



# Qual a Importância de um S.O.?

- Em princípio, um S.O. é um programa...
- Mas sua função é: permitir acesso...
  - Dos usuários aos programas



# Qual a Importância de um S.O.?

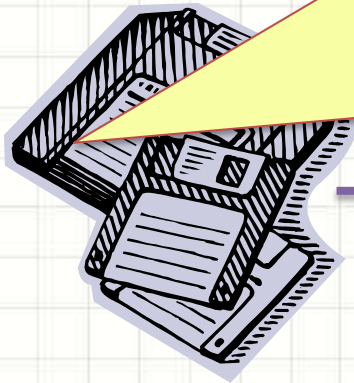
- Em princípio, um S.O. é um programa...
- Mas sua função é: permitir acesso...
  - Dos programas ao hardware



# Qual a Importância de um S.O.?

- Em princípio, um S.O. é um programa...
- Mas sua função é...

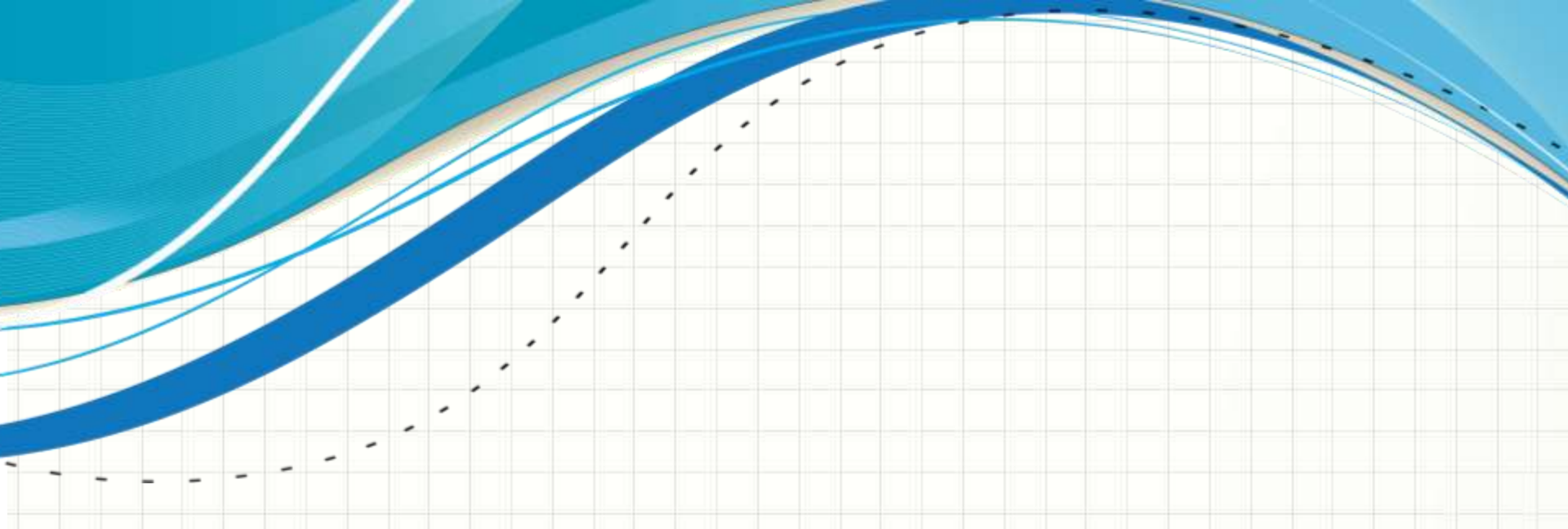
**Torna o  
computador  
ÚTIL!**



ac...

Hardware  
(memória, CPU,  
impressoras,  
Rede...)





# **O CONCEITO DE SISTEMA OPERACIONAL**

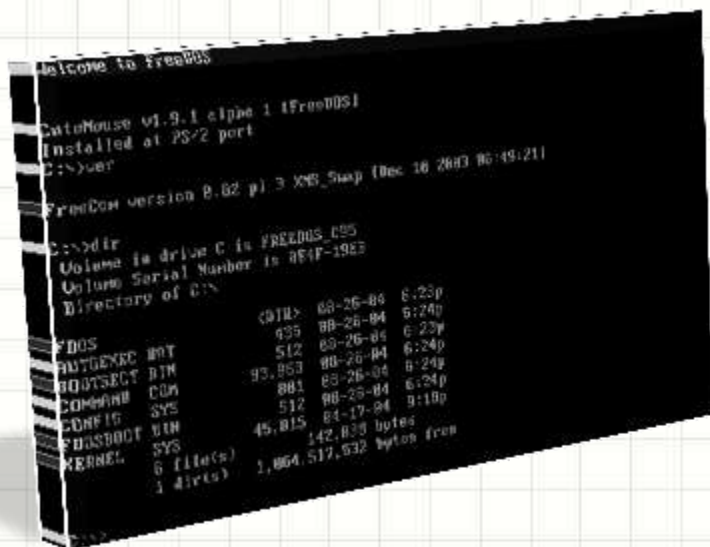
# O Conceito de Sistema Operacional

- Função:
  - Executar ou auxiliar a execução de tarefas básicas
  - Exemplos:
    - Carregar um programa
    - Gerenciar a impressão de um documento
    - Escolher qual programa executar etc.



# O Conceito de Sistema Operacional

- Mas... o que é o S.O.?
  - Conjunto de **rotinas**, em geral de **baixo nível**
- Complexidade variada
  - Carregador de Programas ...
  - Infinitude de Funções





# **S.O. E A PADRONIZAÇÃO DO ACESSO**

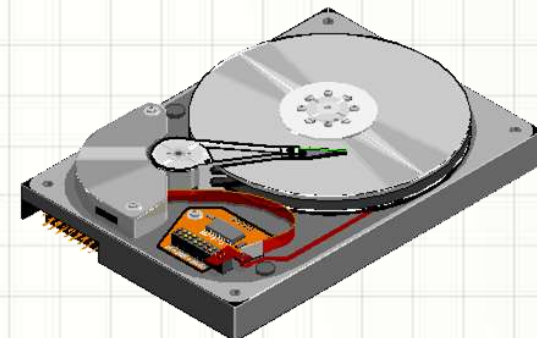
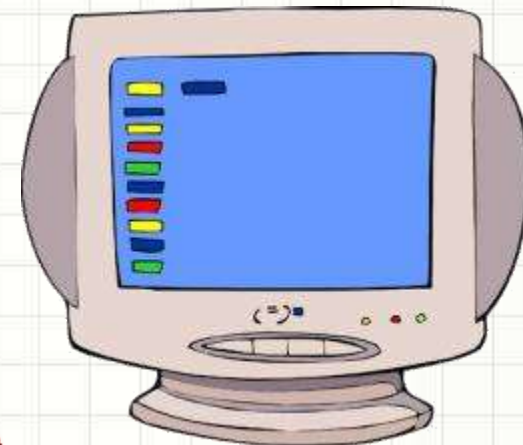


# O Conceito de Sistema Operacional

- Padronização do Acesso a recursos
  - Subsistemas e Drivers



`print(data)`  
`cout << data`





# Padronização de Acesso

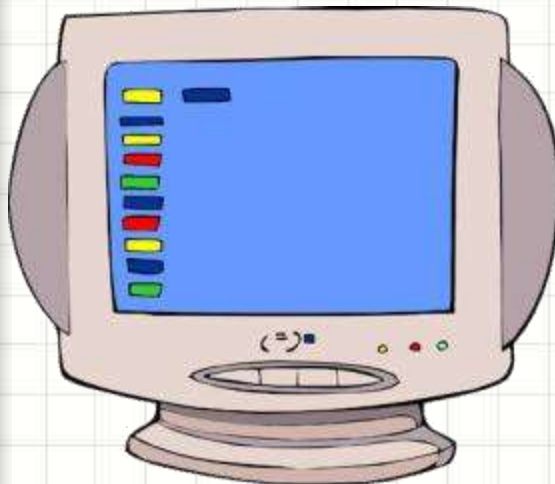
- Como acessar recursos diferentes de maneira igual?



`cout << dado`



`write(dado)`



`out (port1), comando`  
`out (port1), dado`



# Padronização de Acesso

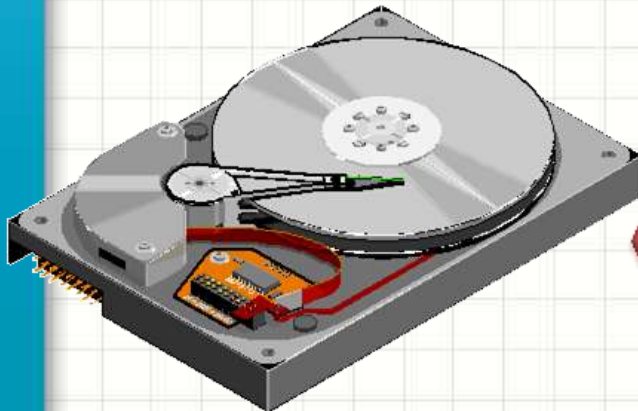
- Como acessar recursos diferentes de maneira igual?



`cout << dado`



`write(dado, pos)`



`out (port2), comando`  
`out (port2), posição`  
`out (port2), dado`





# **S.O. E O COMPARTILHAMENTO DE RECURSOS**

# Compartilhamento de Recursos

- Vários Programas...
  - Mesmos recursos
- Como compartilhá-los de maneira organizada?
- Exemplo:
  - Um programa está imprimindo
  - Outro tenta imprimir
- O que fazer?
  - a) Fazer o segundo programa esperar
  - b) Receber os dados e aguardar



# Compartilhamento de Recursos

- O que mais compartilhar?
  - Tela, teclado, mouse...
    - Múltiplas janelas
  - Disco e rede...
    - Intercalar acessos
- Múltiplos usuários
  - Computador de casa
  - Mainframe
- Sistema Operacional:  
**gerenciador de recursos**

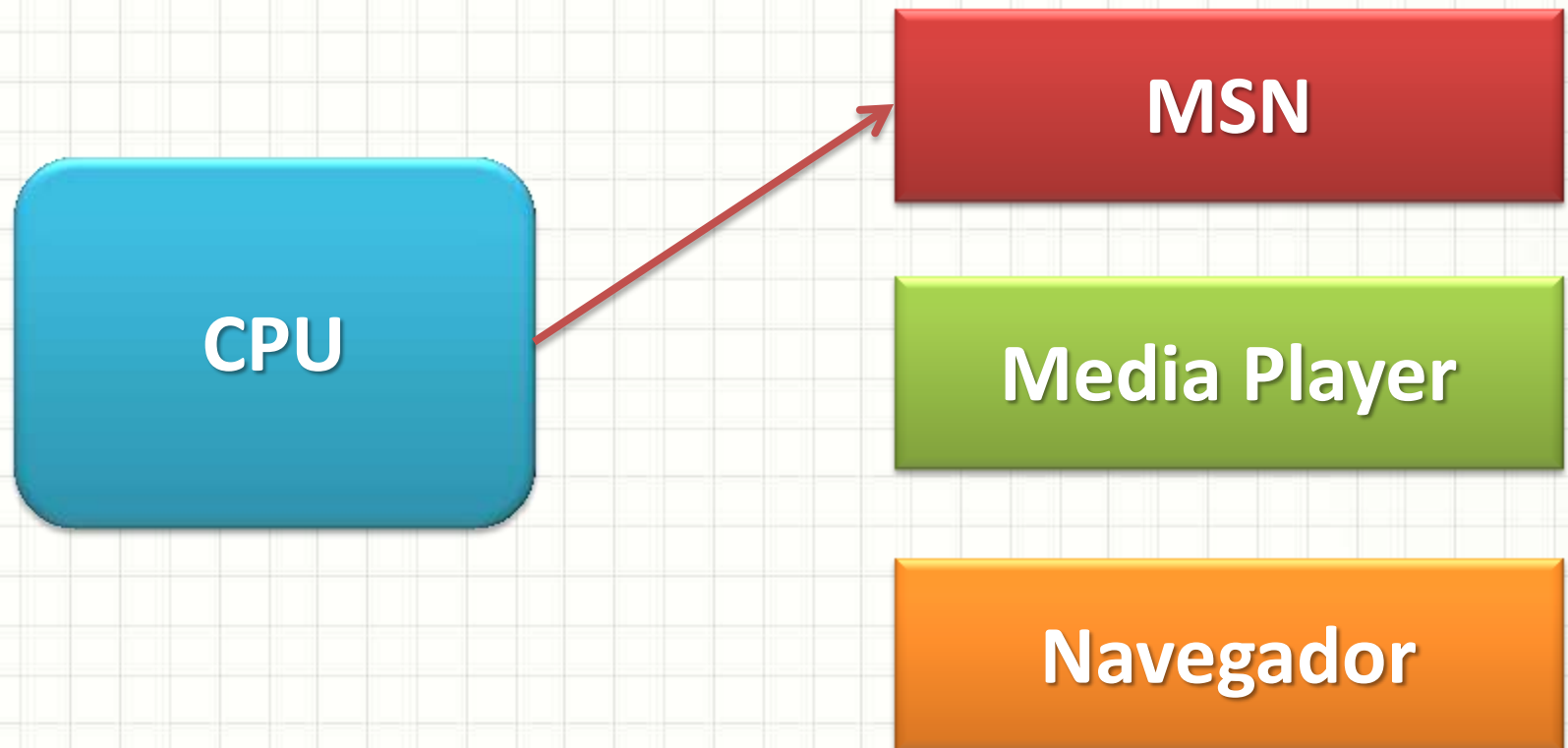




# **GERENCIAMENTO DE PROCESSOS**

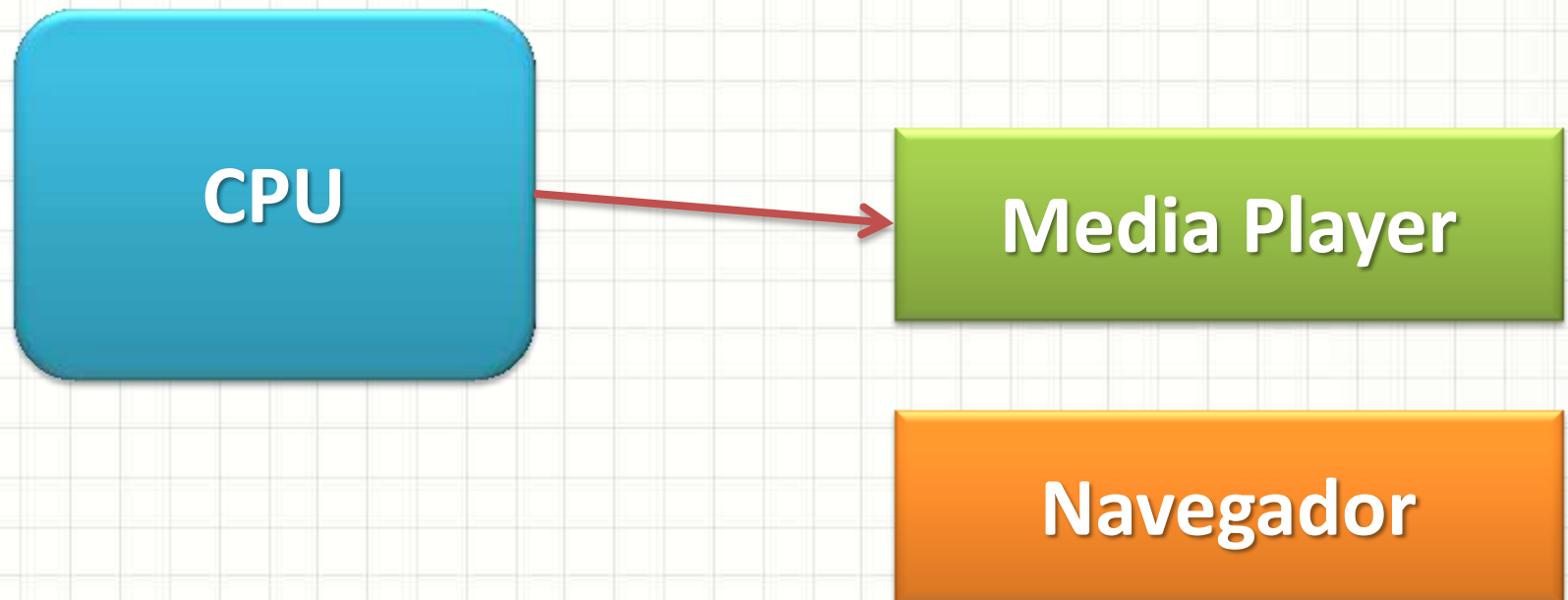
# Gerenciamento de CPU/Processos

- Muitos Processos x Poucas CPUs
  - Processamento Sequencial?



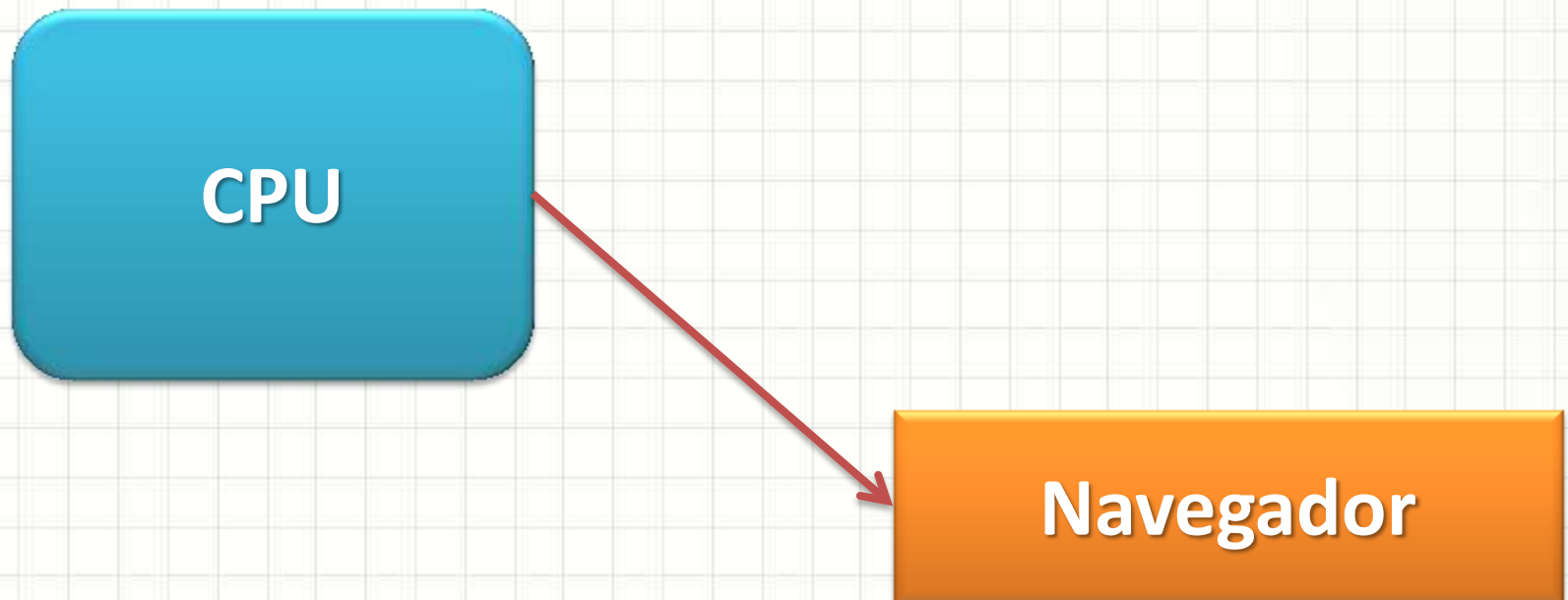
# Gerenciamento de CPU/Processos

- Muitos Processos x Poucas CPUs
  - Processamento Sequencial?



# Gerenciamento de CPU/Processos

- Muitos Processos x Poucas CPUs
  - Processamento Sequencial?



# Gerenciamento de CPU/Processos

- Muitos Processos x Poucas CPUs
  - Problema de gerenciamento

**É assim?**



Navegador



# Escalonamento de Processos

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



**Sistema Operacional**

**A**

**C**

Memória

**B**

**D**

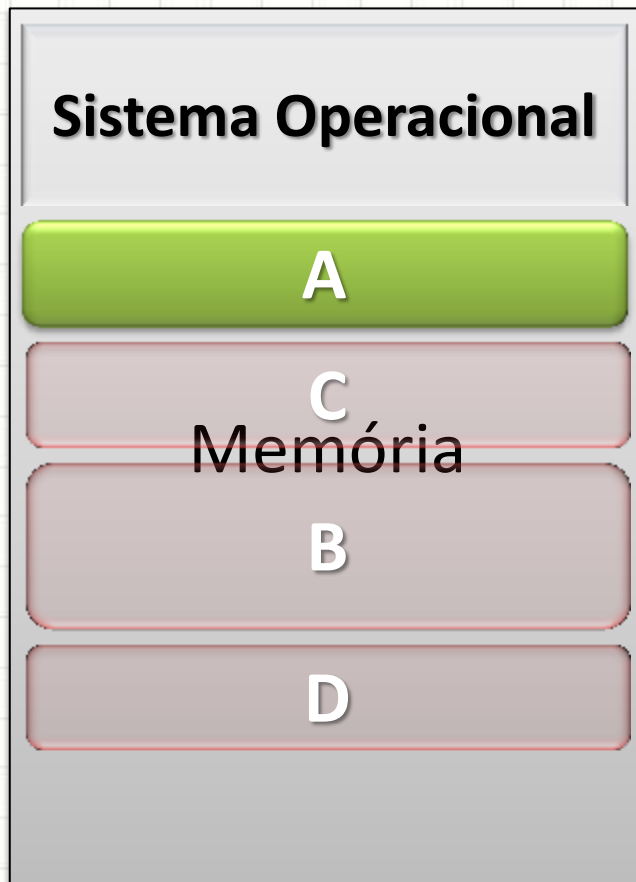
Executando...!



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”



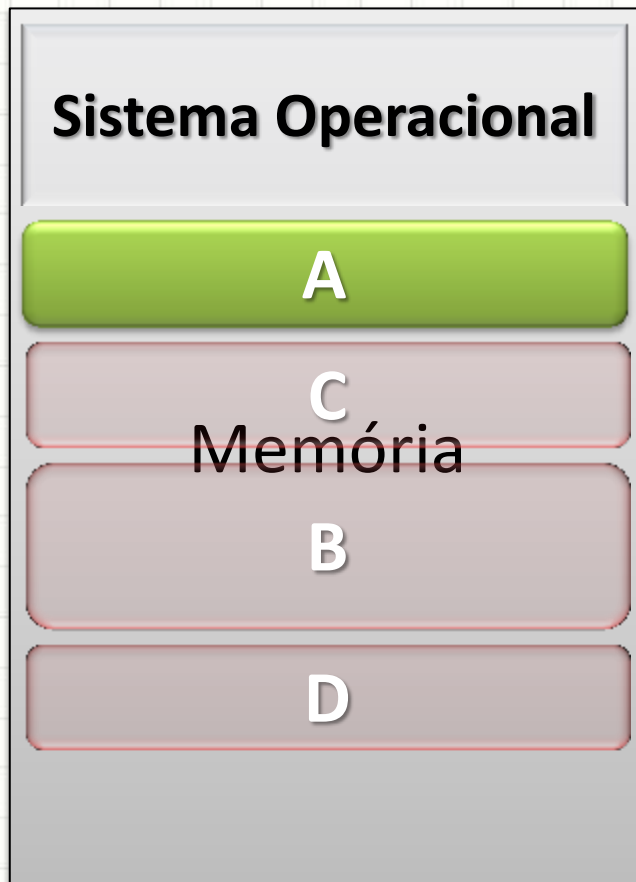
Executando...!



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”



**Sistema Operacional**

**A**

**C**

Memória

**B**

**D**

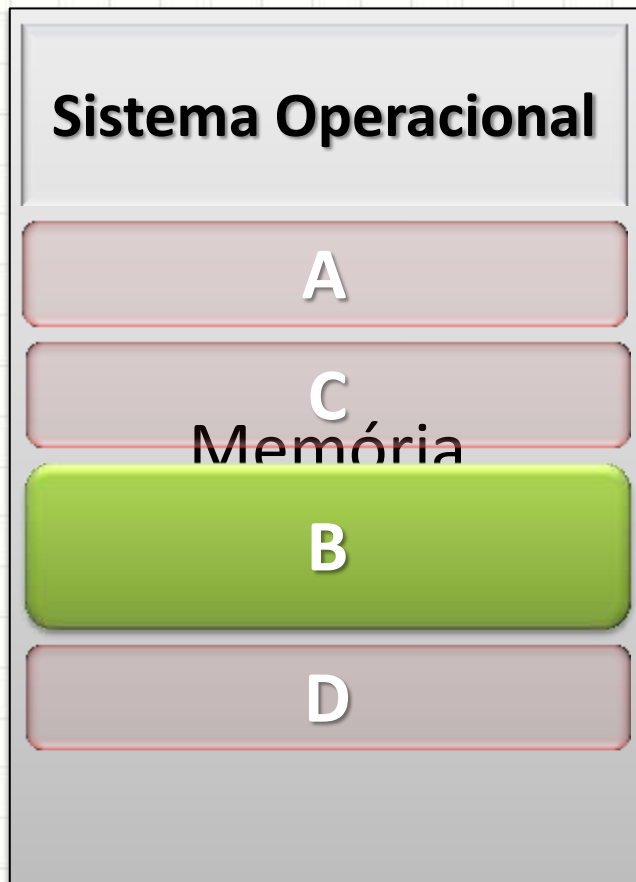
Executando...!



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...!

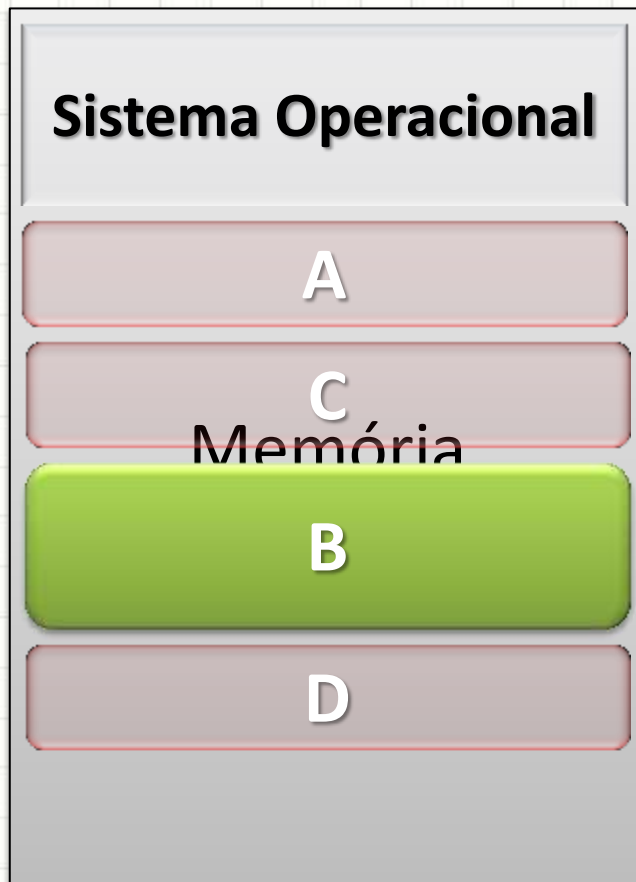


Processo	A.EXE	B.EXE	C.EXE	D.EXE
PID	0	1	2	3
Endereço	0x2000	0x4000	0x3000	0x5500
Som?	0	0	1	0



# 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”



**Sistema Operacional**

**A**

**C**

Memória

**B**

**D**

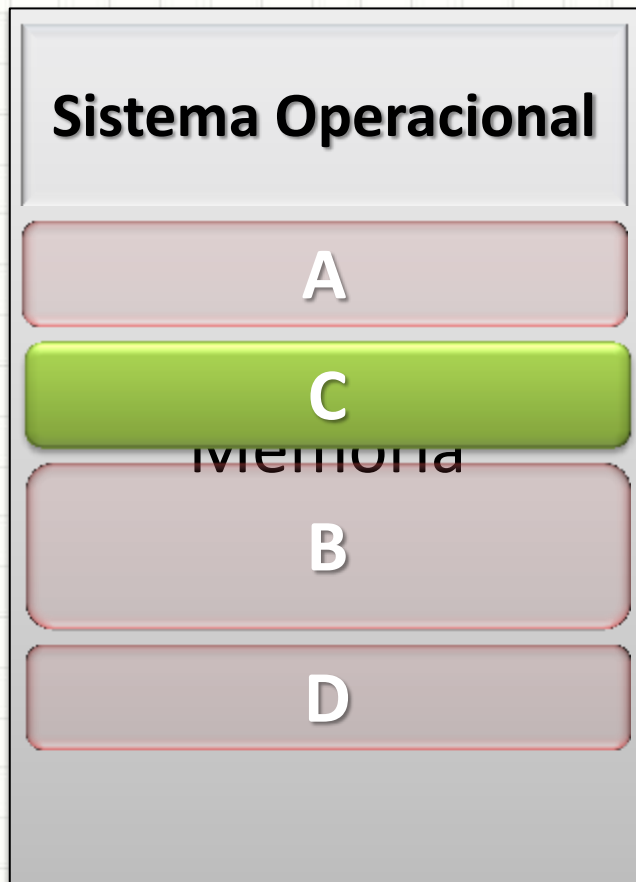
Executando...!



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”



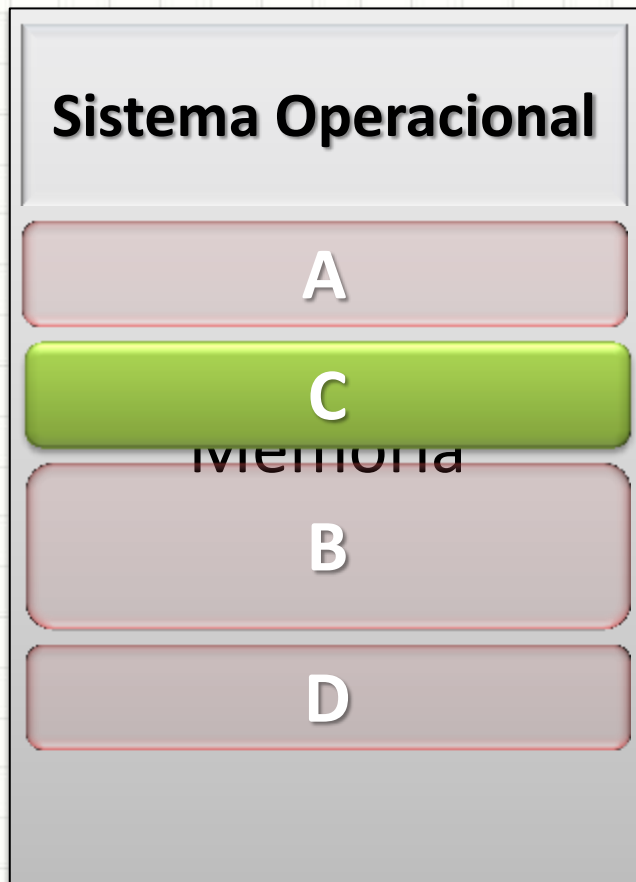
Executando...!



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”



**Sistema Operacional**

**A**


**C**

Memória

**B**

**D**

Executando...!

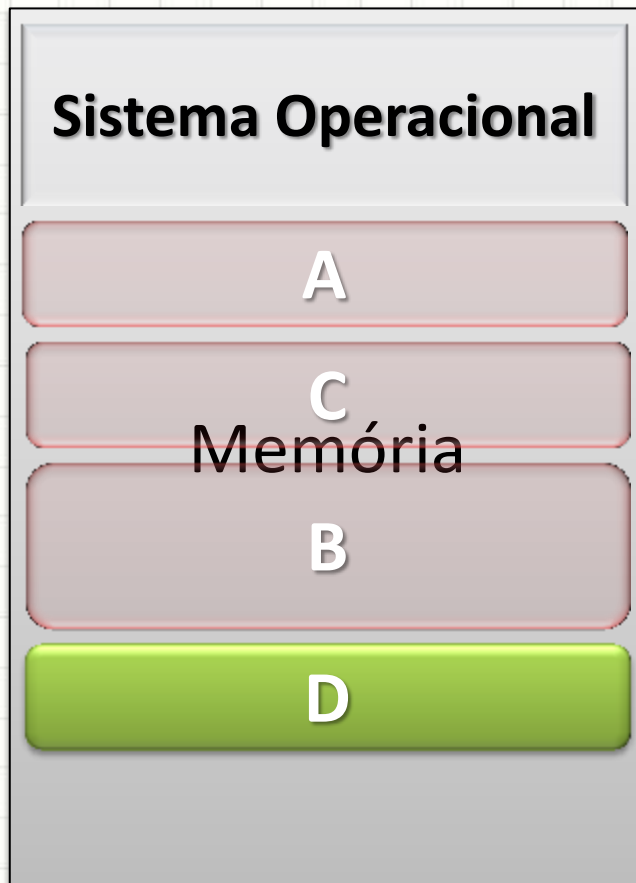


Processo	A.EXE	B.EXE	C.EXE	D.EXE
PID	0	1	2	3
Endereço	0x2000	0x4000	0x3000	0x5500
Som?	0	0	1	0



# Escalonamento de Processos

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



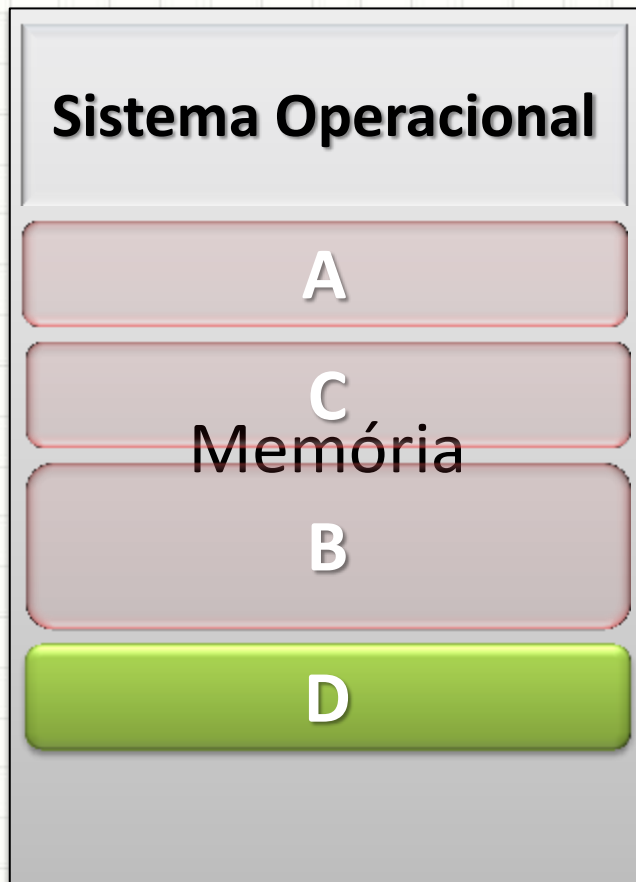
Executando...!



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”



**Sistema Operacional**

**A**

**C**

Memória

**B**

**D**

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

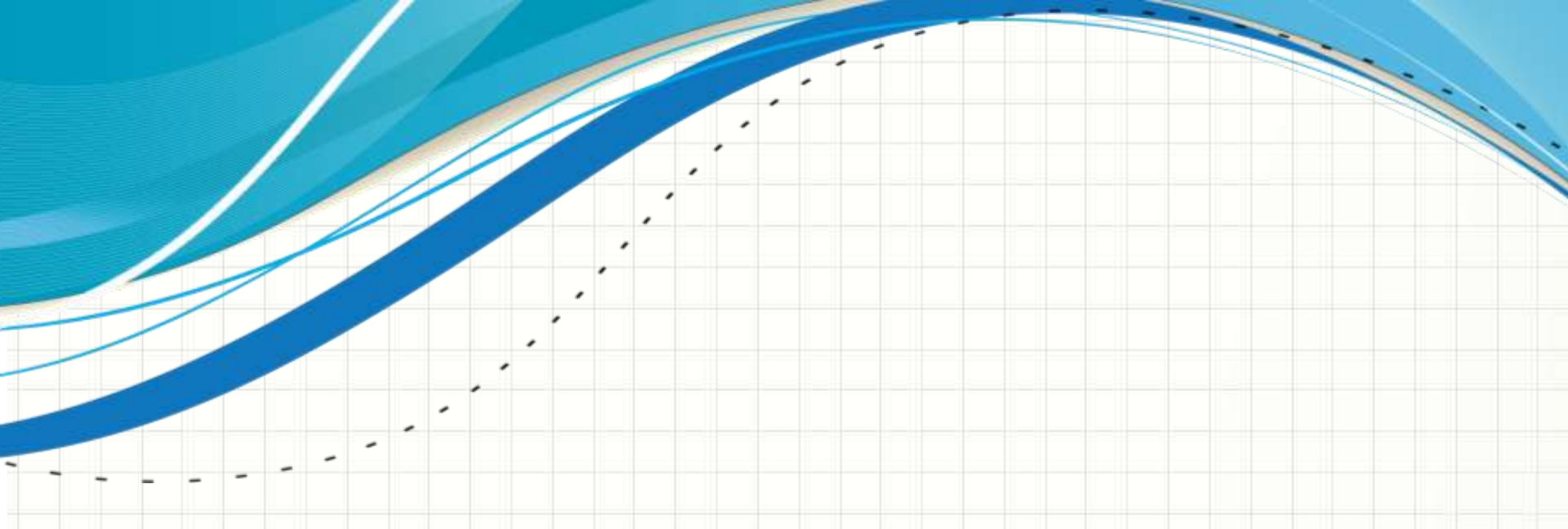


Processo	A.EXE	B.EXE	C.EXE	D.EXE
PID	0	1	2	3
Endereço	0x2000	0x4000	0x3000	0x5500
Som?	0	0	1	0

# 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**

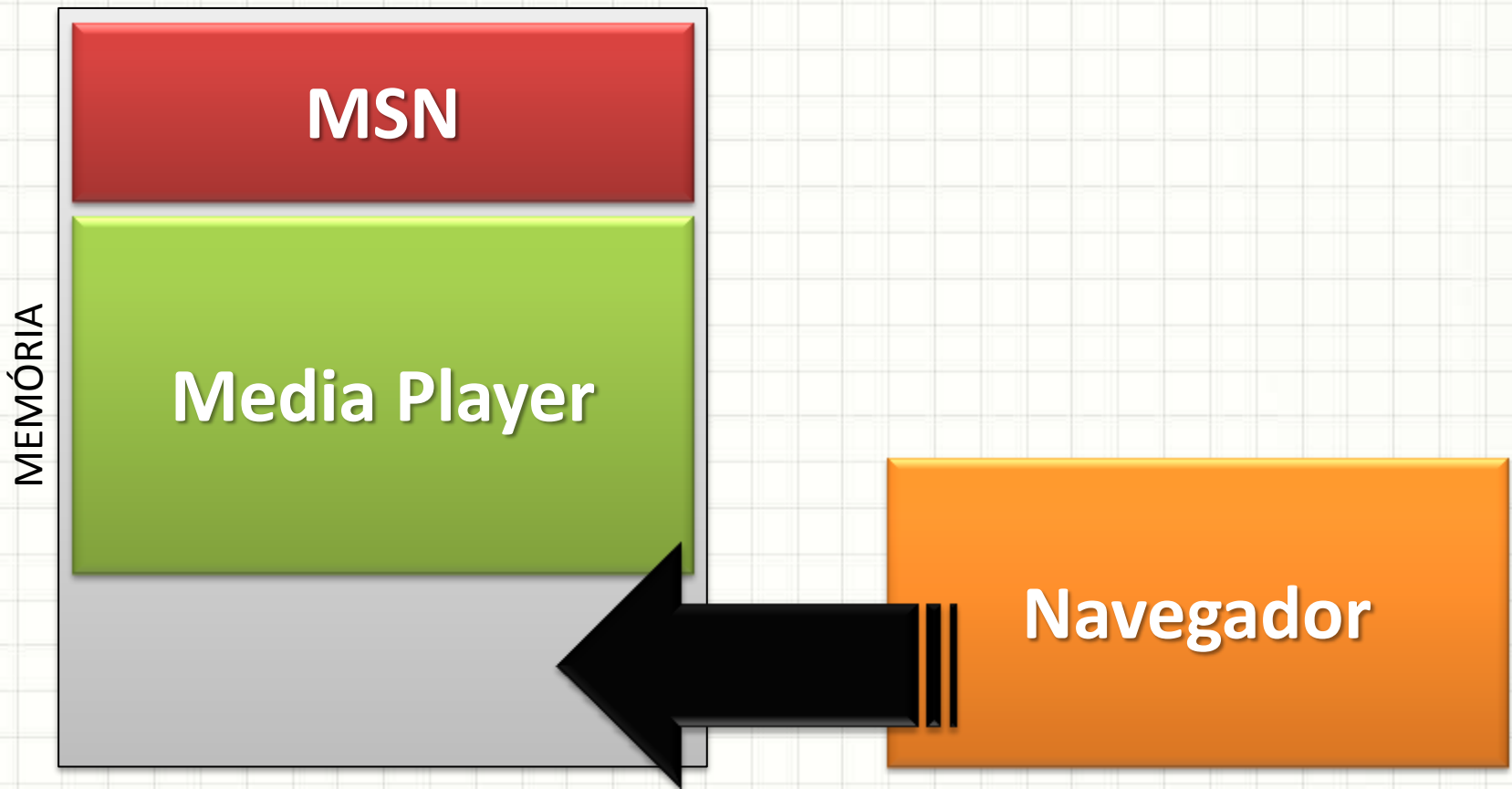
# Gerenciamento de Memória

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



# Gerenciamento de Memória

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



# Gerenciamento de Memória

- Muitos Processos x Pouca Memória
  - O que fazer quando não há memória disponível?

**Falta  
memória?**

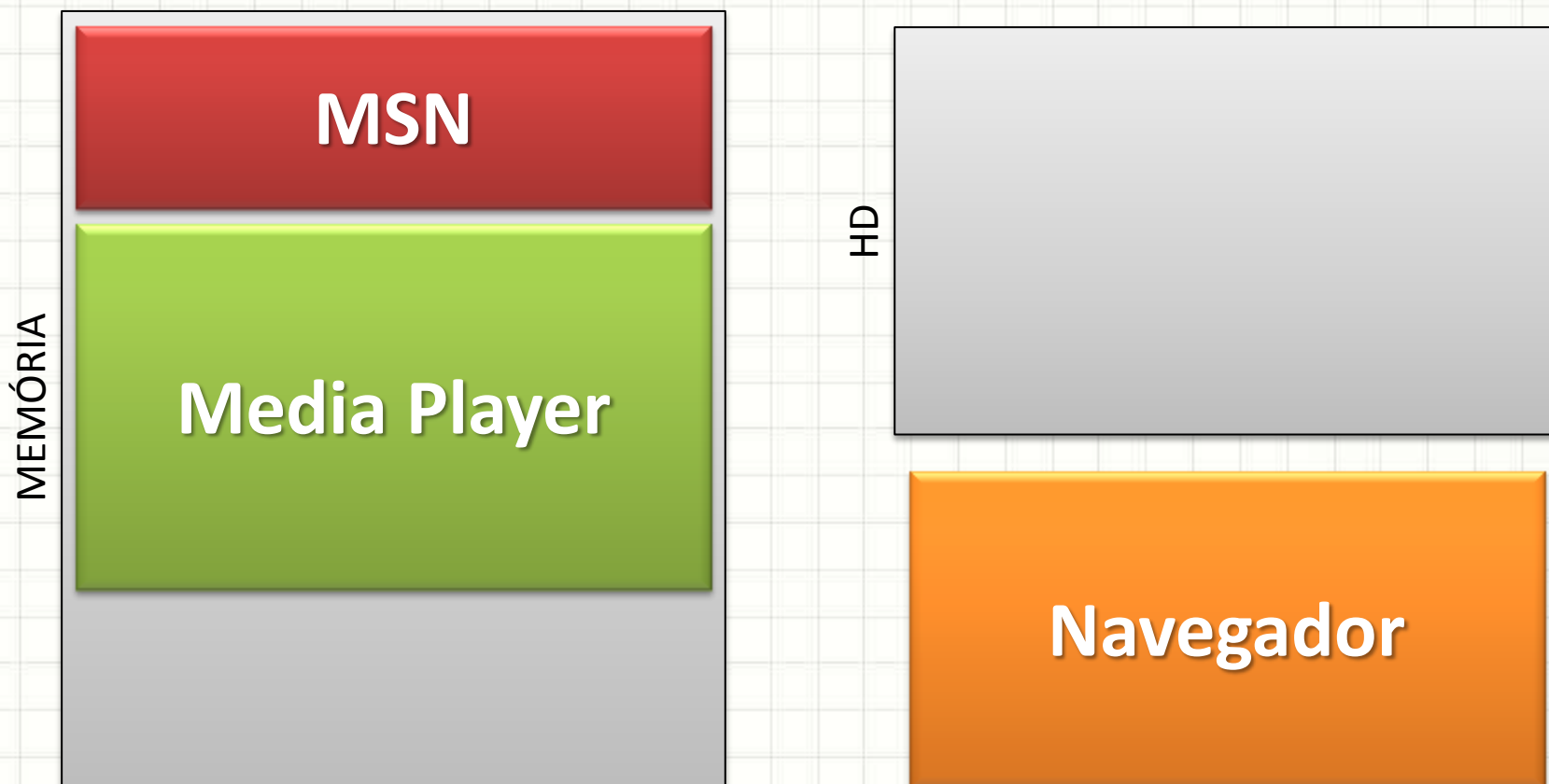
MEMÓRIA

Media P

Navegador

# Gerenciamento de Memória

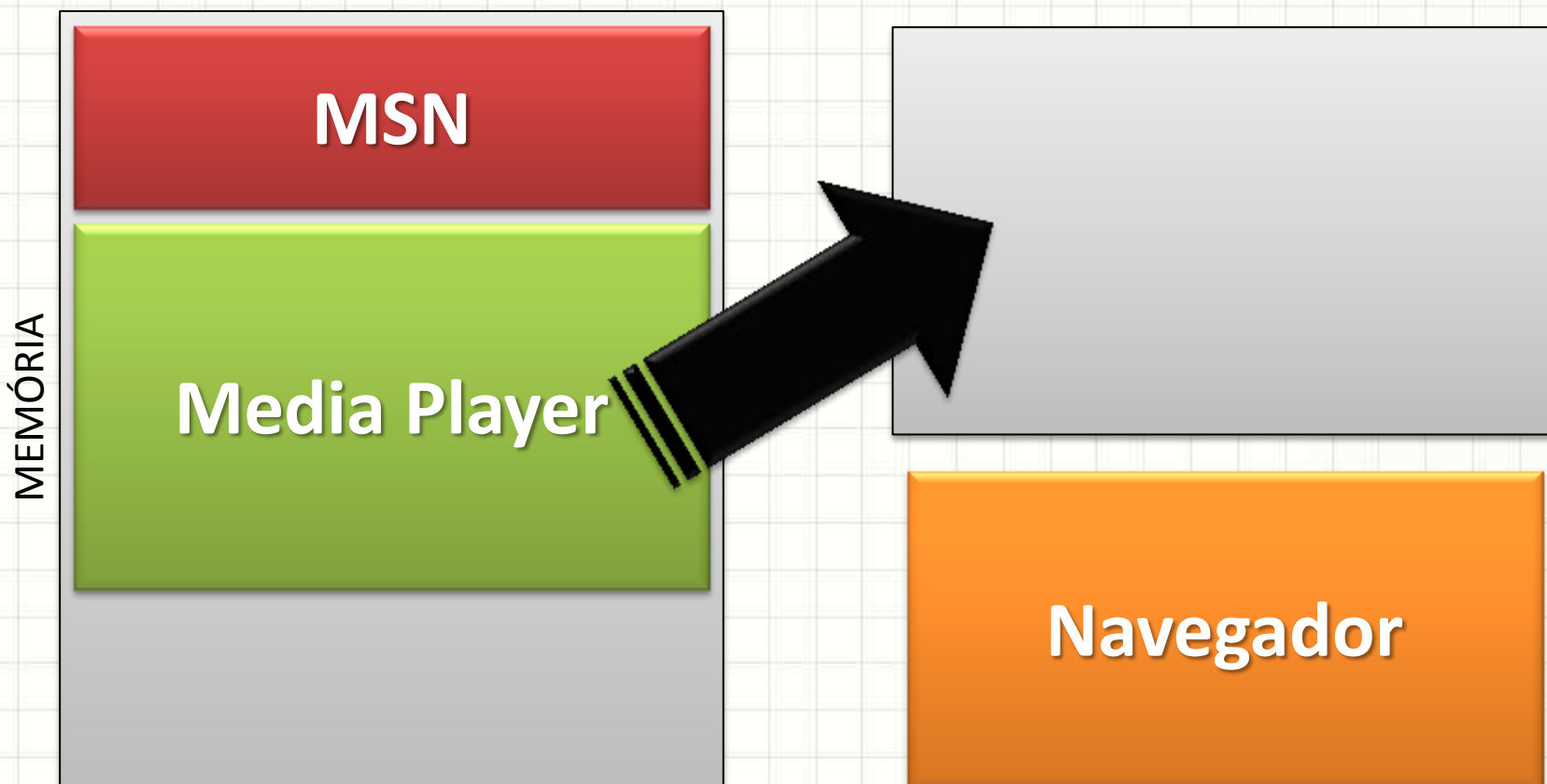
- Muitos Processos x Pouca Memória
  - Quando não cabe: Move programa menos usado





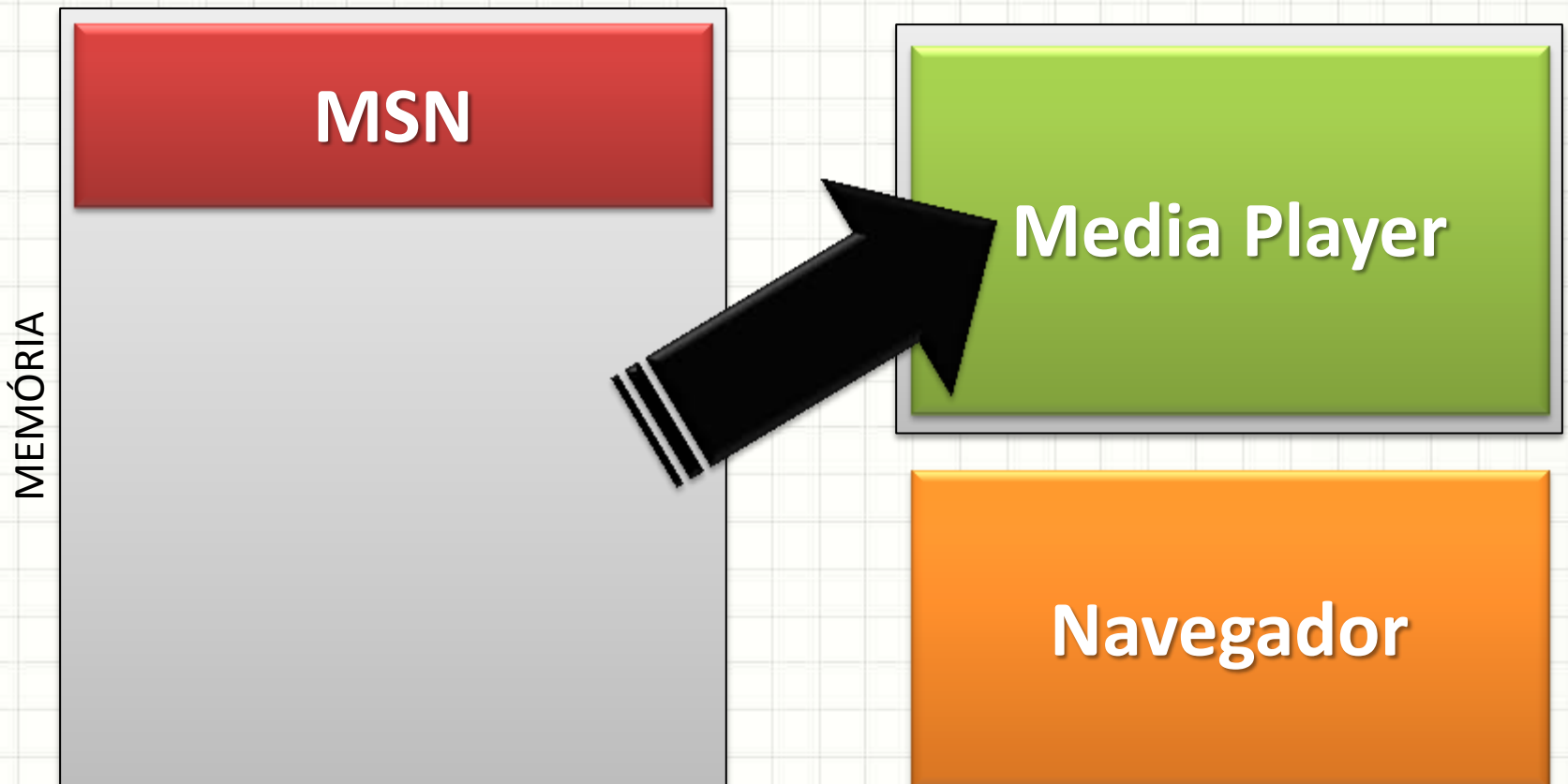
# Gerenciamento de Memória

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



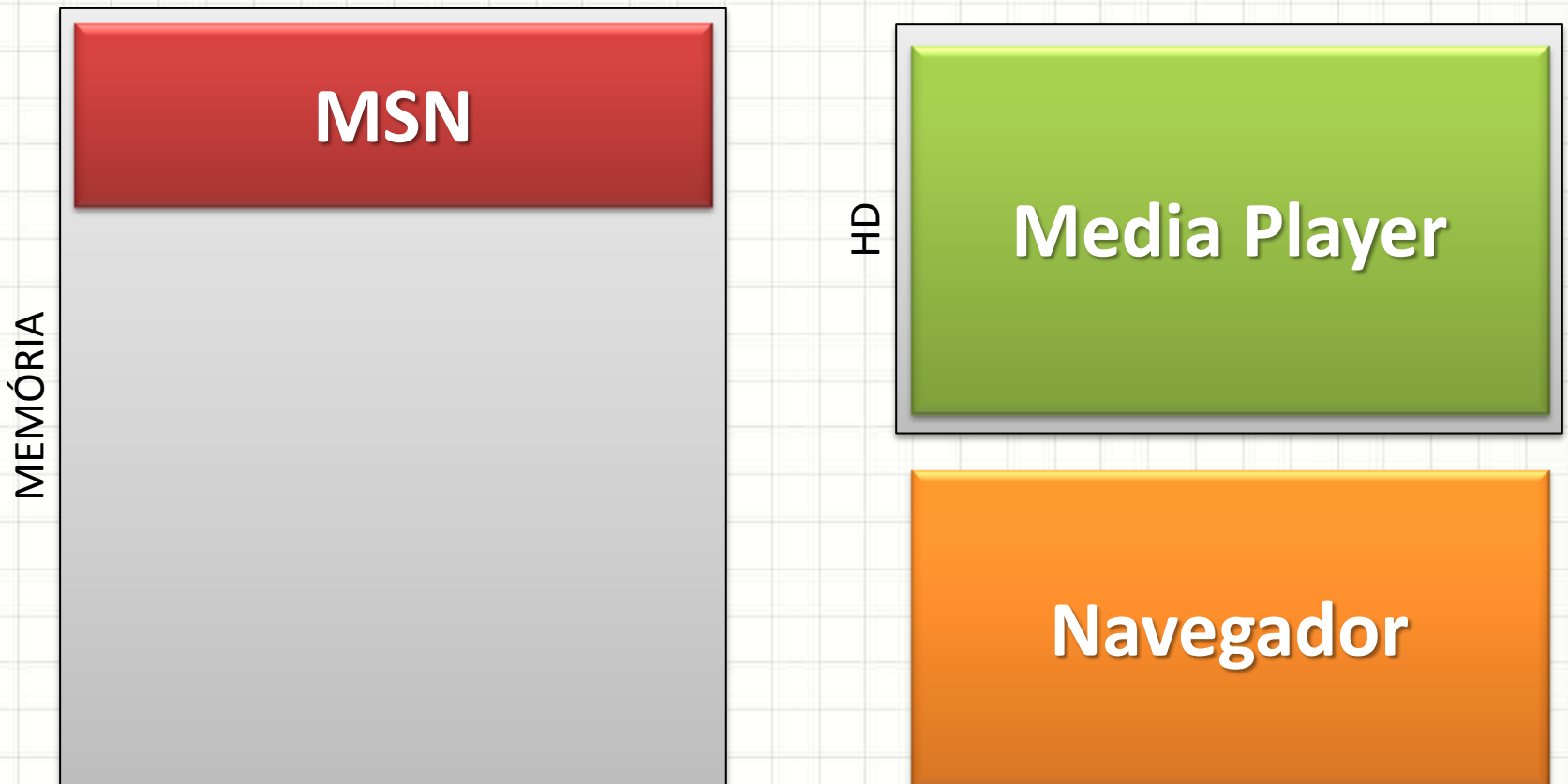
# Gerenciamento de Memória

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



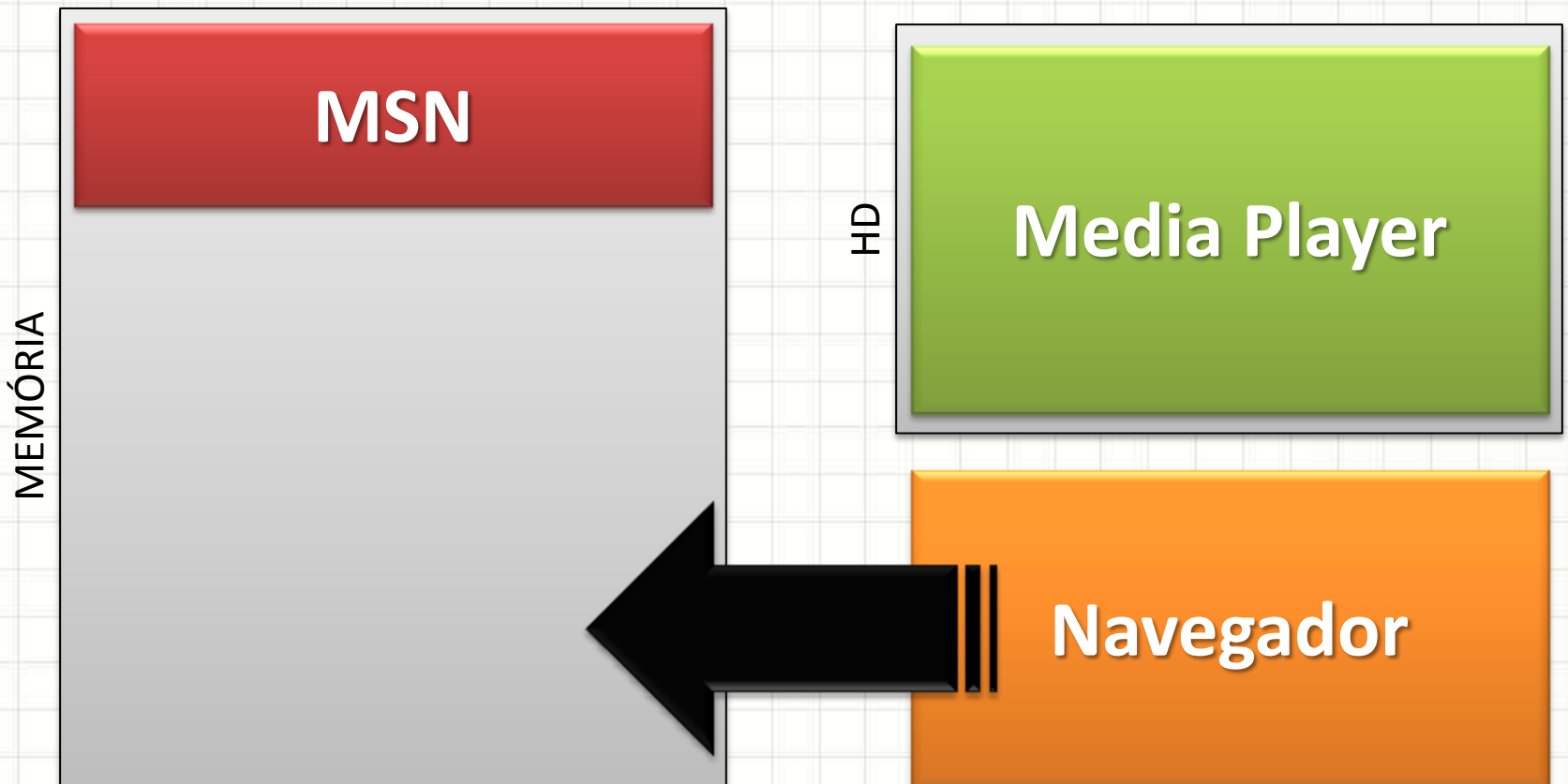
# Gerenciamento de Memória

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



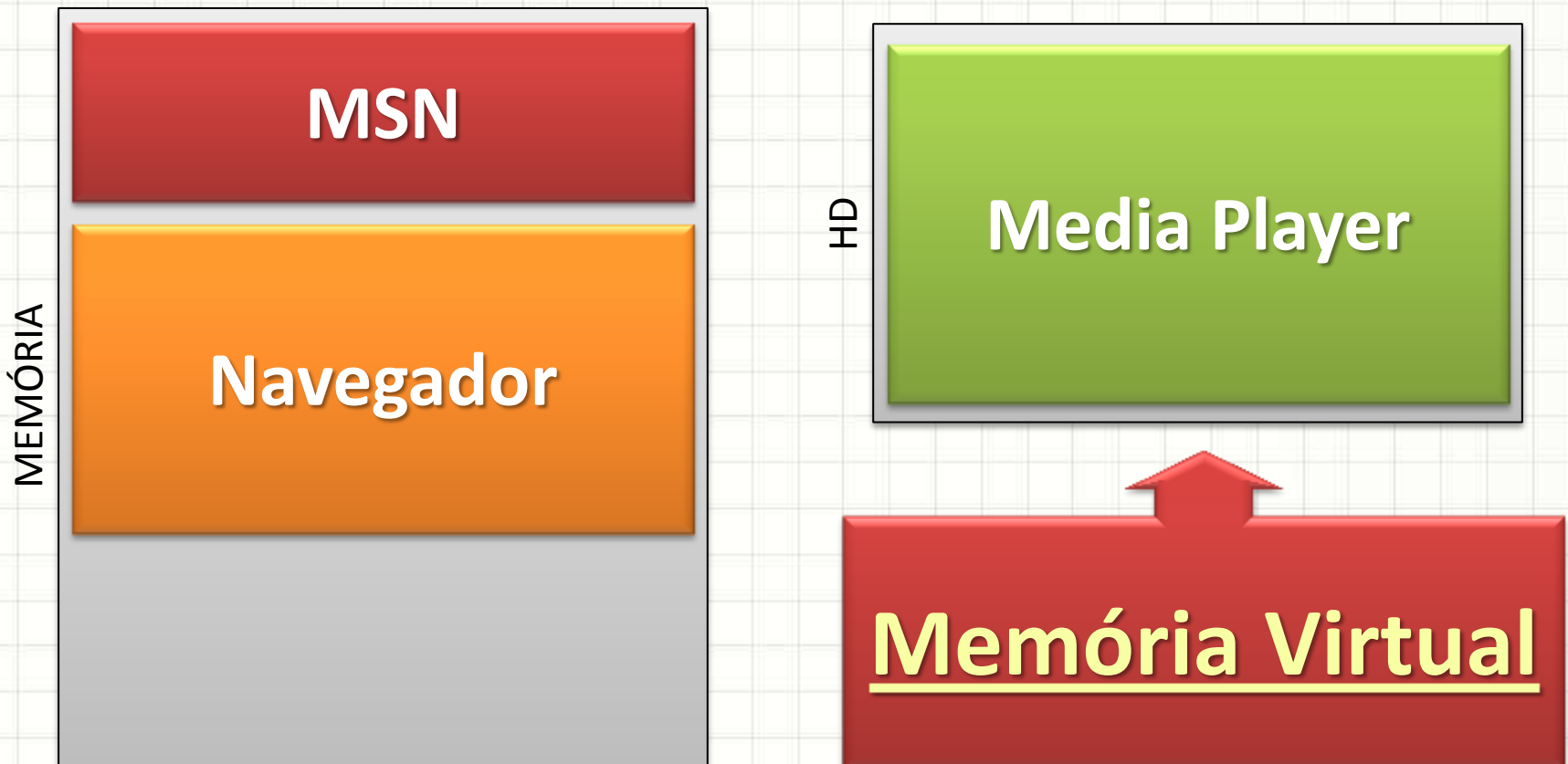
# Gerenciamento de Memória

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



# Gerenciamento de Memória

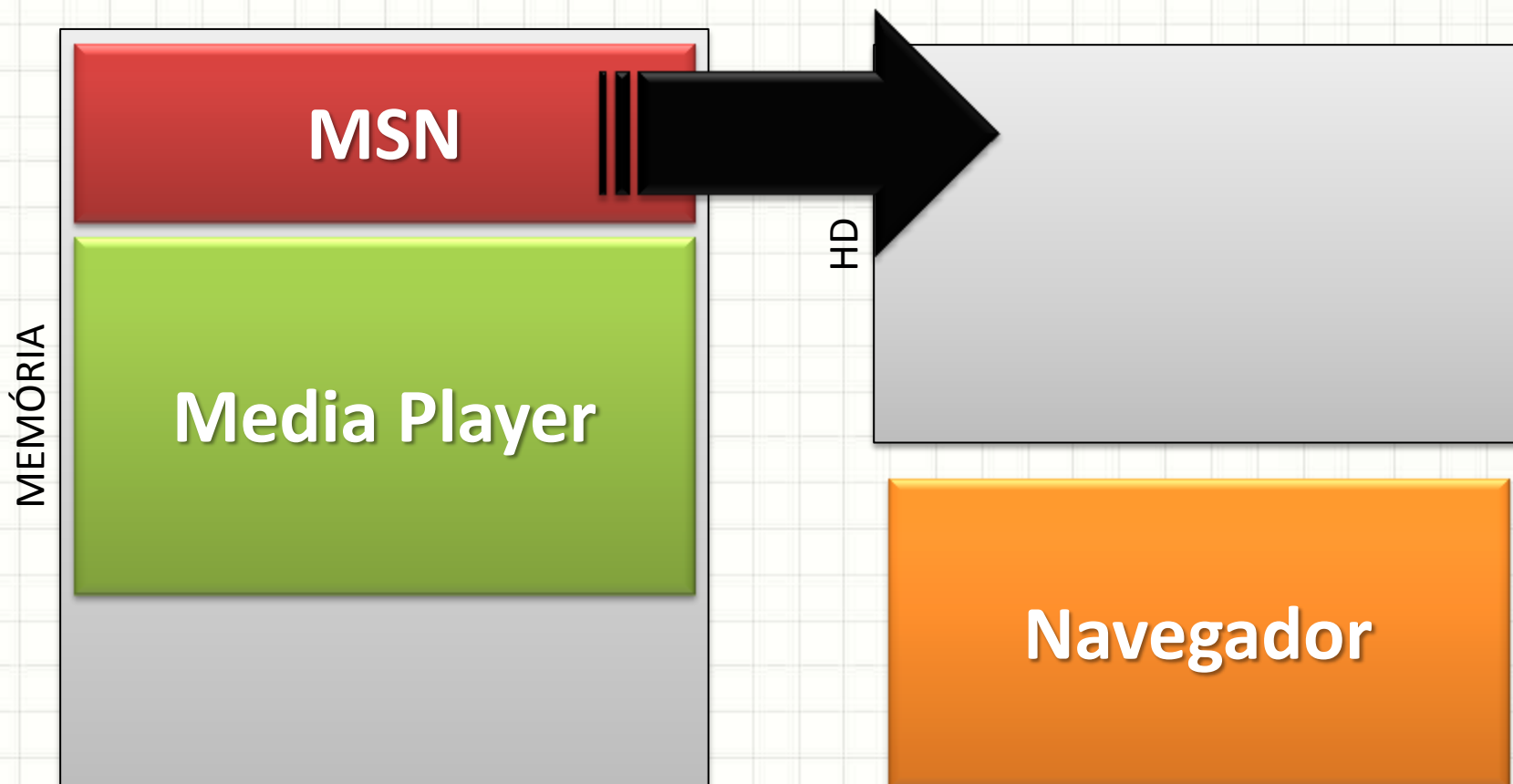
- Muitos Processos x Pouca Memória
  - Quando não cabe: Move programa menos usado





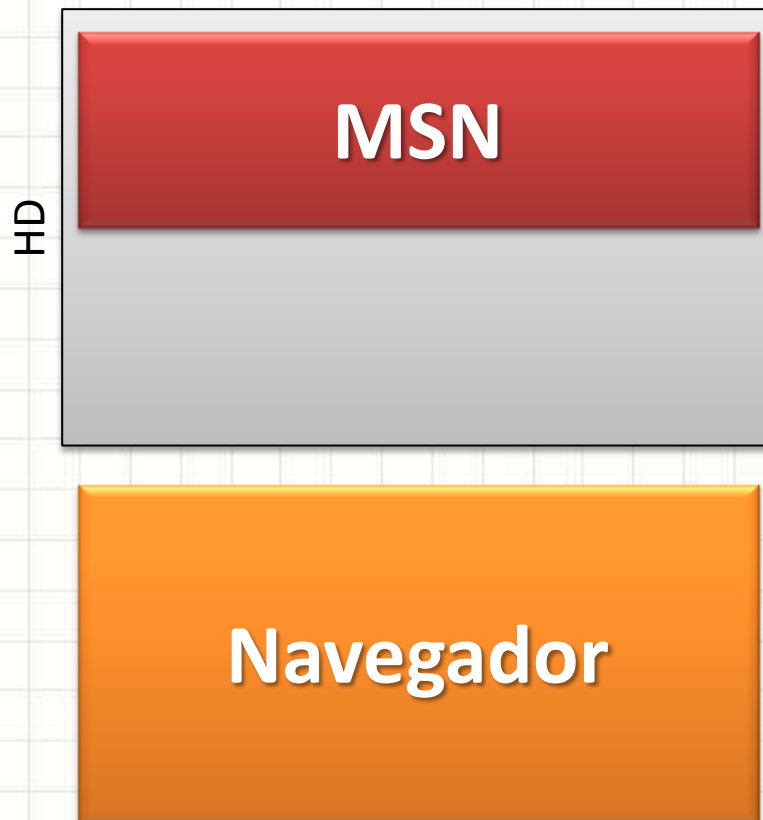
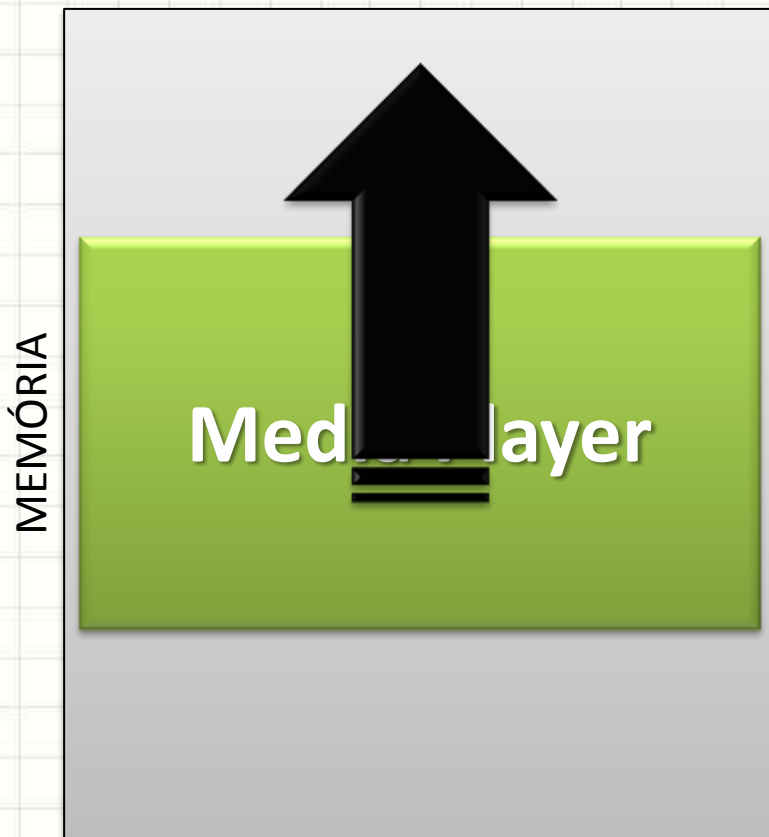
# Gerenciamento de Memória

- Mas e se o programa sem uso fosse o MSN?



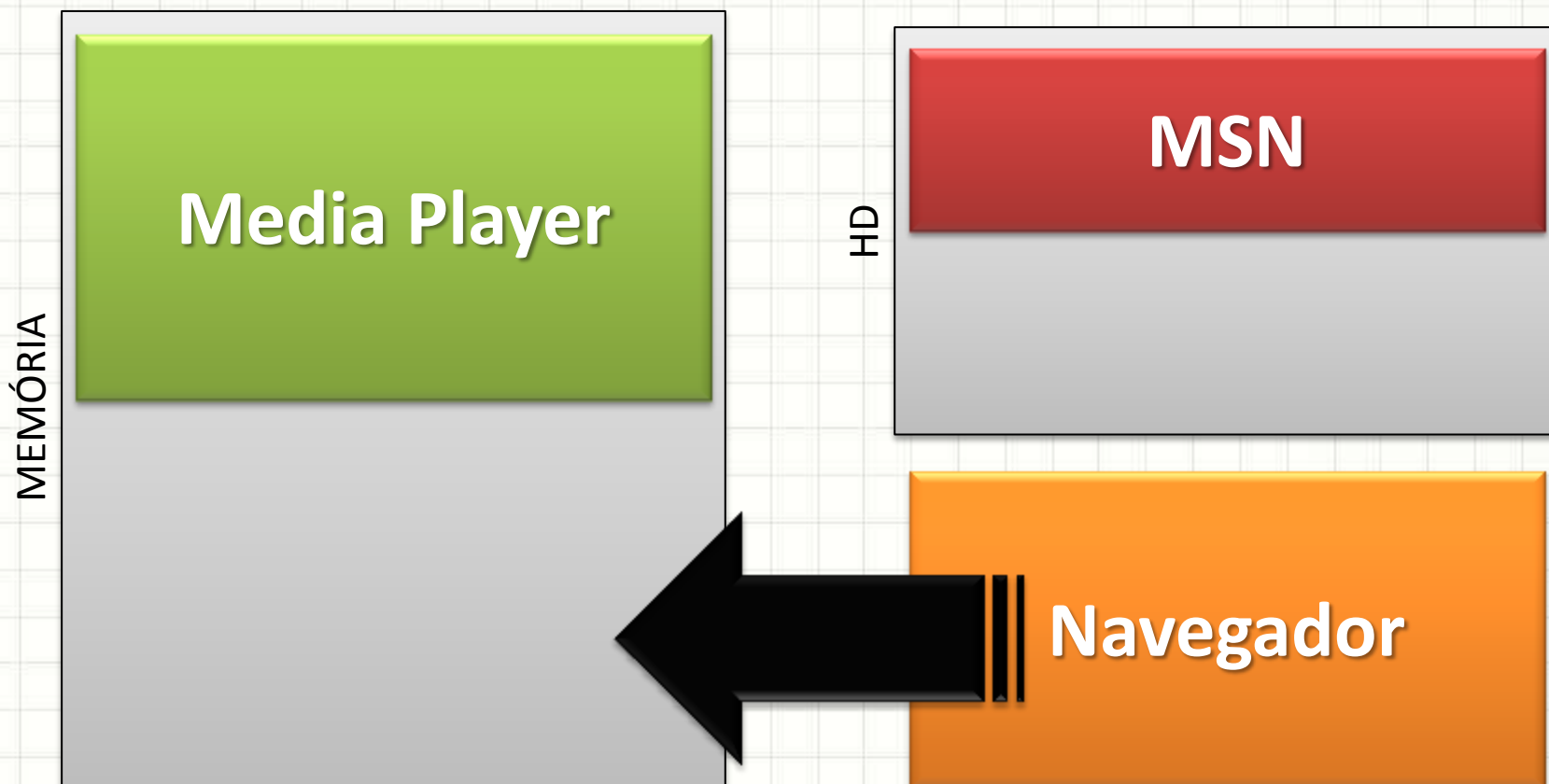
# Gerenciamento de Memória

- Mas e se o programa sem uso fosse o MSN?



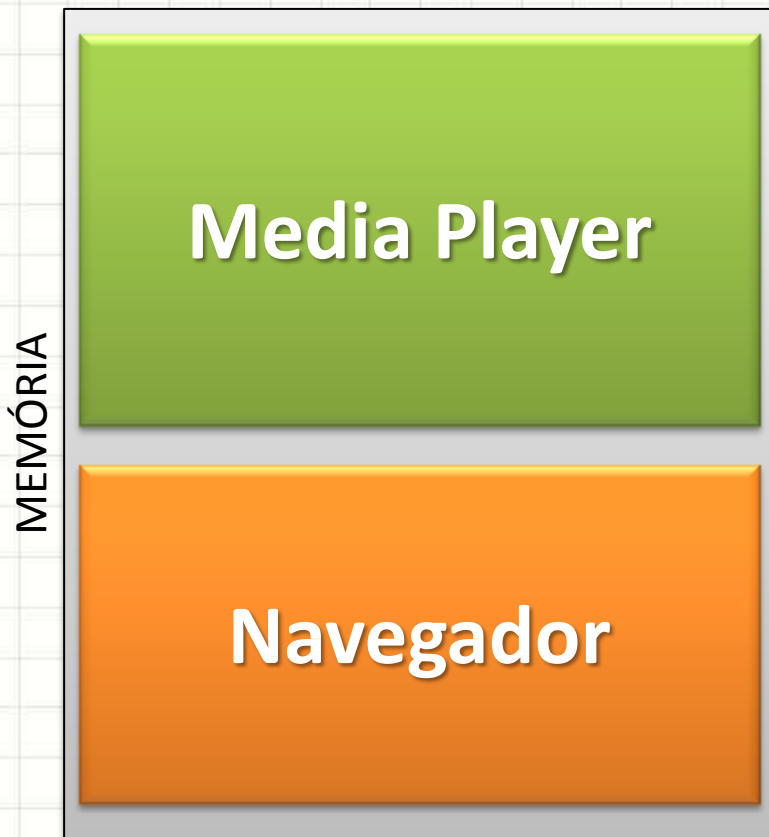
# Gerenciamento de Memória

- Mas e se o programa sem uso fosse o MSN?



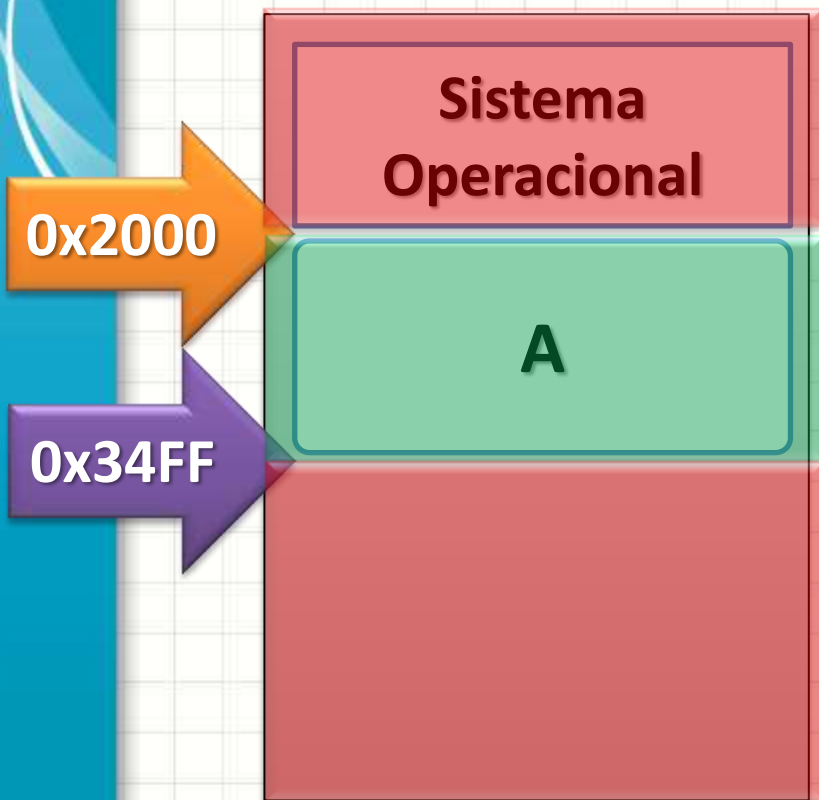
# Gerenciamento de Memória

- Mas e se o programa sem uso fosse o MSN?



# 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)

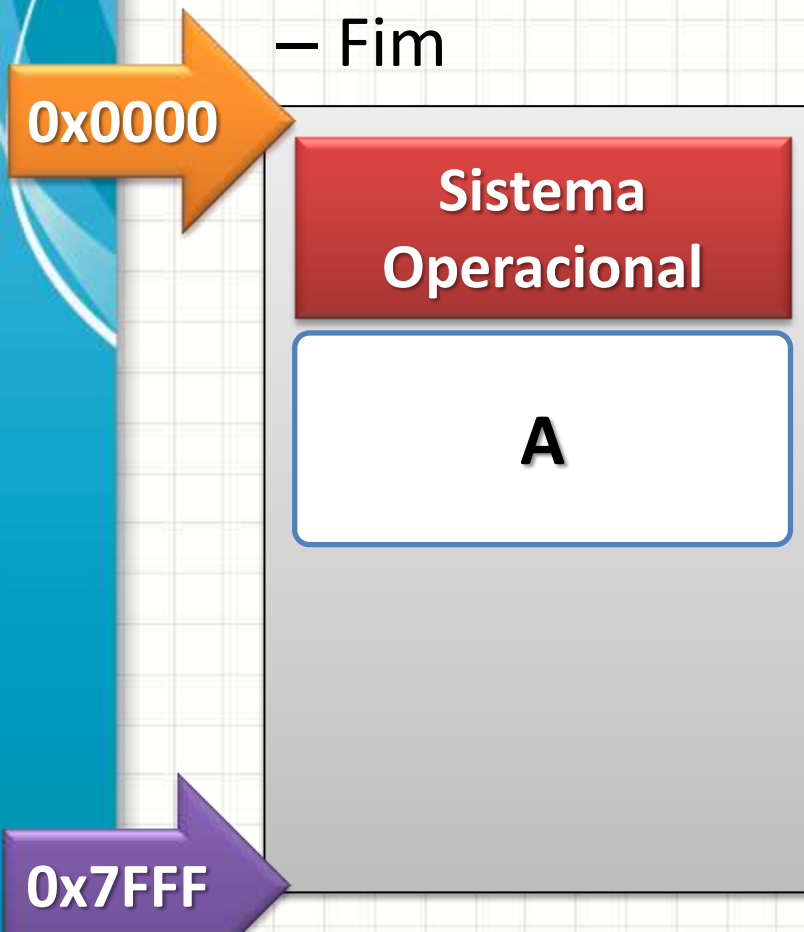


# Gerenciamento de Memória

- MMUs Modernas: Dois Registradores

- Início

- Fim



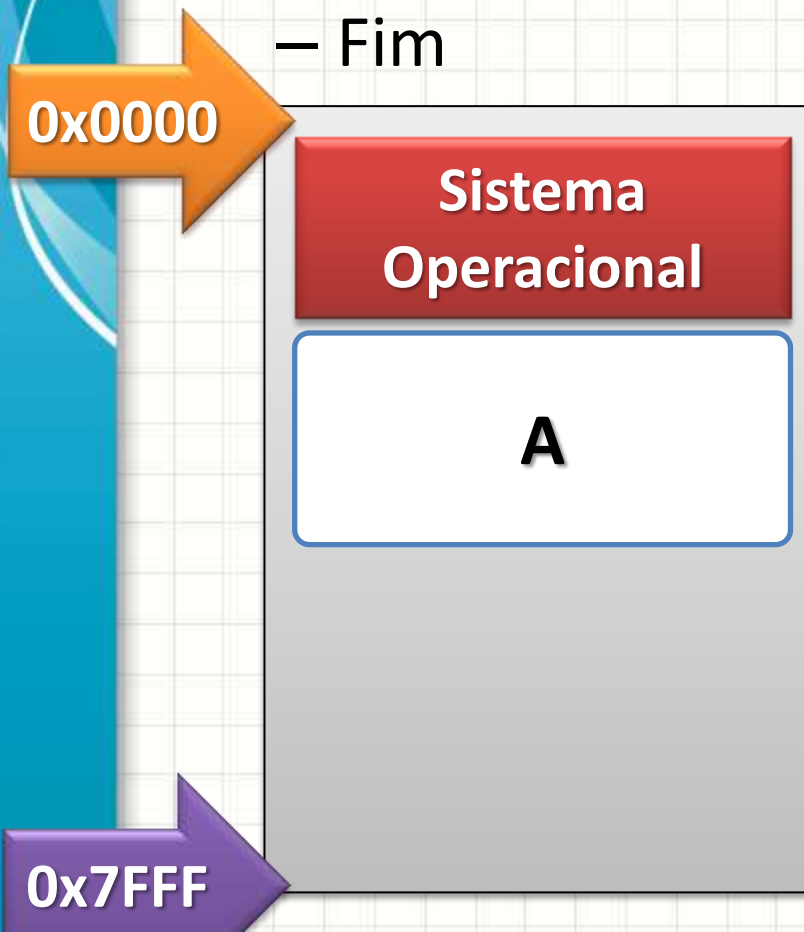
- O sistema operacional tem acesso a tudo...

# Gerenciamento de Memória

- MMUs Modernas: Dois Registradores

- Início

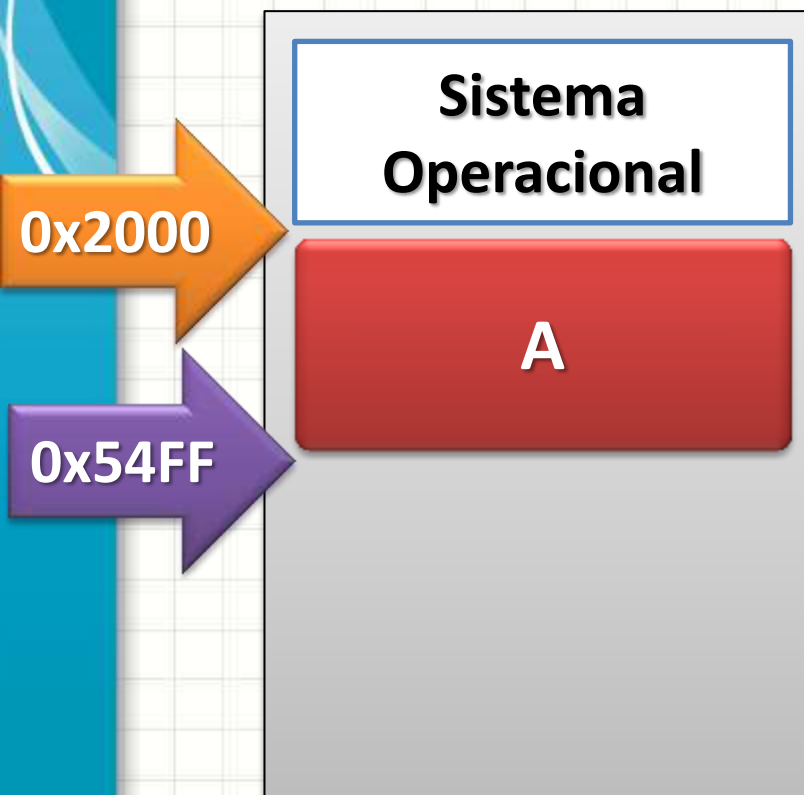
- Fim



- Quando um processo vai entrar em execução...

# Gerenciamento de Memória

- MMUs Modernas: Dois Registradores
  - Início
  - Fim



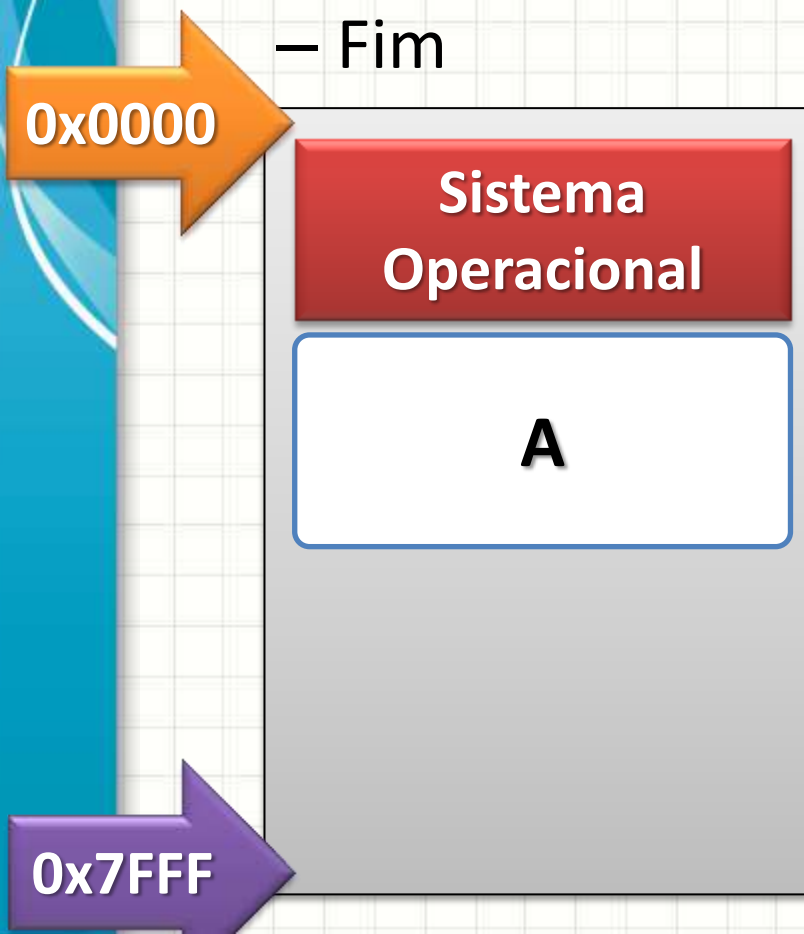
- Quando um processo vai entrar em execução...
- A MMU é reconfigurada!
- Quando acaba a “fatia de tempo” do processo...

# Gerenciamento de Memória

- MMUs Modernas: Dois Registradores

- Início

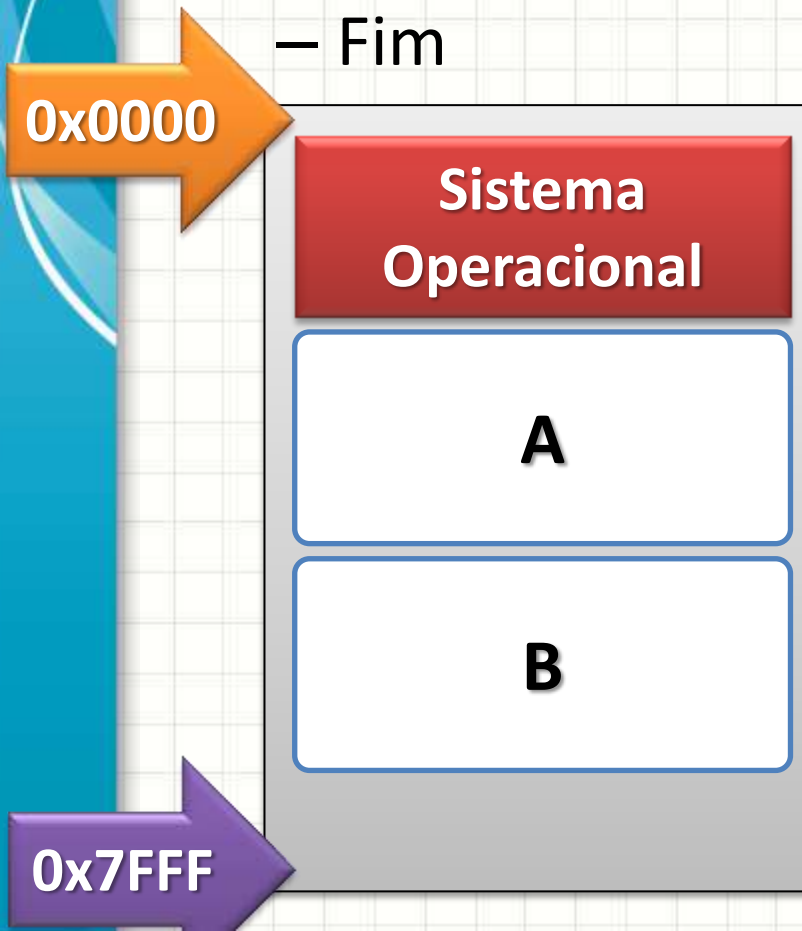
- Fim



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

# Gerenciamento de Memória

- MMUs Modernas: Dois Registradores
  - Início
  - Fim

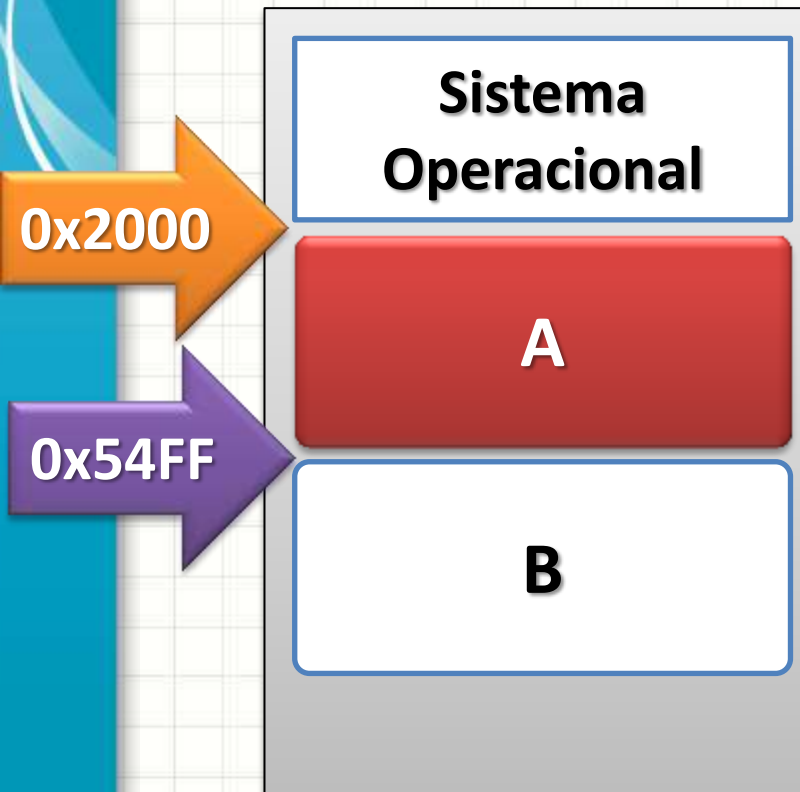


- Vantagem...
- Podemos ter mais processos protegidos entre si



# Gerenciamento de Memória

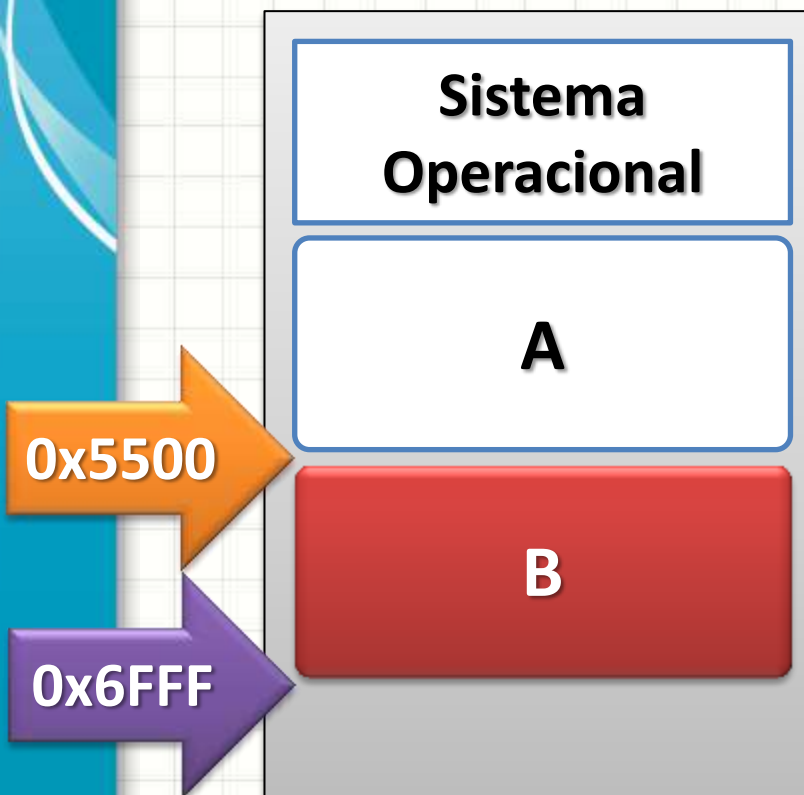
- MMUs Modernas: Dois Registradores
  - Início
  - Fim



- 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



# **GERENCIAMENTO DE DISPOSITIVOS**

# Gerenciamento de Dispositivos

- Responsável pelo controle de funções básicas...
  - Polling, IRQ, DMA...
- Controle compartilhamento dos dispositivos
- Dispositivos podem ser
  - Compartilhados
  - Acesso exclusivo
    - Spool
    - Espera



# CONCLUSÕES



# Resumo

- S.O.: software intermediário
  - usuário  $\leftrightarrow$  programas
  - programas  $\leftrightarrow$  hardware
- Recursos do Hardware: controlados pelo S.O.
- Troca de Processos: timer de hardware  $\rightarrow$  IRQ
- Acesso à Memória: MMU
- **TAREFA**
  - Lista 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!**