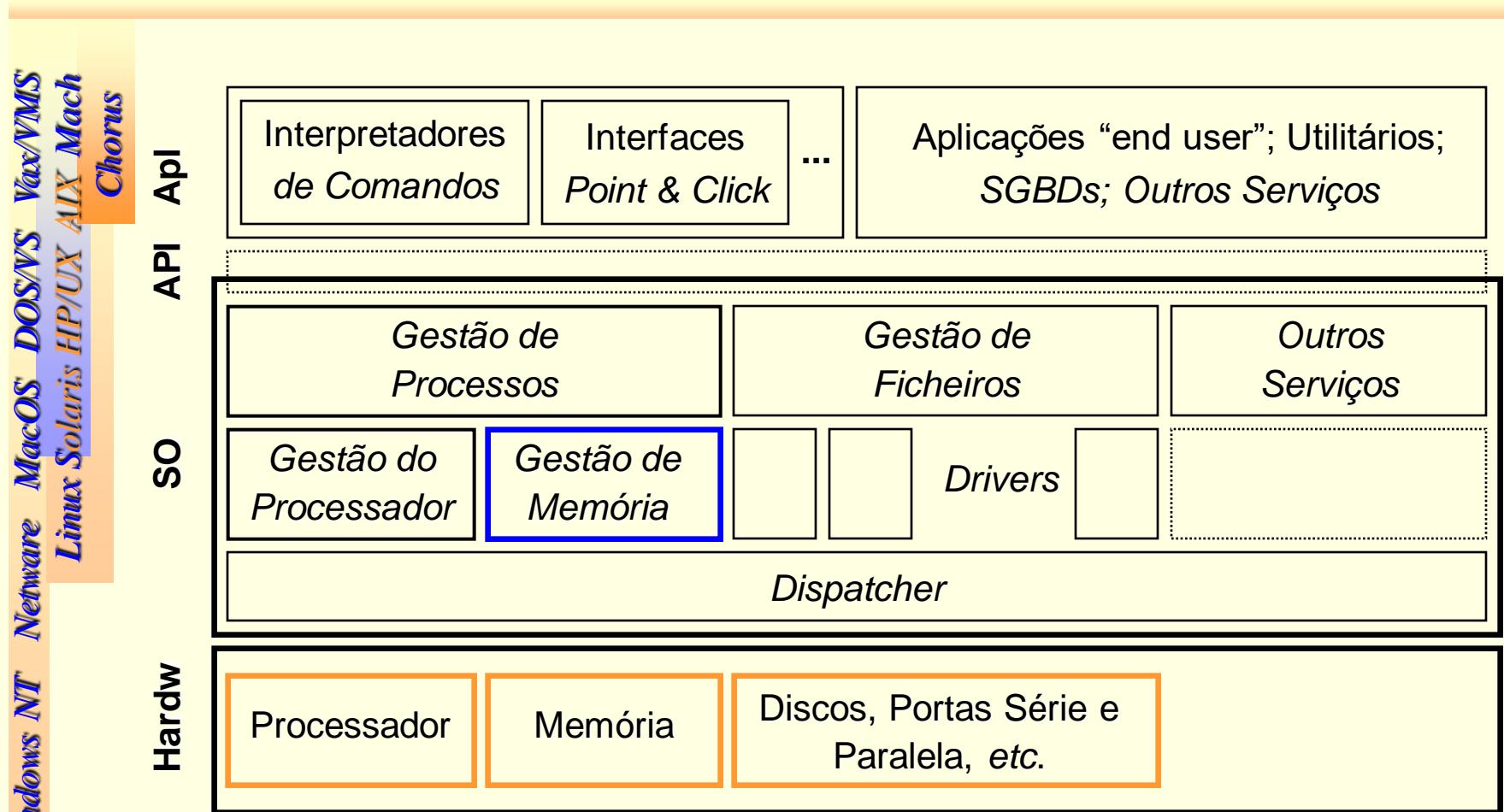


Fundamentos de Sistemas de Operação

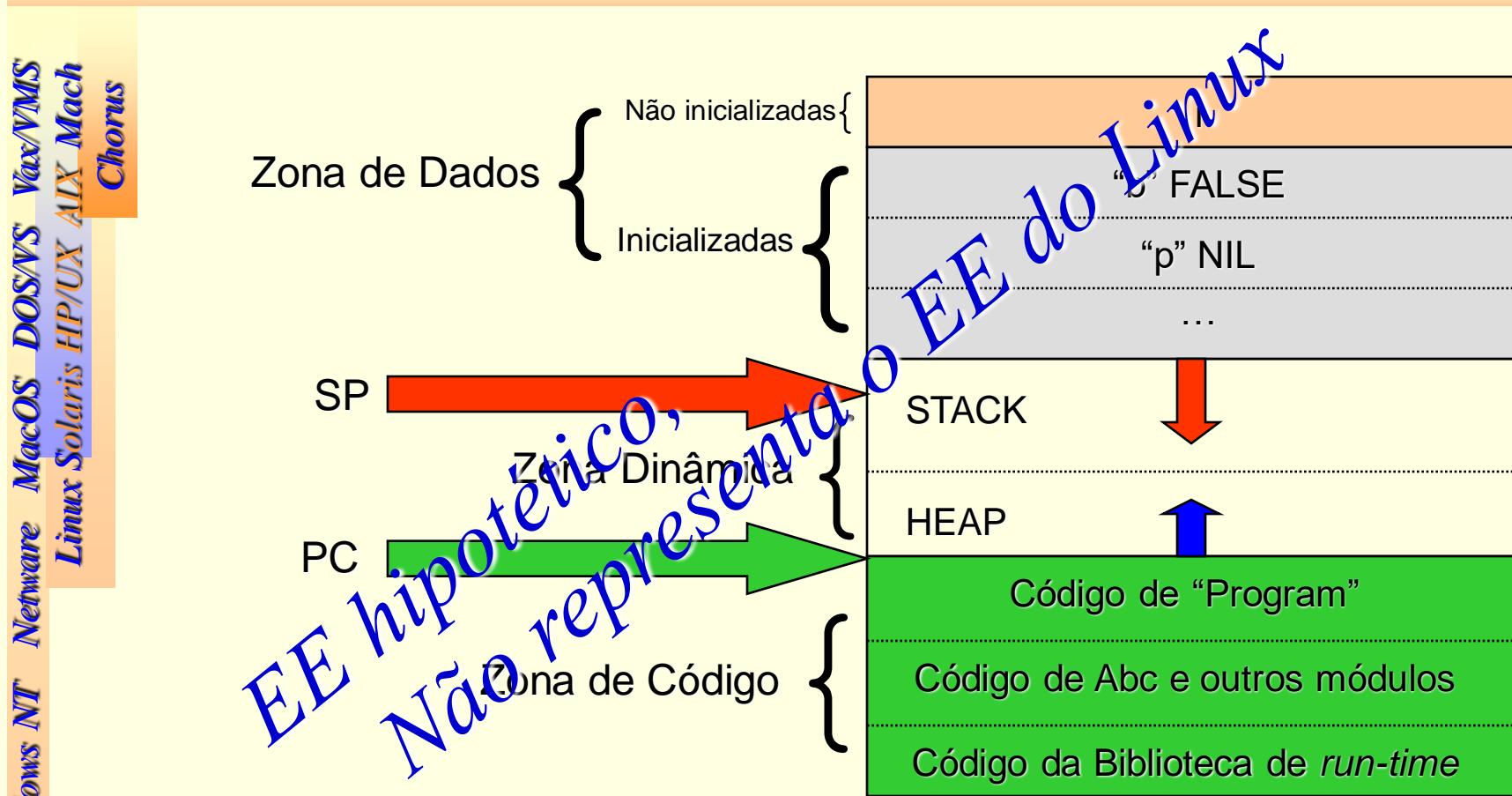
Unix Windows NT Netware Mac OS DOS/V/VS Vax/VMS
Linux Solaris HP/UX AIX Mach Chorus

Evolução das técnicas de Gestão de Memória
a) Partição única

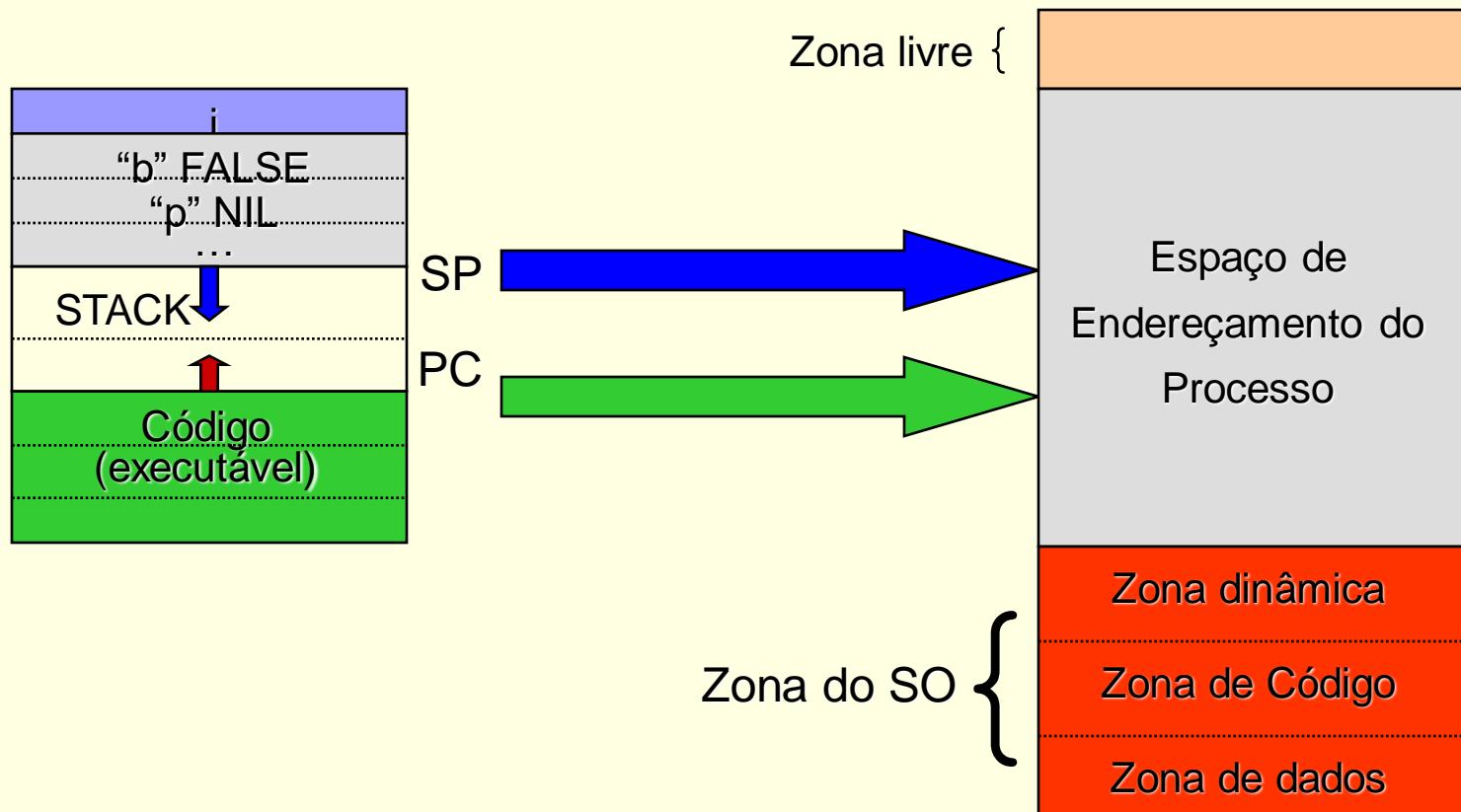
A Gestão de Memória



Como colocar em RAM o Processo?



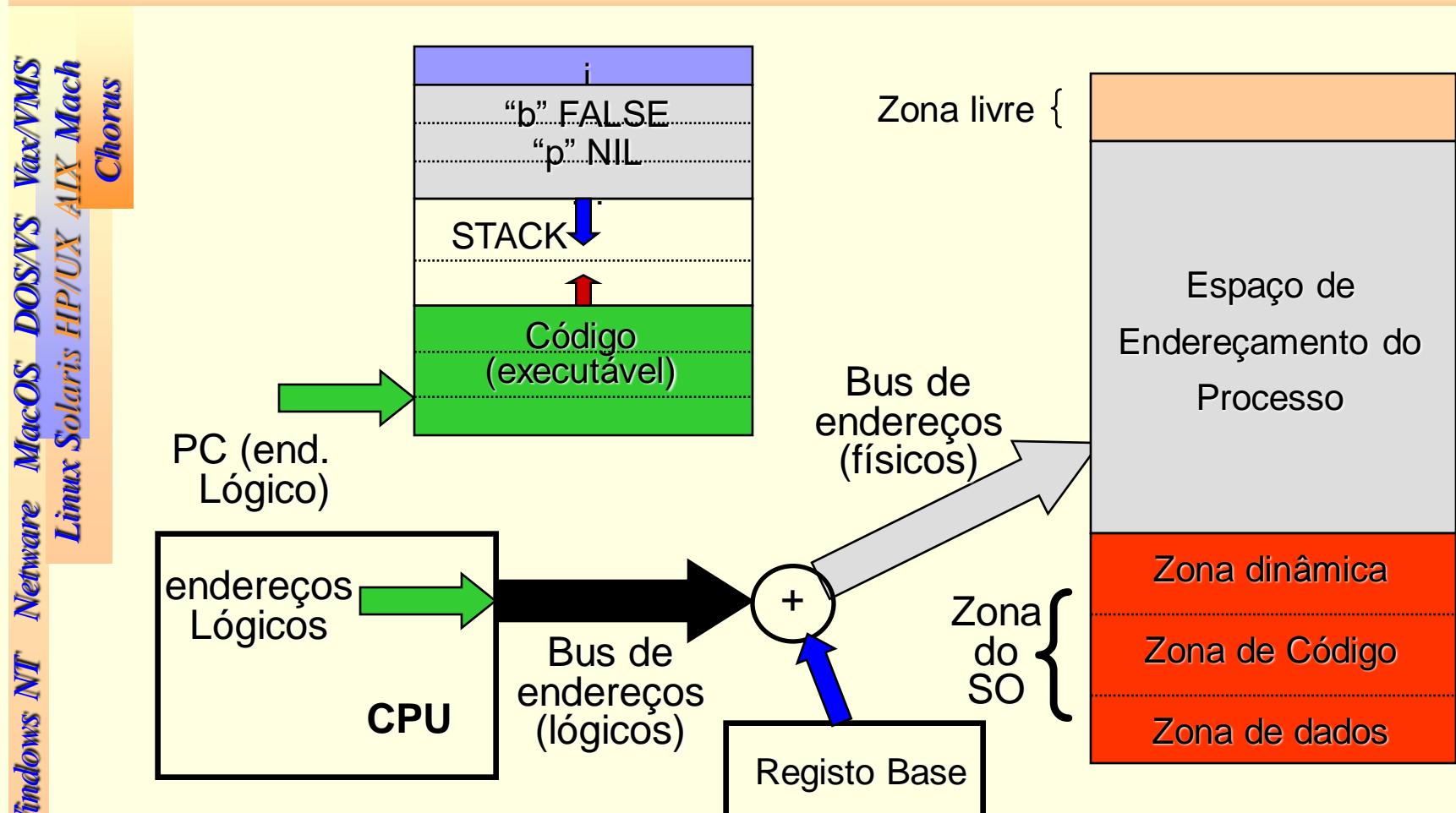
Gestão de Memória por Partição única



Gestão de Memória por Partição única

- Vantagens:
 - Algoritmo de GM muito simples: ler cabeçalho do programa executável para obter dimensão do EE; se dimensão <= RAM livre, carregar e executar; senão, mensagem de erro.
- Desvantagens:
 - Se a dimensão da zona ocupada pelo SO muda, os programas exe têm de ser re-criados (*link* de novo).
 - Nada a fazer quando o executável não cabe na RAM livre.
- Questão
 - Como proteger a zona do SO de um processo errado ou “mal comportado” (que tenta escrever e/ou ler na zona do SO)?

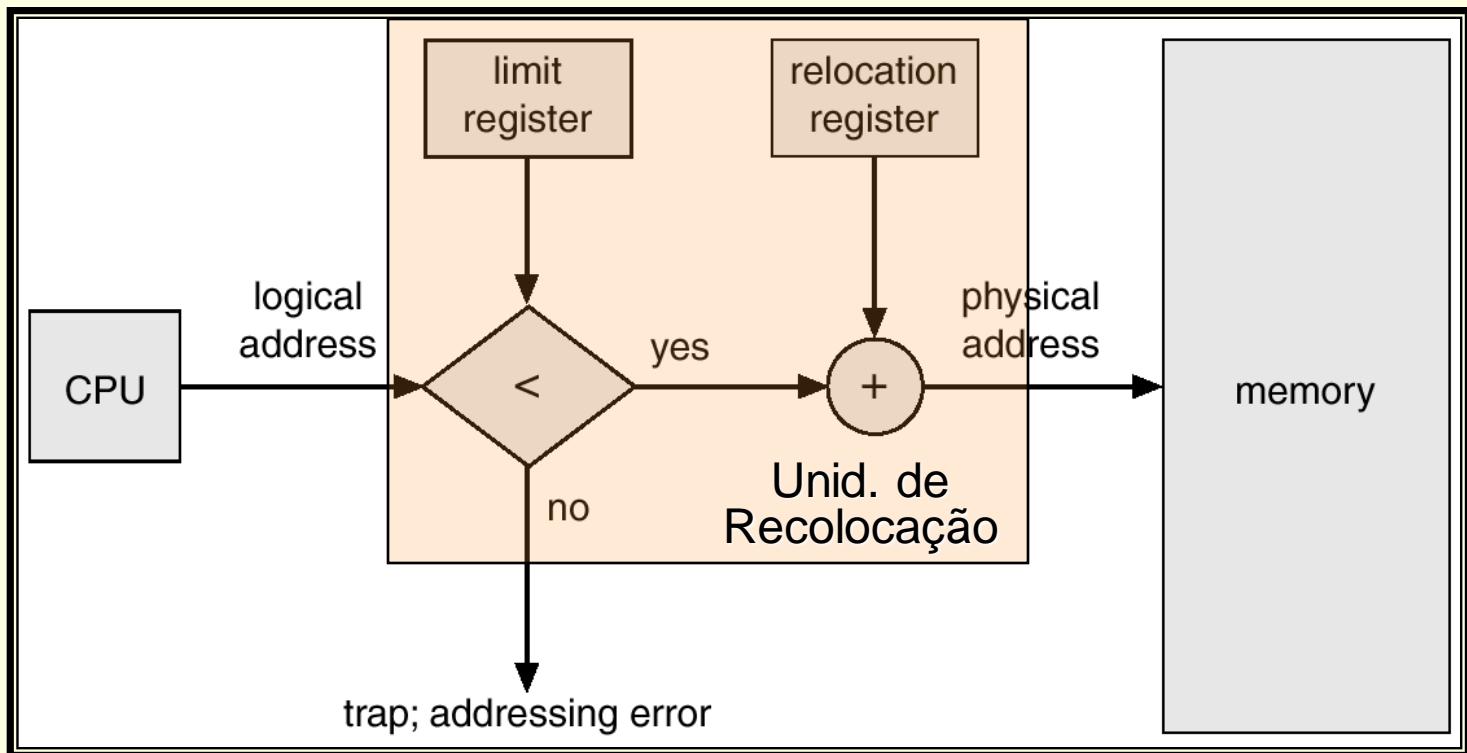
GM com Recolocação: Registo Base



GM com Recolocação: Registo Base

- Espaço de Endereçamento Lógico (ou virtual):
 - O CPU vê/referencia os endereços tal qual como estes se apresentam no espaço do processo – endereços lógicos.
- Unidade de Recolocação:
 - Adiciona um *valor*, guardado num registo chamado de *recolocação* ou *base* a cada endereço “emitido” pelo CPU.
- Espaço de Endereçamento Físico (ou real):
 - O espaço dos endereços efectivamente “comunicados” à memória.

GM com Registo Base e Registo Limite



Fonte: OSC

GM com Registo Base e Registo Limite: vantagens

- Proteger o Espaço de Endereçamento Físico:
 - O Limite – endereço “mais alto” do espaço de endereçamento é guardado num registo limite (RL) da UR.
- Unidade de Recolocação de Endereços:
 - Se o endereço lógico é superior a RL, não efectuar a tradução e gerar uma interrupção.
 - Senão, gerar o endereço físico por adição do *valor base* ao endereço “emitido” pelo CPU.
- Protege o EE dos outros processos (nas técnicas que iremos estudar em seguida) e do SO; ajuda o *debugging*!

GM com Registo Base e Registo Limite: desvantagens

□ Acesso a memória mais lento:

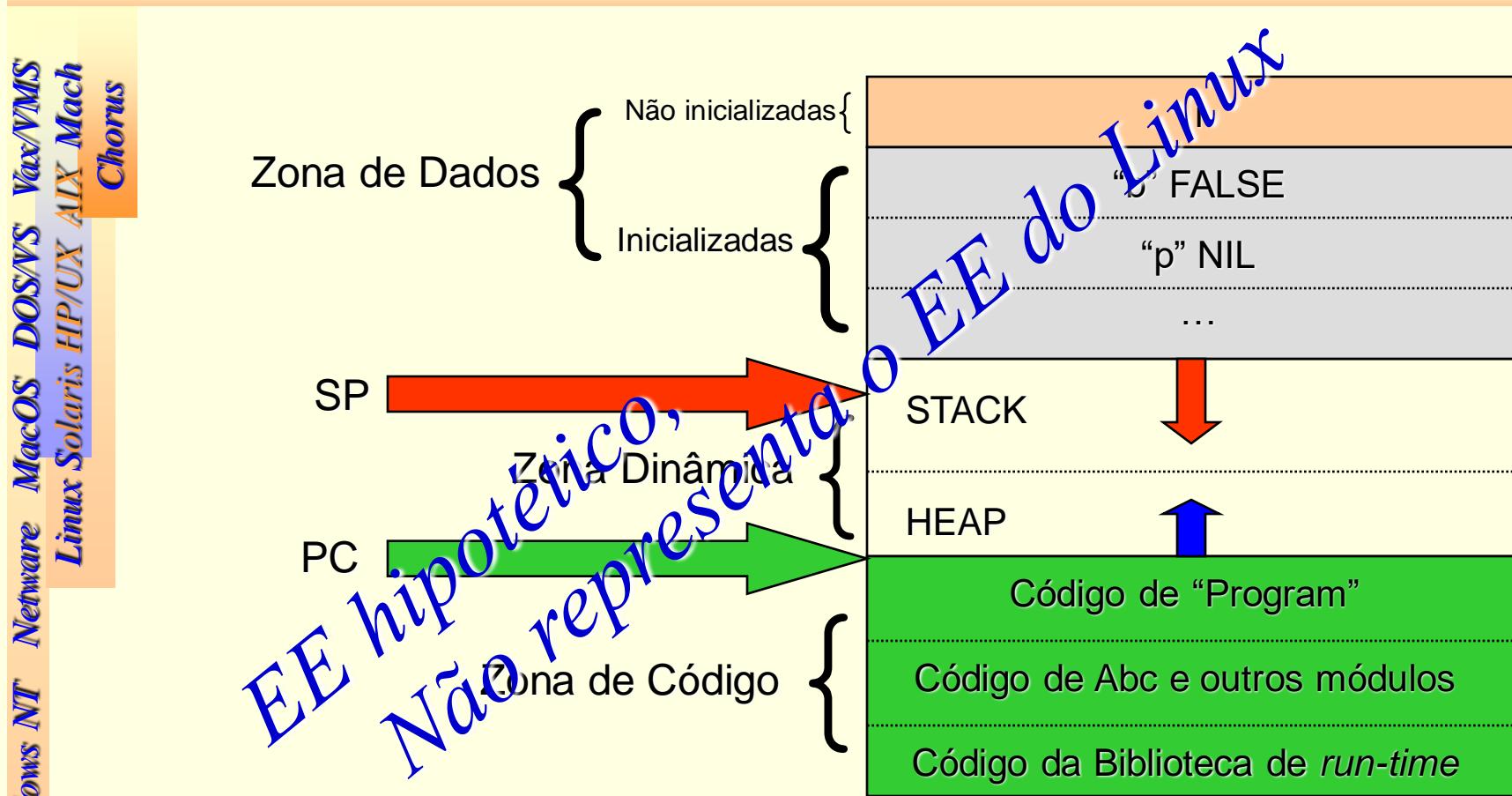
- A Unidade de Recolocação leva algum tempo a comparar o valor do endereço emitido pelo CPU com o RL,
- E leva mais algum tempo a fazer a adição do RB para gerar o endereço físico.
- Mas, sendo essas operações MUITO simples de implementar (em termos de circuitos electrónicos, ou gates), o acréscimo de tempo pode ser muito pequeno face ao tempo total gasto num acesso à memória.

Fundamentos de Sistemas de Operação

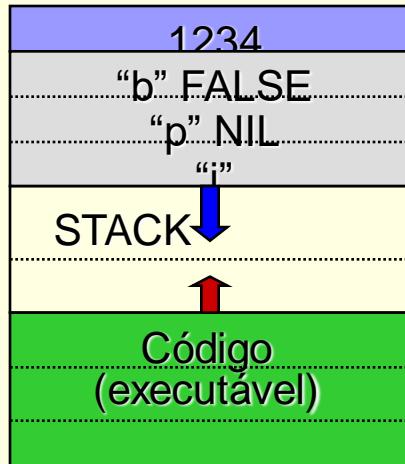
Unix Windows NT Netware Mac OS DOS/V/S Vax/VMS
Linux Solaris HP/UX AIX Mach Chorus

Evolução das técnicas de Gestão de Memória
b) Partições fixas e contíguas

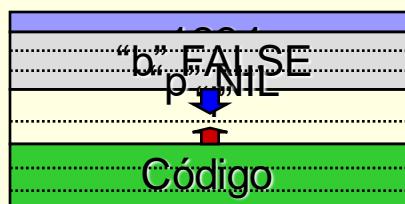
Como colocar em RAM o Processo?



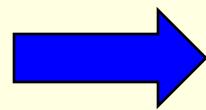
GM por Partições fixas, contíguas



Programa A



Programa B



Não cabe!!!



Número e dimensão das partições definido no "boot" do Sistema!

GM por Partições fixas, contíguas

□ Benefícios:

- Multiprogramação – vários processos em memória, podem ser “simultaneamente” executados pelo CPU.

□ Estratégia de Colocação:

- **Best Fit:** programa colocado na partição “mais ajustada” à sua dimensão.
- **Worst Fit:** programa colocado na maior partição livre!
- **First Fit:** programa colocado na primeira partição que se sabe livre e de dimensão suficiente!
- Na GM por partições fixas usava-se sempre a estratégia Best Fit.

GM por Partições fixas, contíguas

- Desvantagens:
 - Nada a fazer quando o exe não cabe, ou quando nº de processos é superior ao nº de partições.
 - Nº e dimensão das partições definido no arranque do SO; para alterar era preciso fazer shutdown e reboot.
 - Fragmentação interna – memória que fica livre no “interior” de cada partição não pode ser usada para nada!
- Suporte hardware necessário:
 - UR com RB e RL.
 - Carregamento da UR é uma instrução privilegiada (Pense: porquê?)

GM por Partições fixas, contíguas

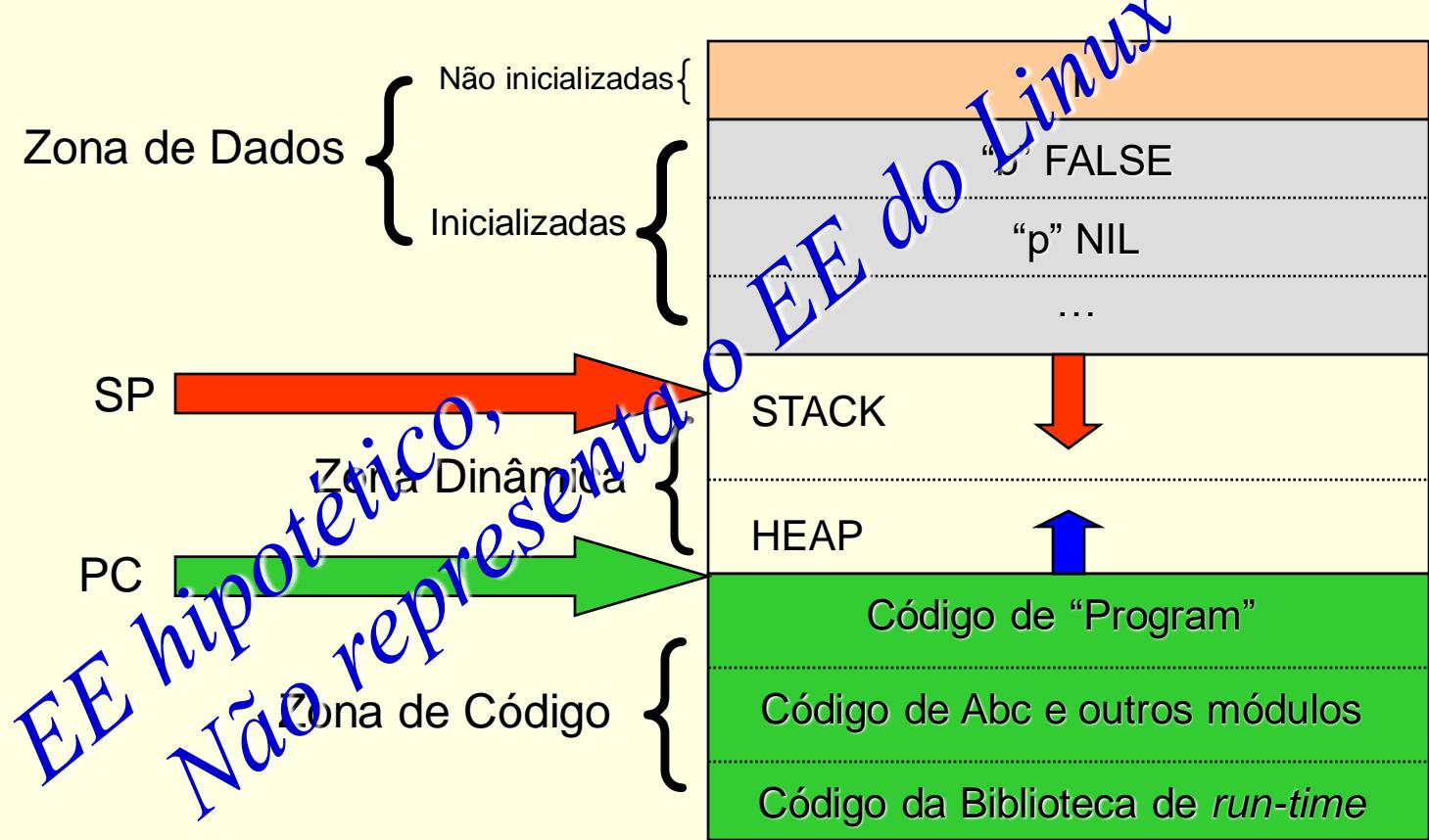
- Estruturas de dados para a GM por P. Fixas:
 - Tabela com uma linha por partição para manter: indicação da dimensão da partição; RB e RL do processo que ocupar a partição ($RB=RL=0$, partição livre);...
- Algoritmo para a GM por Partições Fixas:
 - Estratégia de colocação Best fit!
- Escalonamento dos Processos – um exemplo:
 - Há processos “em fila” à espera de “entrar na memória”
 - Sempre que há uma partição livre, procurar o mais antigo que cabe nessa partição, e carregá-lo!

Fundamentos de Sistemas de Operação

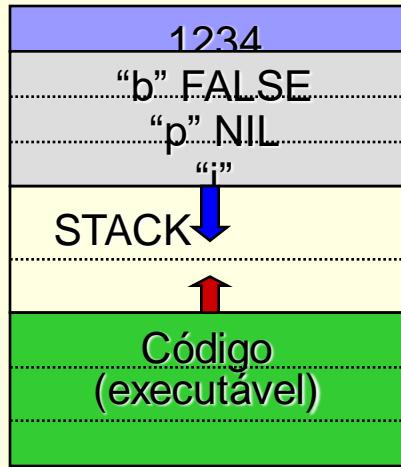
Unix Windows NT Netware Mac OS DOS/V/S Vax/VMS
Linux Solaris HP/UX AIX Mach Chorus

Evolução das técnicas de Gestão de Memória
c) Partições dinâmicas, contíguas

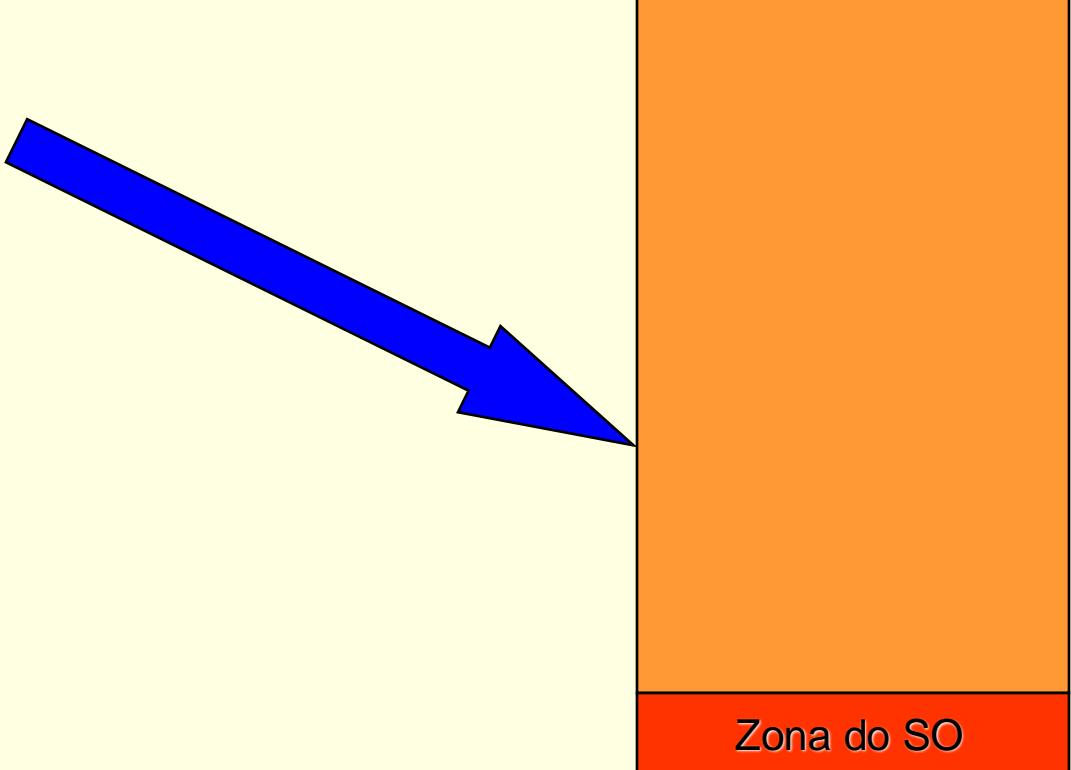
Como colocar em RAM o Processo?



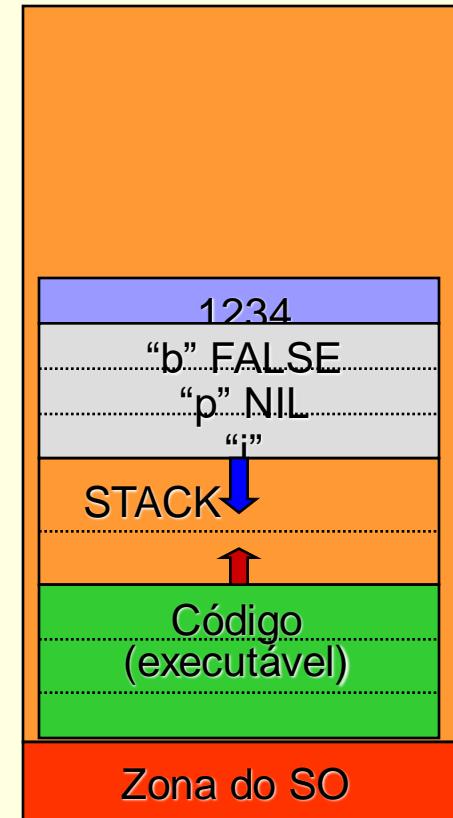
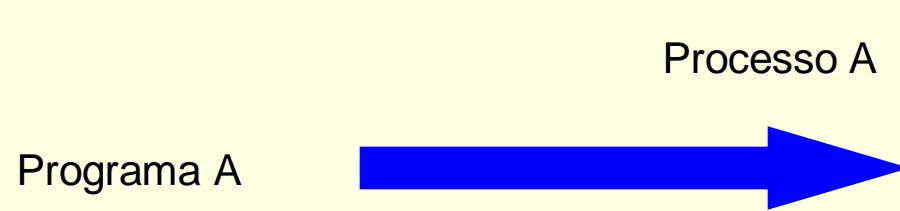
Partições dinâmicas (contíguas): I



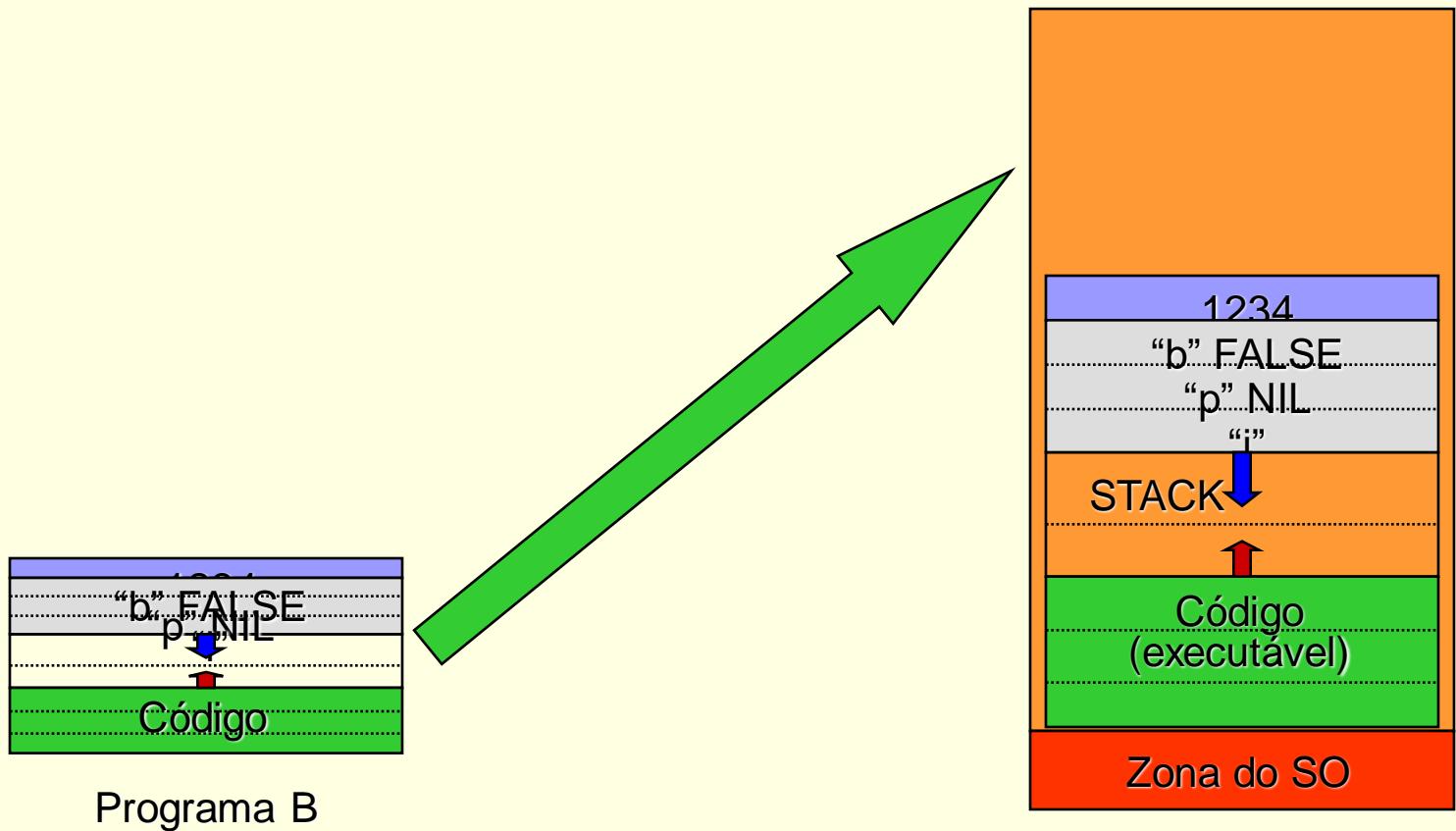
Programa A



Partições dinâmicas (contíguas): II

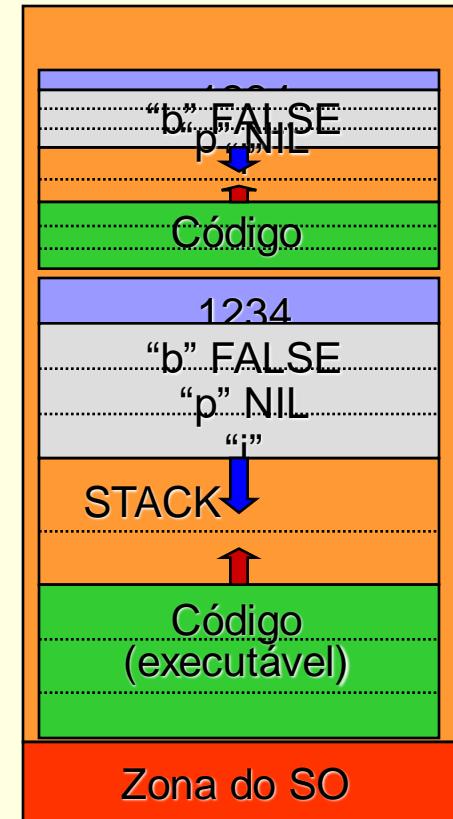


Partições dinâmicas (contíguas): III



Partições dinâmicas (contíguas): IV

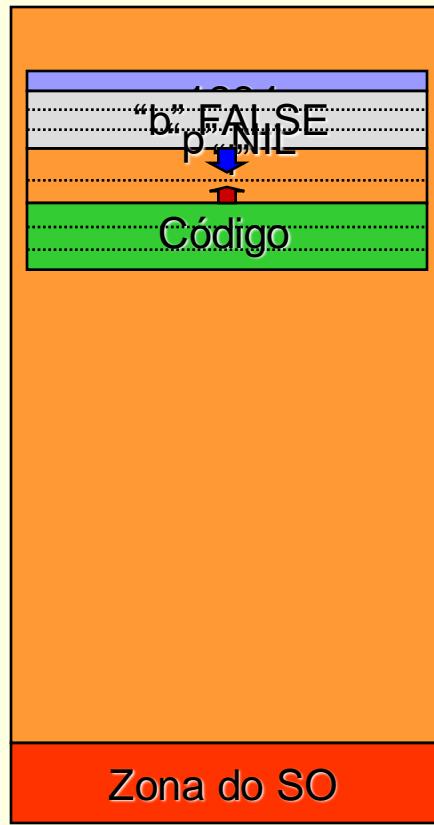
Zona livre
Processo B
Processo A



Partições dinâmicas (contíguas): V, VI

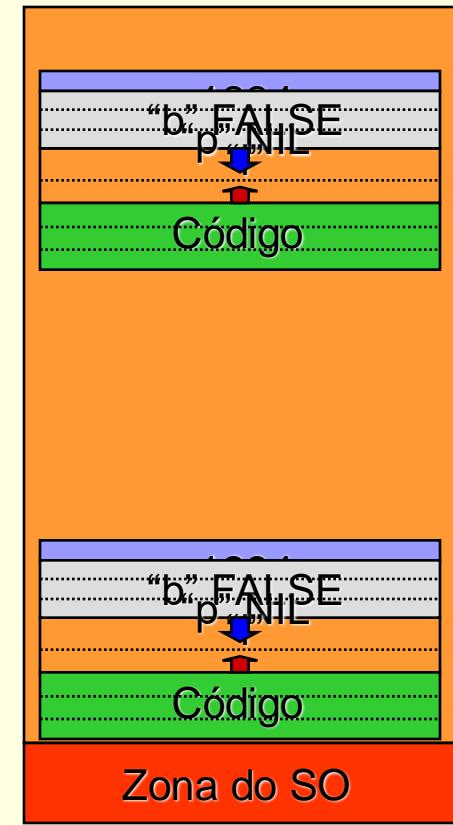
Zona livre
Processo B

Processo A termina

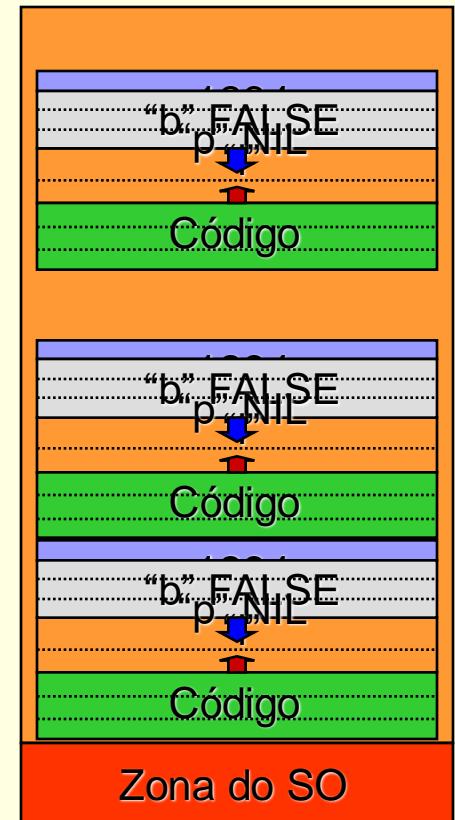


Zona livre
Processo B

Processo C



Partições dinâmicas (contíguas): VII



GM por Partições dinâmicas contíguas

□ Benefícios:

- Multiprogramação – vários processos em memória, podem ser “simultaneamente” executados pelo CPU.

□ Estratégia de Colocação:

- **Best Fit:** programa colocado na região livre “mais ajustada” à sua dimensão.
- **Worst Fit:** programa colocado na maior região livre!
- **First Fit:** programa colocado na primeira região que se sabe livre e de dimensão suficiente!

GM por Partições dinâmicas contíguas

□ Desvantagens:

- Nada a fazer quando exe não cabe.
- Fragmentação externa – “fragmentos” livres de memória que ficam entre regiões ocupadas e não têm dimensão suficiente para albergar programas que se querem correr. (este é um argumento em favor do Worst Fit, que tende a deixar fragmentos grandes)
- Má GM se a fragmentação for excessiva.

□ Suporte hardware necessário:

- UR com RB e RL.
- Carregamento da UR é instrução privilegiada.

GM por Partições dinâmicas contíguas

- Solução: Operações de compactação reduzem o problema da fragmentação externa.
 - Juntar a memória livre num bloco único.
 - Operação feita em tempo de execução; só possível com recolocação dos programas (registos base e limite).
 - Exige muitas operações de I/O e provoca a paragem temporária de todos os programas envolvidos (pode haver swap out/in associado: colocar uma cópia da zona de memória em disco – swap out; libertar, compactar, e trazer de novo de disco para memória – swap in).

GM por Partições dinâmicas contíguas

- Estruturas de dados para a GM:
 - Lista ligada (?) para gerir as regiões livres, ordenada por dimensão da região (?)
 - Tabela (?) com uma linha por processo activo para manter: indicação da dimensão da região; RB e RL do processo;...
 - Que tal lembrar-se de alguns algoritmos que estudou em noutras UCs?
- Algoritmos para a GM:
 - Estratégia de colocação First, Best ou Worst fit
 - Best fit geralmente não usada: tende a criar numerosos pequenos fragmentos.