

Gerenciamento de memória

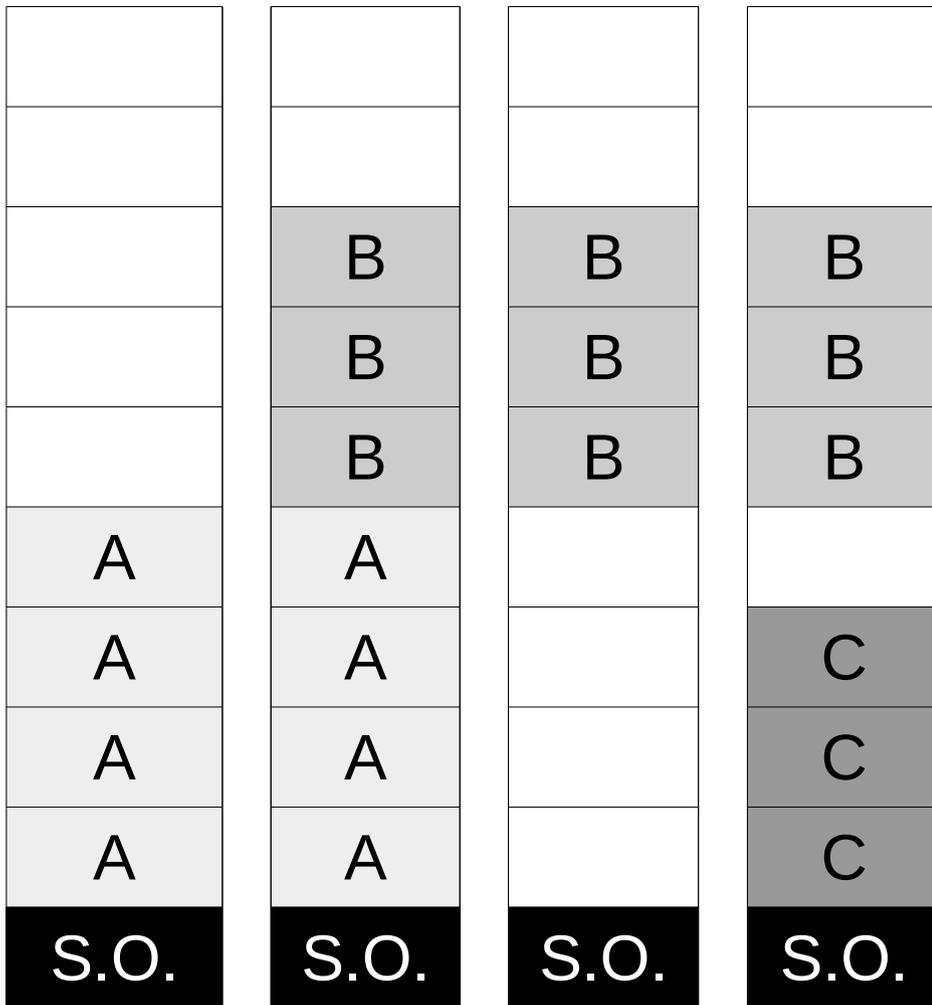
Gerenciamento do espaço livre

Espaço livre

- Como vimos na aula anterior quando é usada alocação dinâmica o S.O. precisa saber onde cada processo está na memória
- Além disso o processo precisa saber quais partes da memória não tem nenhum processo (áreas livres)
- É preciso saber o tamanho de cada área livre para que se possa alocar novos processos à medida que eles são criados



Áreas livres



- À medida que os processos são alocados e liberados da memória, áreas livres vão sendo criadas
- O S.O. precisa conhecer onde começam e qual o tamanho dessas áreas

Tabela de blocos livres

- Considere uma memória de 100 bytes
- Como manter a informação sobre espaço livre e ocupado?
- **Essa informação precisa ficar dentro da memória!**
- Portanto, precisa ocupar o mínimo de espaço possível.

Memória

99								
98								
9								
8								
7								
6								
5								
4								
3								
2								
1								
0								
	0	1	2	3	4	5	6	7

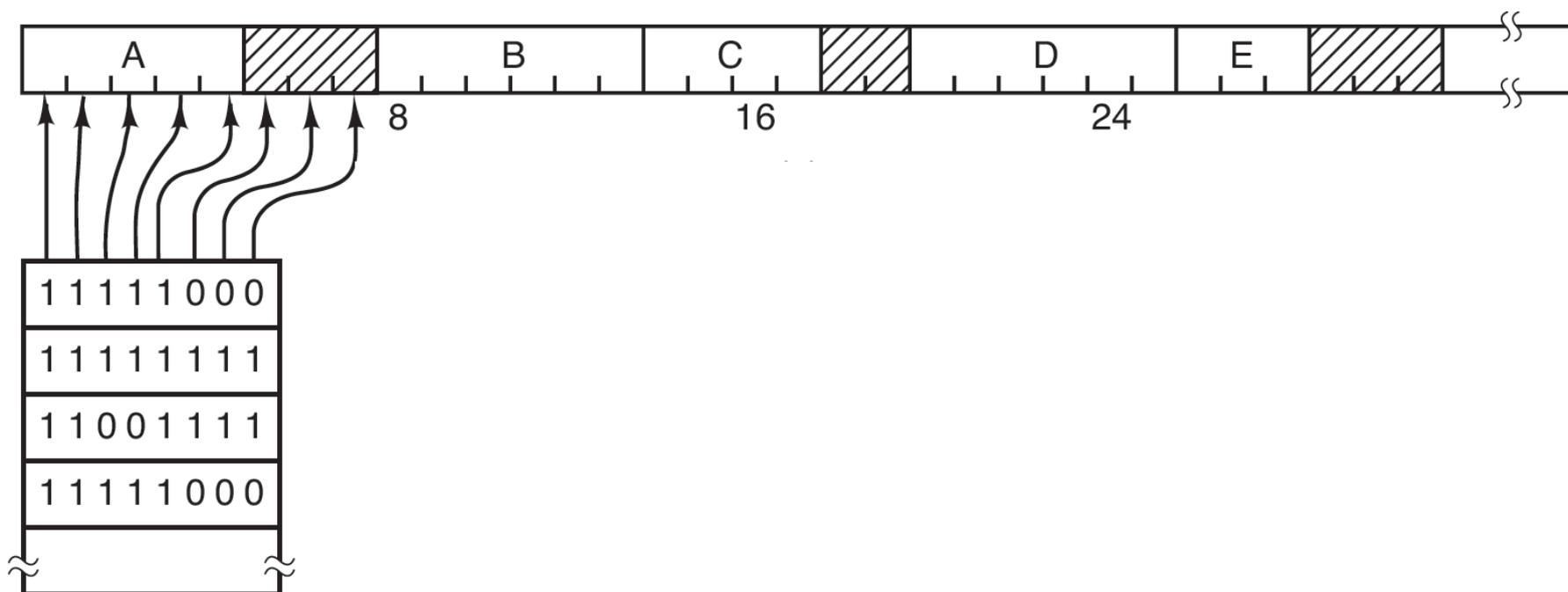


Mapa de áreas livres

- Uma estratégia simples para implementar esta tabela de áreas livres é o mapa de bits.
- Para cada área de tamanho X bytes use 1 bit para informar se está ocupada(1) ou livre(0).
- O consumo de espaço para o mapa é fixo e proporcional ao tamanho da memória.
- Quanto maior X maior o possível desperdício de memória.
- Quanto menor X maior o espaço ocupado pelo mapa
- O tempo para achar uma área livre de um determinado tamanho pode ser muito grande.



Mapa de bits

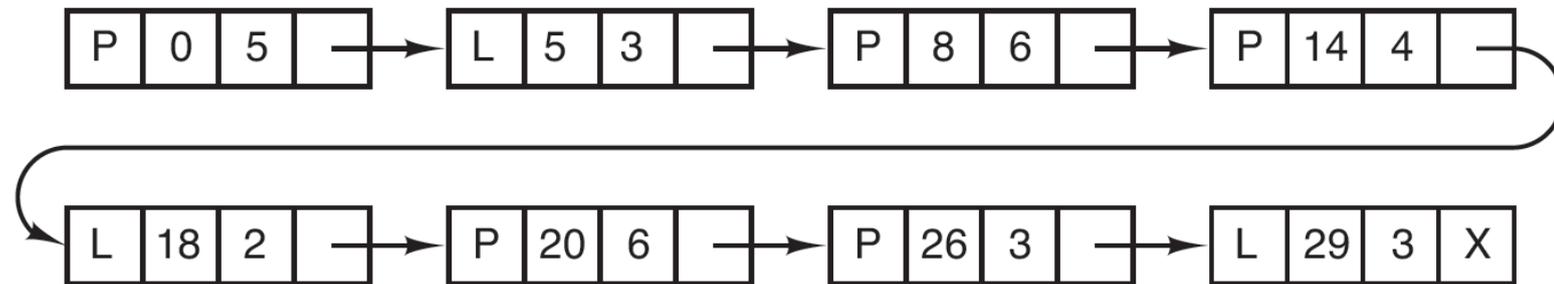
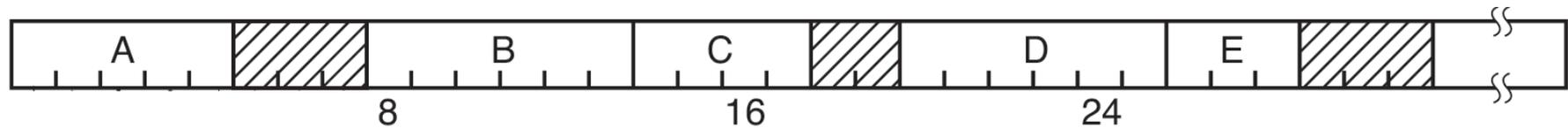


Lista ligada de espaços livres

- Manter uma lista contendo quais os espaços que estão livres e quais estão ocupados e o seu tamanho.
- Na lista teremos também o tamanho do espaço livre
 - Isso pode melhorar o tempo para achar um espaço livre de tamanho determinado
- O tamanho da lista depende do número de processos.



Lista ligada



Segmento de memória
Começa em 18
Comprimento 2

Processo

(c)

Escolha do local

- Uma vez que temos uma lista de locais possíveis para alocar um processo
- Como escolher?
- Algumas alternativas:
 - First Fit (Primeiro que encaixe)
 - Best Fit (Melhor encaixe)
 - Worst Fit (Pior encaixe)
 - Quick fit (Encaixe rápido)



Escolha do local

- First Fit (Primeiro que encaixe)
 - Irá alocar o processo no primeiro espaço livre da lista
- Best Fit (Melhor encaixe)
 - Irá percorrer toda a lista para encontrar o espaço que desperdiçará o mínimo possível
- Worst Fit(Pior encaixe)
 - Irá percorrer toda a lista para encontrar o espaço que “desperdiçará” o máximo possível, esse espaço desperdiçado pode ser depois alocado para outro processo
- Quick fit (Encaixe rápido)
 - Cria várias listas de espaços livres agrupadas por tamanhos comuns de alocação



Atividade

- Considere uma memória de 4 Gigabytes que está separada em blocos de 8 bytes, qual deve ser o tamanho do mapa de bits para esta memória?
- O que é fragmentação interna?
- Compare o algoritmo First Fit e Best Fit descreva uma vantagem e uma desvantagem de cada um comparado ao outro.

Endereço de entrega: <https://goo.gl/2WHbt8>