

Sistemas Operacionais - Lista 3

Aluno: Henrique Oliveira da Cunha Franco

- Q1 A alocação contígua de memória é uma técnica em que cada processo é carregado em um único bloco contínuo de memória física. Isso facilita o acesso direto e simples aos dados, porém causa problemas de fragmentação externa e limita a flexibilidade no gerenciamento de memória.
- Q2 A MMU é responsável por traduzir endereços virtuais, gerados pelos programas, em endereços físicos, utilizados pela memória principal. Ela também pode realizar funções como proteção de memória e controle de acesso.
- Q3 O endereço virtual é o endereço utilizado pelos programas em execução, enquanto o endereço físico é aquele onde os dados realmente estão localizados na memória. A MMU é responsável por mapear os endereços virtuais em físicos.
- Q4 Quando não há memória suficiente para todos os processos, o sistema pode utilizar a técnica de swapping, que consiste em transferir temporariamente processos da memória principal para o disco, liberando espaço para que outros possam ser executados. Quando necessário, o processo é trazido de volta à memória.
- Q5 A fragmentação externa ocorre quando há espaços livres na memória, mas que estão distribuídos em blocos pequenos não contíguos, impedindo que um processo maior seja alocado. Uma possível solução é a compactação da memória, que reorganiza os blocos para formar

um espaço contínuo.

Q6 Quando um processo excede sua área de crescimento (como a pilha ou heap), pode ocorrer uma falha de segmentação (segmentation fault), levando à interrupção do processo, a menos que o sistema permita expansão dinâmica de memória.

Q7 As formas de alocação são:

- First-fit: aloca o primeiro bloco da memória que for suficientemente grande.
- Best-Fit: aloca o menor bloco que seja suficientemente grande, minimizando o espaço desperdiçado.
- Worst-Fit: aloca o maior bloco disponível, esperando manter blocos menores disponíveis para futuras alocações.

Q8

a.	Memória	10K	4K	20K	18K	7K	9K	12K	13K
	Processos	5K	10K	15K	8K	3K	7K	6K	
		P_1	P_2	P_3	P_4	P_5	P_6	P_7	

First - Fit

- P_1 (5K): Aloca no 1º bloco suficiente \rightarrow 10K \rightarrow sobra 5K
- P_2 (10K): Bloco 4K não serve, usa 20K \rightarrow sobra 10K
- P_3 (15K): Pula anteriores, entra em 18K \rightarrow sobra 3K
- P_4 (8K): Pula anteriores até 9K \rightarrow sobra 1K
- P_5 (3K): Aloca em 4K \rightarrow sobra 1K
- P_6 (7K): Encaixe perfeito em 7K \rightarrow sobra 0K
- P_7 (6K): Próximo disponível suficiente é 12K \rightarrow sobra 6K

Blocos: 5K, 1K, 10K, 3K, 1K, 1K, 0, 6K

b.

Memoira	10K	4K	20K	18K	7K	9K	12K	13K
Processos	5K	10K	15K	8K	3K	7K	6K	

P_1 P_2 P_3 P_4 P_5 P_6 P_7

Best - Fit

- P_1 (5K): Melhor encaixe = 7K → sobra 2K
- P_2 (10K): Encaixe perfeito no bloco 10K → não sobra
- P_3 (15K): Melhor entre os grandes é 18K → sobra 3K
- P_4 (8K): Melhor encaixe = 9K → sobra 1K
- P_5 (3K): Melhor bloco = 4K → sobra 1K
- P_6 (7K): Melhor bloco = 12K → sobra 5K
- P_7 (6K): Melhor bloco = 13K → sobra 7K

Blocos: 2K, 0, 3K, 1K, 1K, 5K, 7K

Memoira	10K	4K	20K	18K	7K	9K	12K	13K
Processos	5K	10K	15K	8K	3K	7K	6K	

P_1 P_2 P_3 P_4 P_5 P_6 P_7

Worst - Fit

- P_1 (5K): Maior bloco = 20K → sobra 15K
- P_2 (10K): Maior atual = 18K → sobra 8K
- P_3 (15K): Maior atual = 15K → sobra 0K
- P_4 (8K): Maior atual = 13K → sobra 5K
- P_5 (3K): Maior atual = 12K → sobra 9K
- P_6 (7K): Maior atual = 10K → sobra 3K
- P_7 (6K): Maior atual = 9K → sobra 3K

Blocos: 15K, 8K, 0, 5K, 9K, 3K, 3K