

Disciplina: Sistemas Operacionais
Curso: Análise e Desenvolvimento de Sistemas
Turno: Noite
Professor: Me. José Paulo Lima
Data de entrega: 09/07/2025
Aluno(a): Vinícius Henrique Costa Soares
Matrícula: 20232ADSGR0276

1 - Um sistema de troca de processos elimina lacunas na memória via compactação. Ao supor uma distribuição aleatória de muitas lacunas e diversos segmentos de dados e um tempo de leitura/escrita de 10 ns para uma palavra de memória de 32 bits, quanto tempo ele levará para compactar 128 MB?

Dados:

- 128MB = 2^{27} bytes = 134.217.728 bytes
- 1 ns = 10^{-9} s

R: $10\text{ns} = 4 \text{ bytes} \rightarrow 4x = 2^{27} * 10 * 10^{-9} \rightarrow x = \rightarrow x \approx 1,34 \text{ segundos}$

2 - Qual a função do **gerenciador de memória**?

R: Gerenciar as memórias de forma eficiente e manter o controle das partes que estão livres ou alocadas, e liberá-las quando terminam.

3 - Defina o **espaço de endereçamento** para um processo. Cite dois **espaços de endereçamento virtual**.

R: É o conjunto de endereços que um processo pode usar para endereçar memória.

Endereçamento virtual: identificadores de dispositivos de E/S, UUIDS.

4 - O que representam os **Registradores de Base** e o **Registrador de Limite**?

R: O de base guarda o endereço físico inicial do programa, e o limite o comprimento do mesmo.

5 - O que é a compactação de memória? Como funciona? Qual o benefício alcançado?

R: A técnica de tentar reduzir ao máximo espaços que seriam inutilizados na memória, para aumentar sua disponibilidade. Funciona identificando quais trechos da memória podem ser agrupados, evitando os espaços vazios. Como mencionado, mais memória é tida como disponível.

6 - Fale sobre as **memórias primárias e secundárias**, destacando sua capacidade de armazenamento e velocidade. Após isso, descreva como funciona o **swapping** e mostre os benefícios e limitações. O swapping tenta solucionar que problema?

R: As memórias primárias tem capacidade moderada e são muito velozes, e pela velocidade são usadas para guardar dados dos processos, mas por isso também são mais caras. Em contraste, as secundárias são mais baratas e possuem bem mais capacidade, porém são mais lentas, então ideais para guardar os programas e arquivos em geral.

O swapping se dá pela ação de copiar os dados do processo para a memória secundária, liberando espaço da primária, e permitindo seu retorno à memória quando necessário. Como o tamanho da memória secundária é bem maior, o swapping consegue fazer com que vários dados de processos sejam guardados, porém como contam com velocidade de transferência baixa, haverá essa diferença quando a necessidade de utilizar os dados guardados na memória principal surgir. A técnica tenta justamente lidar com a “sobrecarga” de processos querendo mais e mais memória.

7 - Considere um sistema de troca no qual a memória consiste nos seguintes tamanhos de lacunas na ordem da memória:



Considere agora que chegam sucessivas solicitações de segmentos nos tamanhos de:

- I. 9 KB
- II. 13 KB
- III. 4 KB
- IV. 7 KB
- V. 6 KB

Indique a ordem de segmentos que serão ocupados nos algoritmos de:

(a) First Fit.

R:

3, 8, 5, 13, 2, 20, 7, 6 | 9kb → 13kb

3, 8, 5, 4, 2, 20, 7, 6 | 13kb → 20kb

3, 8, 5, 4, 2, 7, 7, 6 | 4kb → 8kb

3, 4, 5, 4, 2, 7, 7, 6 | 7kb → 7kb (primeiro 7kb)

3, 4, 5, 4, 2, 7, 6 | 6kb → 7kb

(b) Best Fit.

R:

3, 8, 5, 13, 2, 20, 7, 6 | 9kb → 13kb

3, 8, 5, 4, 2, 20, 7, 6 | 13kb → 20kb

3, 8, 5, 4, 2, 7, 7, 6 | 4kb → 4kb

3, 8, 5, 2, 7, 7, 6 | 7kb → 7kb (primeiro 7kb)

3, 4, 5, 4, 2, 6 | 6kb → 6kb

(c) Worst Fit.

R:

3, 8, 5, 13, 2, 20, 7, 6 | 9kb → 20kb

3, 8, 5, 13, 2, 11, 7, 6 | 13kb → 13kb

3, 8, 5, 2, 11, 7, 6 | 4kb → 11kb

3, 8, 5, 2, 7, 7, 6 | 7kb → 8kb

3, 1, 5, 2, 7, 7, 6 | 6kb → 7kb (primeira ocorrência)

(d) Next Fit.

R:

3, 8, 5, 13, 2, 20, 7, 6 | 9kb → 13kb (ponteiro na 4 pos.)

3, 8, 5, 4, 2, 20, 7, 6 | 13kb → 20kb (ponteiro na 6 pos.)

3, 8, 5, 4, 2, 7, 7, 6 | 4kb → 7kb (primeira ocorrência, ponteiro na 6 pos.)

3, 8, 5, 4, 2, 3, 7, 6 | 7kb → 7kb (ponteiro na 7 pos.)

3, 8, 5, 4, 2, 3, 7, 6 | 6kb → 6kb