**Atividade**

1. **Qual a função do escalonador no SISOP(Sistemas Operacional)?**

Tem como principal função implementar os critérios da política de escalonamento.

1. **Quais os parâmetros a serem considerados na avaliação de um *scheduler* (escalonador)?**

Define qual dos processos prontos deve alocar a UCP para execução.

Cada SO necessita de um algoritmo de escalonamento adequado a seu tipo de processamento.

Considerar características dos processos e o tipo do sistema são aspectos fundamentais na escolha do escalonador.

1. **O que são algoritmos de escalonamento preemptivo? Cite um exemplo.**

O sistema operacional pode interromper um processo em execução e passá-lo para o estado de pronto. Por exemplo, ao realizar uma operação de E/S

1. **O que são algoritmos de escalonamento não-preemptivo? Cite um exemplo.**

Um processo ganha o direito de utilizar a UCP e nenhum outro processo pode lhe tirar este recurso. Quando um processo em execução não pode ser interrompido pelo S.O.

1. **Explique o que *Turnaround* e *throughput*.**

Turnaround é o tempo transcorrido entre o momento em que o software entra e o instante em que termina sua execução e throughput são os números de processos terminados por unidade de tempo.

1. **Cite os algoritmos de escalonamento estudados, descrevendo seu funcionamento, bem como, suas principais vantagens e desvantagens.**

FIFO: Uma vez que o processo está com a CPU ele é executado até a sua conclusão. Sendo simples de implementar já que é necessário apenas uma fila.

SJF: Tarefas com menor tempo de utilização são executadas por primeiro. Reduz o tempo médio de espera em relação ao FIFO.

RR: Fatias de tempo são atribuídas a cada processo em partes iguais e em ordem circular, manipulando todos os processos sem prioridade. Não existe a possibilidade de monopólio da UCP

1. **Explique o que é *quantum*.**

É um tempo dado para a execução de cada processo.

1. Calcule o tempo médio de execução e de espera dos processos abaixo. Considerando as seguintes políticas de escalonamento: FIFO, SJF e RR; considerando a ordem de chegada correspondente ao nº do processo e que o *quantum* é igual a 4. Compare os tempos de execução obtidos.

| **Processos** | **Tempos de execução** |
| --- | --- |
| **P1** | 12 |
| **P2** | 3 |
| **P3** | 8 |
| **P4** | 6 |
| **P5** | 2 |

Tempo Médio de execução: Tempo de espera:

FIFO- 15,8 15,8

SJR- 7,4 7,4

RR- 7,4 15,2

1. Calcule o tempo médio de execução e de espera dos processos abaixo. Considerando a política de escalonamento por prioridade.

| **Processos** | **Tempos de execução** | **Prioridade** | **Tempo de chegada na Ready** |
| --- | --- | --- | --- |
| **P1** | 8 | 2 | 1 |
| **P2** | 3 | 1 | 0 |
| **P3** | 3 | 2 | 2 |
| **P4** | 2 | 0 | 1 |
| **P5** | 1 | 3 | 0 |

Tempo Médio de execução: 6,2 Tempo de espera: 6,6

**Atividade** - **Gerência de Memória**

1. **Comente como funciona a alocação contígua simples?**

A memória principal é dividida em duas partes, uma para o SO e a outra para o programa do usuário.

1. **Comente como funciona a alocação particionada?**

Permite que vários programas estejam em memória ao mesmo tempo.

1. **Diferencie alocação particionada estática de alocação dinâmica?**

Na alocação estática, o espaço de memória é definido durante o processo de compilação, já na alocação dinâmica o espaço de memória é reservado durante a execução do programa.

1. **O que é fragmentação interna e externa? Quando elas ocorrem?**

Fragmentação interna é a perda de espaço dentro de uma área de tamanho fixo. Numa memória secundária, ela ocorre quando um arquivo ou fragmento de arquivo não ocupa completamente o espaço da unidade de alocação destinado a ele.

fragmentação externa ocorre quando se desperdiça memória fora do espaço ocupado por um processo. O processo não será executado devido ao esquema que a memória é gerenciada, mesmo que exista memória total livre disponível

1. Explique os algoritmos *best-fit*, *worst-fit* e *first-fit*.

**Best-fit-** Escolhe a melhor partição, ou seja, aquela em que o programa deixa o menor espaço sem utilização.

**Worst-fit-** Escolhe a pior partição, ou seja, aquela em que o programa deixa o maior espaço sem utilização.

**First-fit-** Escolhe a primeira partição livre com tamanho suficiente para alocar o programa.

1. Considere uma situação em que existem 4 partições livres, em um ambiente multiprogramado com partições fixas de memória. Essas partições possuem, respectivamente, 20 Mb, 5 Mb, 8 Mb e 8 Mb. Neste momento existem 4 processos que desejam executar, necessitando as seguintes quantidades de memória: 15 Mb, 2 Mb, 9 Mb e 10 Mb. Após fornecer a partição de 20 Mb ao processo que precisa de 15 Mb e a partição de 5 Mb ao processo que precisa de 2 Mb, calcule a perda de memória com fragmentação externa e interna e a perda total de memória.

Perda de memória com fragmentação externa: 3MB

Perda de memória com fragmentação interna: 8MB

Perda total de memória: 11MB



1. Considere que os processos da tabela a seguir estão aguardando para serem executados e que cada um deles permanecerá na memória durante o tempo especificado. O SO gerencia a memória através da técnica de partição dinâmica e ocupa uma área de 20 Mb no início da memória. A memória total é de 64 Mb; apresente o esquema de alocação dos processos e calcule a perda gerada por fragmentação.

| Processo | Memória | Tempo |
| --- | --- | --- |
| 1 | 30 Mb | 5 |
| 2 | 6 Mb | 10 |
| 3 | 36 Mb | 5 |
| 4 | 15 Mb | 15 |
| 5 | 18 Mb | 10 |

**Processo 1- 19mb**

**Processo 2 - 43mb**

**Processo 3 - 13mb**

**Processo 4 - 34mb**

**Processo 5 - 31mb**

