

## LISTA TEÓRICA

1-Trata-se da abordagem utilizada pelo sistema operacional para gerenciar o processador e tornar possível a multiprogramação do sistema operacional.

2-O **escalonador** é uma rotina do sistema operacional que tem como principal **função** implementar os critérios da política de **escalonamento**. O **dispatcher** é responsável pela troca de contexto dos processos após o **escalonador** determinar qual processo deve fazer uso do processador.

3- São 5 critérios: tempo de espera, tempo de resposta, tempo de turnaround, tempo de cpu e tempo do processador .

4-Tempo de processador: **é o tempo que um processo leva no estado de execução durante seu processamento.** -Tempo de espera: é o tempo total que um processo permanece na fila durante seu processamento, aguardando para ser executado.

**Tempo de Espera:** é o **tempo** em que o processo fica na fila de pronto.

Turnaround: que é o **tempo transcorrido desde o momento em que o software entra e o instante em que termina sua execução**

tempo de resposta: intervalo entre a chegada ao sistema e início de sua execução; tempo de espera: soma dos períodos em que o processo estava no seu estado pronto.

5-No **escalonamento preemptivo**, o sistema operacional pode interromper um processo em execução e passá-lo para o estado de pronto, com o objetivo de alocar outro processo na UCP. No **escalonamento não-preemptivo**, quando um processo está em execução, nenhum evento externo pode ocasionar a perda do uso do processador.

6-O **escalonamento FIFO** procura primeiro terminar um processo antes de começar outro, enquanto o **escalonamento circular**, reserva um determinado tempo para que o processo circule compartilhando o mesmo momento com outros processos.

7- No escalonamento SJF, o algoritmo de escalonamento seleciona o processo que tiver o menor tempo de processador ainda por executar. Dessa forma, o processo em estado de pronto que necessitar de menos tempo de UCP para terminar seu processamento é selecionado para execução.

O escalonamento por prioridades é um escalonamento do tipo preemptivo realizado com base em um valor associado a cada processo denominado prioridade de execução. O processo com maior prioridade no estado de pronto é sempre o escolhido para execução e processos com valores iguais são escalonados seguindo o critério de FIFO. Neste escalonamento, o conceito de fatia de tempo não existe, conseqüentemente, um processo em execução não pode sofrer preempção por tempo.

8-Preempção por tempo ocorre quando o sistema operacional interrompe o processo em execução em função da expiração da sua

fatia de tempo, substituindo-o por outro processo.

Preempção por prioridade, ocorre quando o sistema operacional interrompe o processo em execução em função de um processo entrar em estado de pronto com prioridade superior ao do processo em execução.

9- Maximizar o número de processos na **memória**, permitir a execução de programas maiores que a **memória** física, compartilhamento de dados na **memória** e proteção da **memória** utilizada por processo e pelo sistema operacional.

10- 10KB

11-A-

PA- 2

PB-4

PC- 4

B-

PA- 4

PB- 8

PC- 6

C-

PA- 0

PB- 0

PC- 0

12-

1- p1- esta executando e p2 e p3 estado de pronto

2- p2 esta executando e p1 e p3 estado de pronto

3- p3 esta executando e p1 e p2 estão terminados

13-

fifo

p1= 40

p2= 60

p3=110

p4=140

ta= 87,5 u.t

sjf

P2

P4

P1

P3

CIRCULAR

P1-40

P2-60

P3-80

P4-100

PRIORIDADES

p1

p2

p4

p3

14-

a- Entre os instantes 24-25 e 59-60

b- Entre os instantes 79-80

## LISTA PRÁTICA 4 - Ambiente Linux

1) Fiz tudo.

2)

Procedimento 1:

a) 20, 0.

b) Sim, 0, -20.

c) O nice muda a prioridade do processo na hora da sua inicialização/execução e o procedimento 11 fez com que a prioridade do processo fosse máxima.

d) Quando queremos que o processo tenha prioridade máxima.

3)

a) 20, 0.

PID -> 3678

b) Houve alteração, 35 e 15.

c) O processo ficou mais "bonzinho", portanto, sua prioridade é menor agora.

d) A principal diferença é que o nice muda a prioridade na hora da execução, por outro lado o renice, não. O renice muda com o processo já em execução, sem contar que ele recebe o PID por parâmetro.