

1.

Marque a(s) alternativa(s) INCORRETA(S).

(V) Não é possível ocorrer starvation em um sistema monoprocessado baseado em escalonamento não preemptivo.

de tarefas

(F) O throughput proporcionado por um escalonador é medido pelo número de vezes que uma tarefa passa pelo processador em um intervalo de tempo.

É possível

(F) Não é possível ocorrer starvation em um sistema multiprocessado baseado em escalonamento preemptivo.

(F) O critério de escalonamento turnaround objetiva maximizar o tempo de execução dos processos do sistema. minimizar

2.

Marque a(s) alternativa(s) CORRETA(S).

(V) No escalonamento baseado em múltiplas filas (com realimentação) as prioridades são dinâmicas..

(F) No escalonamento baseado em múltiplas filas (sem realimentação) os processos curtos tendem a possuir maior timeslice e prioridade.

Não existe prioridade no escalonamento baseado em múltiplas filas (sem realimentação) e processos curtos não possuem um timeslice maior

(F) O sistema operacional Linux utiliza um algoritmo de escalonamento cooperativo, portanto, do tipo preemptivo.

O linux não utiliza escalonamento cooperativo e o escalonamento cooperativo não é preemptivo

(F) O sistema operacional Windows 10 utiliza um algoritmo de escalonamento cooperativo, tal como o Linux.

Somente o windows 3.1 / 3.11 utiliza escalonamento preemptivo

3.

Normalmente, quem tem maior prioridade, processos CPU-Bound ou IO-Bound?
Justifique sua resposta.

Normalmente, os processos IO-Bound possuem a prioridade maior, pois na maior parte das vezes tais processos estão associados a processos interativos, ou seja, processos que interagem com o usuário. Ao priorizar processos IO-Bound, a experiência com o usuário é melhorada e o desempenho dos processos CPU-Bound não é afetado. Tal desempenho não é afetado devido ao fato de que na maioria das vezes os processos IO-Bound não utilizam todo o time-slice disponível, com isso os processos CPU-Bound (normalmente, utilizam todo o time-slice disponível) precisam esperar um tempo insignificante para começarem a ser executados.

4.

O que é processor affinity e quais suas vantagens e desvantagens?

O que é:

O processor affinity é uma funcionalidade que permite que se escolha em quais CPUS um determinado processo será executado.

Vantagens:

Ao utilizar o processor affinity, a velocidade de execução de um processo se torna maior, pois os atrasos gerados pelo deslocamento/atualização da estrutura de dados, exclusão mútua e perda do cache ao transferir o processo inúmeras vezes entre as CPUS disponíveis se torna menor.

Desvantagens:

Se o processor affinity não for utilizado com muita cautela em um processo que possui muitas threads, sua velocidade de execução pode diminuir ao invés de aumentar.

5.

Qual a system call no Linux para alterar a prioridade de um processo. Descreva e explique cada um dos seus argumentos de entrada e valores de retorno, caso existam.

PR (prioridade): 0 (maior prioridade) a 39 (menor prioridade)

$PR = 20 + NI$

NI (valor nice): -20 (maior valor nice) a 19 (menor valor nice)

Comando utilizado para exibir os processos em execução, suas prioridades e seus valores nice:

top

System call do Linux utilizada para alterar a prioridade de um processo:

nice [opções] [comando]

[opções] ->

- n NI ou -NI (aumenta prioridade)
- n ou -n -NI (diminui prioridade)
- adjustment=NI (aumenta prioridade)
- adjustment=-NI (diminui prioridade)
- help (exibe informações sobre a utilização da system call nice)
- version (exibe a versão da system call nice)

[comando] ->

exs:

- chrome
- teams
- vlc
- code
- top
- java
- ...

Diminui a prioridade (padrão):

top (prioridade maior)

$NI = 0$

$PR = 20 + 0$

$PR = 20$ (prioridade padrão)

Retorno: execução do comando top

nice top (prioridade menor)

NI = 10 (valor nice padrão)

PR = 20 + 10

PR = 30

Retorno: execução do comando top

Diminui a prioridade:

nice -4 top ou nice -n 4 top ou nice --adjustment=4

NI = 4

PR = 20 + 4

PR = 24

Retorno: execução do comando top

Aumenta a prioridade:

sudo nice -- 15 top ou sudo nice -n -15 top ou nice --adjustment=-15

NI = -15

PR = 20 - 15

PR = 5

Retorno: execução do comando top

Prioridade mínima:

nice -30 top ou nice -n 30 ou nice --adjustment=30

NI = 19

PR = 20 + 19

PR = 39

Retorno: execução do comando top

Prioridade máxima:

sudo nice --30 top ou sudo nice -n -30 ou sudo nice --adjustment= - 30

NI = -20

PR = 20 - 20

PR = 0

Retorno: execução do comando top