* 1. Multiprogramação: Escalonamento de Processos e Threads

1. Escalonador
   1. Parte do S.O. que escolhe qual processo deve assumir o controle da CPU
   2. Algoritmo de escalonamento
2. Chaveamento de Processos
   1. Procedimento Demorado
   2. Pode levar a uso ineficiente da CPU
3. Comportamento do Processo (Slide 47)
   1. CPU – Bound

3.2. I/O – Bound

1. Quando Escalonar

4.1. Criação de processo (Pai/Filho)

4.2. Término de um processo

4.3. Bloqueio de processo

4.4. Interrupção de E/S (Pronto/anterior)

4.5. Hardware de relógio (timer)

4.5.1. Algoritmo não preemptivo (processo executa até término ou bloqueio)

4.5.2. Processo executa por tempo máximo especificado

5. Categorias de algoritmos (slide 48)

6. Objetivos do algoritmo de escalonamento (slide 49)

7. Algoritmos de escalonamento

7.1. Sistemas em lote

7.1.1. First-come, first-server(FCF5) ou FIRST-IN, FIRST-OUT(FIFO)

7.1.2. Shortest Job First (SJF) (Slide 51) – Minimiza tempo médio de espera

7.1.3. Shortest remainig time next (SJF preemptivo)

8. Sistemas Interativos

8.1. Round-Robin (Circular) (slide 53)

8.1.1. Quantum (ex: 4 ms, 100 ms) Típico hoje em dia: 20 ms a 50ms

8.1.2. Chaveamento de Processo (ex: 1 ms)

8.2. Por prioridades (slide 54)

8.2.1 Starving (aumento de prioridade)

8.2.2. Unix: Comando Nice diminui prioridade (usuários nunca utilizam)

8.2.3. Ideia: I/O – bound prioridade mais alta

Prioridade = 1/F, F=fração do último quantum

8.3. Filas Múltiplas

8.3.1. Classes de prioridades com quanta diferente

8.3.2. Prioridade mais alta, 1 quantum, prioridade mais baixa, vai dobrando

8.3.3. Se processo usa todo seu quantum, baixa prioridade

8.3.4. Ex: 100 quanta: 1, 2, 4, 8, 16, 32, 64 (37)

8.4. Shortest Process Next

8.4.1. Em vez de job, usar tarefa parcial do processo

8.4.2. Usar histórico

8.5. Escolanamento garantido

8.5.1. Dividir tempo de CPU por processo

8.6. Escalonamento por loteria

8.6.1. Cada Processo recebe um ticket, que é sorteado

8.6.2. Um processo pode receber mais de um ticket

8.6.3 Previsível, dependendo do número total de tickets, ao contrário do por prioridades

8.7. Fair-Share

8.7.1 Dividir o tempo de CPU por usuário

AE BE CE DE AE....

ABC DE A BC DE...

ABE CDE ABE..

8.8. Sistemas de tempo-real

8.8.1. Tempo real crítico x tempo real não crítico

8. 8.2. Comportamento dos processos deve ser previamente conhecido

8.8.. Sistema escalonável:

8.3. Algoritmos estáticos x dinâmicos

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Prova 01-2013

Letra A – FIFO

P1 – 12 tempos / P2- 8 tempos/P3 – 5 tempos/P4 - 3 tempos –

12 20 25 28

Finalizado com 28 tempos

Tempos

P1=zero

P2 = 12-2=10

P3=20-4=16

P4=25-7 =18

Médio =( 0+10+16+18)/4=11

Sempre a diferença entre o tempo que foi executado e foi criado.

Letra B – SJF Preemptivo

Ele troca para o que tiver menos tempo para terminar – termina em 28

Tempos

P1=18-2=16

P2=12-4=8

P3=0

P4=9-7=2

Media=(16+8+0+2)/6,5 – tempo de espera médio é o menor de todos

Letra C- Prioridades

O número maior é o de maior prioridade

P1(0)-P3(4)-P4(7)-P3(10)

Encerra em 28

P1=12-4=8

P2=20-2=18

P3=10-7

P4=0

Media(8+18+7+0)/4=7,25