

Sistemas Operacionais

Escalonamento de CPU (cap. 6)

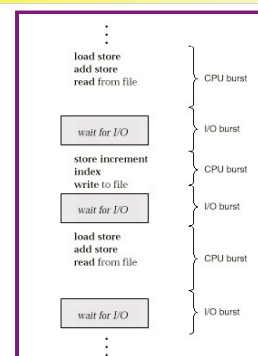
Sumário

- Conceitos básicos
- Critérios usados no escalonamento
- Algoritmos de escalonamento
- Escalonamento em multi-processadores
- Escalonamento em tempo real
- Modelos de escalonamento

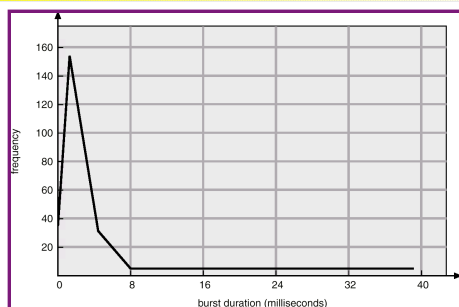
Conceitos básicos

- Objetivo da multiprogramação:
 - Utilização máxima da CPU
- Processos normalmente alternam picos de processamento (uso da CPU) e E/S
- Quando um processo começa um pico de E/S outro deve assumir a CPU para evitar ociosidade
- Por outro lado, nenhum processo deve controlar a CPU indefinidamente p/ melhorar a interação

Alternância de picos de CPU e E/S



Distribuição de picos de CPU



Escalonador da CPU (*scheduler*)

- Controla a mudança de estado dos processos
- Escalonamento ocorre quando um processo:
 - a) chaveia de "em execução" para "em espera"
 - b) chaveia de "em execução" para "pronto"
 - c) chaveia de "em espera" para "pronto"
 - d) termina

Preempção ou não preempção

- Escalonamento não preemptivo
 - Processo só deixa a CPU se tiver que esperar por E/S ou intencionalmente
 - Implementação mais simples do escalonador
- Escalonamento preemptivo
 - Periodicamente o escalonador interrompe o processo em execução e muda-o para “pronto”
 - Escalonador mais complexo
 - Compartilhamento da CPU é garantido

Dispatcher (despachante)

- Módulo responsável por dar o controle da CPU a cada processo no escalonador
 - Troca de contexto de execução
 - Chaveamento para modo usuário
 - Desvio para o ponto apropriado do programa
- Latência de despacho:
 - Tempo gasto para o despachante interromper um processo e iniciar a execução de outro

Crítérios de escalonamento

- Utilização de CPU (mantê-la ocupada ao máx.)
- *Throughput* – taxa de término dos processos
- *Turnaround* – tempo para completar um processo
- Tempo de espera – tempo gasto na fila “pronto”
- Tempo de resposta – tempo entre uma requisição e o início da entrega da resposta em sistemas interativos (de tempo compartilhado)

Crítérios de otimização

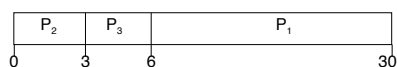
- Maximizar a utilização de CPU
- Maximizar o *throughput*
- Minimizar o tempo de *turnaround*
- Minimizar o tempo de espera
- Minimizar o tempo de resposta

First-Come, First-Served (FCFS)

Processo	Duração
P1	24
P2	3
P3	3

Ordem de chegada: P2, P3, P1

- Gantt Chart do escalonamento:



- Tempo de espera: P1 = 6; P2 = 0; P3 = 3
- Tempo de espera médio: $(6 + 0 + 3)/3 = 3$

First-Come, First-Served (FCFS)

Processo	Duração
P1	24
P2	3
P3	3

Ordem de chegada: P1, P2, P3

- Gantt Chart do escalonamento:



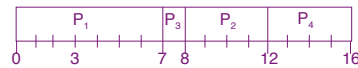
- Tempo de espera: P1 = 0; P2 = 24; P3 = 27
- Tempo de espera médio: $(0 + 24 + 27)/3 = 17$
- Efeito comboio: pequenos atrasados pelo grande

Shortest-Job-First (SJF)

- Associa-se a cada processo seu próximo pulso
 - Próximo processo: o de menor pulso
- SJF preemptivo:
 - Um processo de pulso menor que o que falta do processo corrente, força a preempção do último
 - Chamado *Shortest Remaining Time First* (SRTF)
- SJF não preemptivo:
 - Processos que chegam são comparados apenas com aqueles à espera da CPU
- SJF é ótimo quanto ao tempo médio de espera

SJF não preemptivo

Processo	Chegada	Duração
P1	0	7
P2	2	4
P3	4	1
P4	5	4



Tempo de espera médio = $(0 + 6 + 3 + 7)/4 = 4$

SJF preemptivo

Processo	Chegada	Duração
P1	0	7
P2	2	4
P3	4	1
P4	5	4



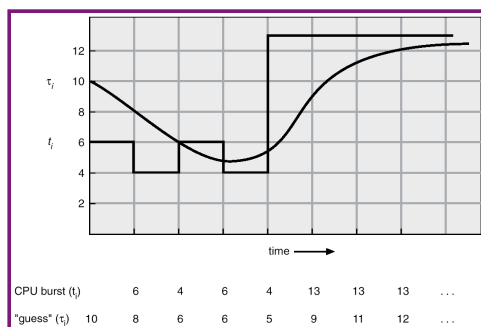
Tempo de espera médio = $(9 + 1 + 0 + 2)/4 = 3$

Problema: duração do próximo pulso

- Só podemos estimar sua duração
 - Pode ser feita por vários tipos de estimadores
 - p.ex., média exponencial:
- t_n = tamanho observado do $n^{\text{ésimo}}$ pico de CPU
 - τ_{n+1} = valor previsto para o próximo pico de CPU
 - $\alpha, 0 \leq \alpha \leq 1$
 - Define-se:

$$\tau_{n+1} = \alpha t_n + (1 - \alpha) \tau_n$$

Previsão da duração do próximo pico de utilização de CPU



Escalonamento com prioridades

- Um valor de prioridade (inteiro) p/ cada proc.
- A CPU é alocada p/ o proc. de maior prioridade
 - Usualmente, menor valor = maior prioridade
 - Preemptivo ou não preemptivo
- SJF é um escalonamento com prioridade, onde a prioridade é o tamanho previsto do pulso de CPU

Escalonamento com prioridades

- Problema: inanição (*starvation*)
 - Processos de baixa prioridade podem nunca ser executados
- Solução: envelhecimento (*aging*)
 - Prioridade de cada processo que fica na fila aumenta à medida que o tempo passa

Round Robin (RR)

- Cada processo recebe a CPU por um tempo pequeno (quantum)
 - Usualmente, de 10 a 100 milissegundos
 - Ao fim do tempo, processo é preemptado e vai para a fim da fila de prontos
 - Preempção periódica é um *overhead*
- quantum muito grande: FIFO
- quantum muito pequeno: *overhead* de preempção se torna proibitivo

Round Robin (RR)

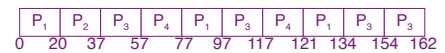
- Se há n processos na fila de prontos:
 - cada processo recebe $1/n$ do tempo da CPU
 - executam por max. um quantum de cada vez
 - nenhum processo espera mais que $(n-1)$ quanta na fila até receber a CPU

Round Robin (RR)

- Exemplo (quantum = 20 unidades de tempo)

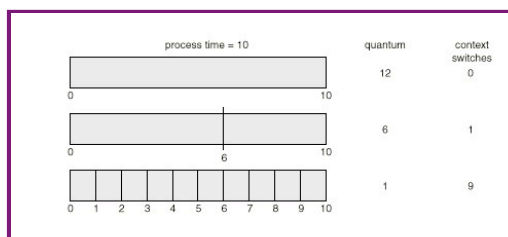
Processo	Duração
P1	53
P2	17
P3	68
P4	24

- Gantt chart:

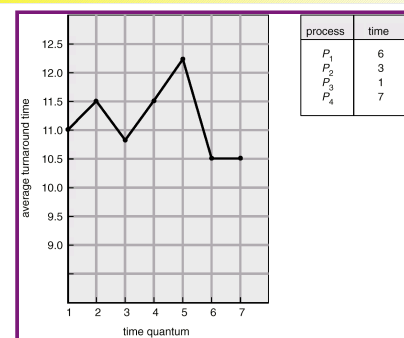


- Usualmente, *turnaround* superior a SJF, mas melhor capacidade de resposta

Quantum e trocas de contexto



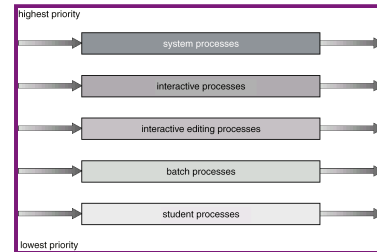
Turnaround varia com o quantum



Escalonamento (filas) multi-nível

- Fila de prontos é dividida:
 - processos interativos (*foreground*) – RR
 - processos em lote/*batch* (*background*) – FCFS
- Escalonamento entre filas
 - Prioridade fixa
 - P.ex.: atender sempre processo interativos primeiro
 - Possibilidade de inanição
 - Fatias de tempo
 - Cada fila é escalonada por uma fração do tempo total

Escalonamento (filas) multi-nível



Filas multinível com realimentação

- Um processo pode se mover entre filas
 - Forma de implementar envelhecimento, p.ex.
- Escalonamento pode ser definido por:
 - no. de filas
 - algoritmo de escalonamento de cada fila
 - método usado para promover um processo
 - método usado para rebaixar um processo
 - método usado para decidir em que fila cada processo entra no sistema

Escalonamento de multi-processadores

- Escalonamento se complica com múltiplas CPUs
 - Coisas acontecem realmente ao mesmo tempo
- Livro só aborda o caso de processadores homogêneos e com acesso uniforme à memória
 - Podem fazer compartilhamento de carga
 - CPUs acessam uma mesma fila
 - Acesso à fila pode gerar contenção

Escalonamento de multi-processadores

Multiprocessamento simétrico (SMP):

- todas as CPUs acessam dados e recursos de forma igual

Multiprocessamento assimétrico (AMP):

- só um processador acessa as estruturas de dados do sistema, reduzindo a complexidade do sistema
 - P.ex.: primeiras versões do Linux multiprocessado

Escalonamento de tempo real

- Processos podem estabelecer demandas em termos de tempo de ativação de tarefas
- Essencial para sistemas de controle de processos de precisão e multimídia

Escalonamento de tempo real

- *Hard real-time* (tempo rígido) – devem completar tarefas em um tempo garantido.
- *Soft real-time* (tempo maleável) – requerem que processos críticos recebam prioridade superior aos demais.

Latência de despacho

- Tempo gasto até que um processo de tempo real que solicita a CPU realmente a receba
- Técnicas podem reduzir essa latência
 - pontos de interceptação: pontos onde operações potencialmente longas podem ser interrompidas
 - kernel interceptável (interrompível)
 - liberação forçada de recursos de processos de baixa prioridade (resolução de conflitos)

Latência de despacho

