



Sistemas Operacionais

CCP/SIF

UNISUL – Tubarão

Cassio Brodbeck Caporal

`cassio{NOSPAM}ostec.com.br`

Agenda

- Revisão;
- Gerenciamento do processador;
- Critérios de escalonamento;
- Escalonamento preemptivo e não-preemptivo;
- Escalonamento FIFO, SJF, Cooperativo, Circular, Prioridades, Múltiplas filas;

Critérios de Escalonamento

- Multiprogramação:
 - Idéia de executar diversos processos simultaneamente concorrente pela utilização de recursos, em especial, a CPU;
- Necessidade de estabelecer critérios para definir o processo que deverá utilizar a CPU:
 - Política de escalonamento;

Critérios de Escalonamento

- Funções de uma política de escalonamento:
 - Manter o máximo possível o processador em uso;
 - Segmentar o uso da CPU entre vários processos;
 - Privilegiar a execução de aplicações críticas;
 - ...

Critérios de Escalonamento

- Quem implementa estes critérios?
 - O tal do ***escalador***.
 - Sem isto, fica difícil imaginar um sistema multiprogramável;
- *Dispatcher*:
 - Elemento fundamental responsável por implementar a troca de contexto de processos.

Critérios de Escalonamento

- A política de escalonamento varia de acordo com o tipo de SO (e suas necessidades);
- Principais critérios utilizados:
 - *Throughput*: número de processos executados em um intervalo de tempo (vazão);
 - Tempo de processador: tempo do processo em seu estado de execução;

Critérios de Escalonamento

- Principais critérios utilizados:
 - Tempo de espera: tempo que um processo fica em estado de pronto aguardando a execução;
 - Tempo de *turnaround*: tempo de criação até o término (inclui tempo para alocação e limpeza de recursos utilizados);
 - Tempo de resposta: tempo decorrente de uma requisição até a apresentação da “resposta”.

Escalonamento preemptivo e não-preemptivo

- Políticas são definidas em preemptiva e não-preemptiva;
- Preempção é a possibilidade de interrupção e substituição da execução de um processo;
- Escalonamento não-preemptivo:
 - Que pesadelo! Processamento *batch*;
 - O processo não perde NUNCA a execução;

Escalonamento preemptivo e não-preemptivo

- Escalonamento preemptivo:
 - É possível interromper a execução de um processo, dando vez a outro;
 - Desta forma é possível priorizar, com base em diversos critérios, processos a serem executados;
 - Maior complexidade, mas todos os SOs utilizados atualmente usam preempção;

Escalonamento FIFO

- Hmm.. *first-in-first-out*;
- O processo que chegar primeiro ao estado de pronto é executado;
- Utilização de uma única fila de escalonamento;
- Implementação bastante simplificada;

Escalonamento FIFO

- Problemas:
 - Identificar quando um processo terá sua execução iniciada;
 - Processos *CPU-bound* levam vantagem no uso do processador com relação a processos *I/O-bound*;
 - É do tipo *não-preemptivo*;

Escalonamento SJF

- Ahnnnn.. *shortest job first*;
- O processo que tiver o **menor** tempo do processador para executar, tem prioridade na fila;
- É considerado, também, não-preemptivo, como uma solução inicial ao FIFO;

Escalonamento SJF

- A vantagem sobre o FIFO é a redução do tempo de *turnaround*;
- Variação com preempção: *shortest remaining time*;
 - Sempre que um processo na fila de pronto tiver menor tempo de processador, o SO faz uma preempção;

Escalonamento Cooperativo

- Objetivo:
 - Aumentar a multiprogramação de algoritmos de escalonamento não-preemptivos;
- Escalonamento “solidário”:
 - Um processo deixa outro utilizar a CPU por livre vontade;
- Interrupção não é reponsabilidade do SO;

Escalonamento Circular

- *Round robing scheduling;*
- Projeto para sistemas de tempo compartilhado;
- Similar ao FIFO, mas cada processo tem sua fatia de tempo (*time slice*) para usar a CPU;
- Conhecido como *preempção por tempo*;

Escalonamento Circular

- Escalonamento democrático:
 - Não permite a monopolização da CPU por um processo;
- Problema?
 - Sempre tem: processos CPU-bound são beneficiados por sua própria estrutura;
 - Balanceamento inadequado do processador;

Escalonamento Circular

- Solução?
 - Escalonamento circular virtual;
 - Processos que saem do estado de espera, utilizam uma fila de pronto **auxiliar**;
 - A fila de pronto auxiliar possui prioridade de escalonamento em relação a fila de pronto;
 - A fila de pronto só é analisada, quando a fila de pronto auxiliar estiver vazia.

Escalonamento por Prioridades

- Escalonamento baseado em prioridade de execução;
- Processo na fila de pronto com maior prioridade é executado, independente de sua posição;
- Mesma prioridade? Usa-se FIFO;

Escalonamento por Prioridades

- Troca de processos só ocorre em uma ação voluntária ou quando um processo de maior prioridade passa para a fila de pronto;
- Onde fica a prioridade?
 - Contexto de software;
 - Prioridades estáticas;
 - Prioridades dinâmicas;

Escalonamento por Prioridades

- Troca de processos só ocorre em uma ação voluntária ou quando um processo de maior prioridade passa para a fila de pronto;
- Onde fica a prioridade?
 - Contexto de software;
 - Prioridades estáticas;
 - Prioridades dinâmicas;

Escalonamento por Prioridades

- Problema?
 - Possibilidade de um processo com prioridade baixa nunca ser executado;
- Solução?
 - *Aging*: processos antigos (idosos) vão ganhando prioridade;
 - Igual fila de banco! :-P

Escalonamento por Prioridades

- Variante: escalonamento circular com prioridades;
 - Utiliza tanto *time slice* quanto prioridade na execução de processos;
 - O processo fica na CPU até desejar sair ou sofrer uma preempção baseada em tempo ou prioridade;

Escalonamento com Múltiplas Filas

- *Multilevel queue scheduling;*
- Várias filas no estado de pronto com prioridades diferentes;
- Processos são associadas às filas de acordo com suas características;
- Importância: usar diferentes algoritmos em diferentes filas;

Escalonamento com Múltiplas Filas

- Desvantagem:
 - Sempre tem!
 - Uma vez que o processo inicia, não é possível trocar sua fila de escalonamento!

Escalonamento com Múltiplas Filas: Realimentação

- Funcionamento similar ao anterior, entretanto, processos podem alternar de fila durante a execução;
- Atende a necessidades de todo tipo de processo: *CPU-bound* e *IO-bound*;
- Quanto mais tempo de execução, com menor prioridade o processo fica;

Escalonamento com Múltiplas Filas: Realimentação

- Nem tudo são rosas...
 - Complexo para implementação;
 - Gera bastante *overhead* pois precisa monitorar diversos aspectos dos processos e alocá-los ou realocá-los dinamicamente em múltiplas filas de pronto.