Sistemas Operacionais

Gerência do Processador

- Com o surgimento dos sistemas multiprogramáveis, onde múltiplos processos poderiam permanecer na memória principal compartilhando o uso da UCP , a gerência do processador tornou-se uma das atividades mais importantes em um SO.

- A partir do momento em que diversos processos podem estar no estado de pronto, devem ser estabelecidos critérios para determinar qual processo será escolhido para fazer uso do processador.

- Os critérios utilizados para esta seleção compõem a chamada “política de escalonamento”, que é a base da gerência do processador e da multiprogramação em um SO.

Funções

- A política de escalonamento de um SO possui diversas funções básicas, como:
  - manter o processador ocupado a maior parte do tempo;
  - balancear o uso da UCP entre processos;
  - privilegiar a execução de aplicações críticas;
  - maximizar o throughput do sistema;
  - oferecer tempos de resposta razoáveis para usuários interativos.

Funções

- Cada SO possui sua política de escalonamento adequada ao seu propósito e às suas características.
- A rotina do SO que tem como principal função implementar os critérios da política de escalonamento é denomidado “escalonador (schedule)”.

Funções

- Em um sistema multiprogramável, o escalonador é fundamental, pois todo o compartilhamento do processador é dependente dessa rotina.
- Outra rotina importante na gerência do processador é conhecida como “dispatcher”, responsável pela troca de contexto dos processos após o escalonador determinar qual processo deve fazer uso do processador.

Funções

- O período de tempo gasto na substituição de um processo em execução por outro é denominado “latência do dispatcher”.

Critérios de Escalonamento

- As características de cada SO determinam quais são os principais aspectos para a implementação de uma política de escalonamento adequada.

Critérios de Escalonamento

- Por exemplo, sistemas de tempo compartilhado exigem que o escalonamento trate todos os processos de forma igual, evitando, assim, a ocorrência de “starvation”, ou seja, que um processo fique indefinidamente esperando pela utilização do processador. Já em sistemas de tempo real, o escalonamento deve priorizar a execução de processos críticos em detrimento da execução de outros processos.

Critérios de Escalonamento

- Os principais critérios que devem ser considerados em uma política de escalonamento são:
  - Utilização do Processador
  - Throughput
  - Tempo de Processador / Tempo de UCP
  - Tempo de Espera
  - Tempo de Turnaround
  - Tempo de Resposta

Critérios de Escalonamento: Utilização do Processador

- Na maioria dos sistemas, é desejável que o processador permaneça ocupado a maior parte do tempo.

Critérios de Escalonamento: Troughput

- Representa o número de processos executados em um determinado intervalo de tempo. Quanto maior o throughput, maior o número de tarefas executadas em função do tempo. A maximização é desejada na maioria dos sistemas.

Critérios de Escalonamento: Tempo de Processador (UCP)

- É o tempo que um processo leva no estado de execução durante seu processamento.

Critérios de Escalonamento: Tempo de Espera

- É o tempo total que um processo permanece na fila de pronto durante seu processamento, aguardando ser executado.

Critérios de Escalonamento: Tempo de Turnaround

- É o tempo que um processo leva desde a sua criação até ao seu término, levando em consideração todo o tempo gasto na espera para alocação de memória, espera na fila de pronto, processamento na UCP e na fila de espera.

Critérios de Escalonamento: Tempo de Resposta

- É o tempo decorrido entre uma requisição ao sistema ou à aplicação e o instante em que a resposta é exibida.

Escalonamentos Não-Preemptivos e Preemptivos

- As políticas de escalonamento podem ser classificadas segundo a possibilidade de o SO interromper um processo em execução e substituí-lo por um outro, atividade esta conhecida como “preempção”. SO que implementam escalonamento com preempção são mais complexos, porém possibilitam políticas de escalonamento mais flexíveis.

Escalonamentos Não-Preemptivos e Preemptivos

- O escalonamento não-preemptivo foi o primeiro tipo de escalonamento implementado nos sistemas multiprogramáveis, onde predominava tipicamente o processamento batch. Nesse tipo de escalonamento, quando um processo está em execução nenhum evento externo pode ocasionar a perda do uso do processador.

Escalonamentos Não-Preemptivos e Preemptivos

- Atualmente, a maioria dos SO possui políticas de escalonamento preemptivas que, apesar de tornarem os sistemas mais complexos, possibilitam a implementação dos diversos critérios de escalonamento apresentados.

Escalonamento First-In-First-Out (FIFO)

- No escalonamento FIFO, também conhecido como first-come-first-served (FCFS), o processo que chegar primeiro ao estado de pronto é o selecionado para execução. Este algoritmo é bastante simples, sendo necessária apenas uma fila.

Escalonamento First-In-First-Out (FIFO)

- Apesar de simples, o escalonamento FIFO apresenta algumas deficiências. O principal problema é a impossibilidade de se prever quando um processo terá sua execução iniciada, já que isso varia em função do tempo de execução dos demais processos posicionados à sua frente na fila de pronto.

Escalonamento First-In-First-Out (FIFO)

- Outro problema nesse tipo de escalonamento é que processos CPU-bound levam vantagem no uso do processador sobre processos I/O-bound.
- O escalonamento FIFO é do tipo não-preemptivo e foi inicialmente implementado em sistemas monoprogramáveis com processamento batch, sendo ineficiente se aplicado na forma original em sistemas interativos de tempo compartilhado.

Escalonamento First-In-First-Out (FIFO)

- Atualmente sistemas de tempo compartilhado utilizam o escalonamento FIFO com variações, permitindo, assim, parcialmente sua implementação.

Escalonamento Shortest-Job-First (SJF)

- No escalonamento SJF, também conhecido como shortest-process-next (SPN), o algoritmo de escalonamento seleciona o processo que tiver o menor tempo de processador ainda por executar.

Escalonamento Shortest-Job-First (SJF)

- Na sua concepção inicial, o escalonamento SJF é não-preemptivo. Sua vantagem sobre o escalonamento FIFO está na redução do tempo médio de turnaround dos processos, porém no SJF é possível haver starvation (um processo que fique indefinidamente esperando pela utilização do processador). para processos com tempo de processador muito longo ou do tipo CPU-bound

Escalonamento Cooperativo

- É uma implementação que busca aumentar o grau de multiprogramação em políticas de escalonamento que não possuam mecanismos de preempção, como o FIFO e o SJF não-preemptivo.
- Neste caso, um processo em execução pode voluntariamente liberar o processador retornando à fila de pronto, possibilitando que um novo processo seja escalonado e, assim, permitir melhor distribuição no uso do processador.

Escalonamento Cooperativo

- A principal característica do escalonamento cooperativo está no fato de a liberação do processador ser uma tarefa realizada exclusivamente pelo processo em execução, que de uma maneira cooperativa libera a UCP para um outro processo. Neste mecanismo, o processo em execução verifica periodicamente uma fila de mensagens para determinar se existem outros processos na fila de pronto.

Escalonamento Circular

- È um escalonamento do tipo preemptivo, projetado especialmente para sistemas de tempo compartilhado. Esse algoritmo é bastante semelhante ao FIFO; porém, quando um processo passa para o estado de execução, existe um tempo limite para o uso contínuo do processador denominado “fatia de tempo (time-slice)” ou “quantum”.

Escalonamento Circular

- Toda vez que um processo é escalonado para execução, uma nova fatia de tempo é concedida. Caso a fatia de tempo expire, o SO interrompe o processo em execução, salva seu contexto e direciona-o para o final da fila de pronto. Esse mecanismo é conhecido como “preempção por tempo”.

Escalonamento Circular

- A principal vantagem é não permitir que um processo monopolize a UCP, sendo o tempo máximo alocado continuamente igual à fatia de tempo definido no sistema.

Escalonamento Circular

- Um problema presente nessa política é que o processo e CPU-bound são beneficiados no uso do processador em relação ao processos I/O-bound.

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.

Escalonamento por Prioridades

- Neste escalonamento, o conceito de fatia de tempo não existe; consequentemente, um processo em execução não pode sofrer preempção por tempo.
- A perda do uso do processador só ocorrerá no caso de uma mudança voluntária para o estado de espera ou quando um processo de prioridade maior passa para o estado de pronto.

Escalonamento por Prioridades

- É implementada através de uma interrupção de clock, gerada em determinados intervalos de tempo, para que a rotina de escalonamento reavalie as prioridades dos processos no estado de pronto. Caso haja processos na fila de pronto com maior prioridade do que o processo em execução, o SO realiza a preempção.

Escalonamento por Prioridades

- A prioridade de execução é uma característica do contexto de software de um processo e pode ser classificada como estática ou dinâmica.
- A prioridade estática não tem seu valor alterado durante a existência do processo.
- A prioridade dinâmica pode ser ajustada de acordo com critérios definidos pelo SO.

Escalonamento por Prioridades

- Um dos principais problemas é o “starvation (um processo que fique indefinidamente esperando pela utilização do processador)”.
- Uma solução para esse problema, possível em sistemas que implementam prioridade dinâmica, é a técnica de “aging”. Este mecanismo incrementa gradualmente a prioridade de processos que permanecem por muito tempo na fila de pronto.

Escalonamento Circular com Prioridades

- Implementa o conceito de fatia de tempo e de prioridade de execução associada a cada processo. Neste tipo de escalonamento, um processo permanece no estado de execução até que termine seu processamento, voluntariamente passe para o estado de espera ou sofra uma preempção por tempo ou prioridade.

Escalonamento Circular com Prioridades

- A principal vantagem desse escalonamento é permitir o melhor balanceamento no uso da UCP, com a possibilidade de diferenciar o grau de importância dos processos.
- Esse tipo de escalonamento é amplamente utilizado em sistemas de tempo compartilhado.

Escalonamento por Múltiplas Filas

- Existem diversas filas de processos no estado de pronto, cada qual com uma prioridade específica.
- Os processos são associados às filas em função de características próprias, como importância para a aplicação, tipo de processamento ou área de memória utilizada.

Escalonamento por Múltiplas Filas

- A principal vantagem de múltiplas filas é a possibilidade da convivência de mecanismos de escalonamento distintos em um mesmo SO.
- Cada fila possui um mecanismo próprio, permitindo que alguns processos sejam escalonados pelo mecanismo FIFO, enquanto outros pelo circular.

Escalonamento por Múltiplas Filas

- Uma desvantagem deste escalonamento é que, no caso de um processo alterar seu comportamento no decorrer do tempo, o processo não poderá ser redirecionado para uma outra fila mais adequada. A associação de um processo à fila é determinada na criação do processo, permanecendo até o término do seu processamento.

Escalonamento por Múltiplas Filas com Realimentação

- É semelhante ao escalonamento por múltiplas filas, porém os processos podem trocar de fila durante seu processamento.
- Sua grande vantagem é permitir ao SO identificar dinamicamente o comportamento de cada processo, direcionando-o para fila com prioridade de execução e mecanismo de escalonamento mais adequados ao longo do processamento.

Escalonamento por Múltiplas Filas com Realimentação

- Esse esquema permite que os processos sejam redirecionados entre as diversas filas, fazendo com que o SO implemente um mecanismo de ajuste dinâmico denominado “mecanismo adaptativo”. Os processos não são previamente associados às filas de pronto, e, sim, direcionados pelo sistema para as filas existentes com base no seu comportamento.

Escalonamento por Múltiplas Filas com Realimentação

- Um mecanismo FIFO adaptado com fatia de tempo é implementado para escalonamento em todas as filas, com exceção da fila de menor prioridade que utiliza o escalonamento circular.

Escalonamento por Múltiplas Filas com Realimentação

- É um algoritmo de escalonamento generalista, podendo ser implementado em qualquer tipo de SO. Um dos problemas encontrados nessa política é que a mudança de comportamento de um processo CPU-bound para um I/O-bound pode comprometer seu tempo de resposta. Outro aspecto a ser considerado é sua complexidade de implementação, ocasionando um grande overhead (sobrecarga de processamento) no sistema.

Política de Escalonamento em Sistemas de Tempo Compartilhado

- Caracterizam-se pelo processamento interativo, onde usuários interagem com as aplicações exigindo tempos de respostas baixos.

Política de Escalonamento em Sistemas de Tempo Real

- Algumas aplicações específicas exigem respostas imediatas para a execução de determinadas tarefas. Nesse caso a aplicação deve ser executada em SO de tempo real, onde é garantida a execução de processos dentro de limites rígidos de tempo, sem o risco de a aplicação ficar comprometida.

Política de Escalonamento em Sistemas de Tempo Real

- Aplicações de controle de processos, como sistemas de controle de produção de bens industriais e controle de tráfego aéreo, são exemplos de aplicação de tempo real.
- O escalonamento em sistemas de tempo real deve levar em consideração a importância relativa de cada tarefa na aplicação.

Política de Escalonamento em Sistemas de Tempo Real

- Não deve existir o conceito de fatia de tempo e a prioridade de cada processo deve ser estática.