O que são #Processos?

Processos são softwares ***(programas)*** que executam alguma ação e que podem ser controlados de alguma maneira seja pelo: usuário (processo em primeiro plano) bem como por algum aplicativo / sistema operacional (processo em segundo plano).

Esses programas são um conjunto organizado de instruções em linguagem natural ou codificada ***(algoritmos)*** que descrevem uma tarefa que será realizada pelo computador.

Grande parte dos processos são executados em ***background*** o que permite que o sistema continue trabalhando desta forma executando as demais aplicações que estamos utilizando, ou seja as interfaces gráficas que estão sendo executadas.

De forma bem simplificada um processo é um programa em execução.

O que contêm nos #Processos?

Um processo é composto por uma série de características próprias, no qual sua estrutura básica é formada por uma imagem do código executável associado a um programa. Cada processo possui pelo menos uma linha de execução, ou seja, uma ***Thread*** sendo executada.

Cada processo possui:

- ***Um conjunto de instruções (instruction set)*** – Fornece comandos ao processador para dizer o que ele precisa fazer.
- ***Espaço de endereçamento*** - Espaço reservado para que o processo possa ler e escrever.
- ***Contexto de Hardware –*** Mantém informações nos ***Registradores*** (posições de memória dentro do processador), enquanto um processo está em execução, podendo salvá-las caso o processo seja interrompido***.***
- ***Contexto de Software –*** É onde são especificadas as características e limites dos recursos que podem ser alocadas pelo processo, ele é composto por três grupos de informações sobre o processo:
    - ***Identificação*** - Cada processo possui um identificador ***PID*** *(process identification)* que é representado por um número. Ou um ***UID*** *(user identification)* número de identificação do usuário que o criou.
    - ***Quotas*** - Limites de cada recurso do sistema que um processo pode alocar.
    - **P*rivilégios*** - O que o processo pode ou não fazer em relação a ele mesmo, ao sistema e aos outros processos.

Características de um #Processo

Os processos podem ser **classificados de acordo com o uso do processador ou dos dispositivos de E/S.**

- **Processos *CPU-bound*:** Processos que utilizam muito o processador;
- **Processos I/O-bound**: Processos que realizam muito E/S;

Os processos também podem ser **classificados pela forma de comunicação com o usuário ou com outros processos**.

- ***Processo Foreground*:** Permite a comunicação direta do usuário com o processo durante o seu processamento.
- ***Processo Background:*** Não permite a comunicação direta do usuário com o processo durante a sua execução.

Existem dois **tipos de processos**, sendo eles os:

- ***Processos Independentes:*** Os quais não são afetados pela execução de outros processos.

- ***Processos Cooperantes:*** Que podem afetar a execução de outros processos cooperantes e podem ter sua execução afetadas por eles.

As principais **vantagens da cooperação entre processos** são:
- Compartilhamento de informações
- Aumento da velocidade de computação ***(speed−up)***

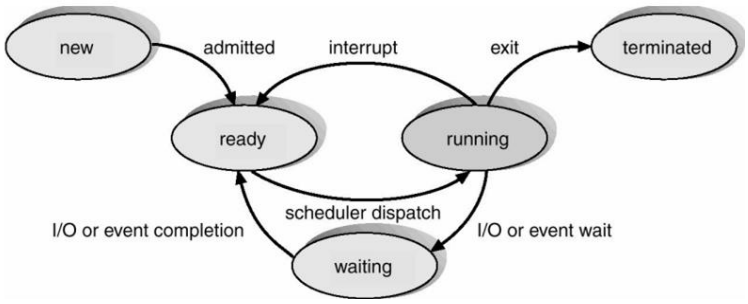
Os processos passam por vários estados durante a sua execução, em função de eventos gerados pelo SO ou pelo próprio processo, um **processo ativo pode encontrar-se em um de três diferentes estados:**

- ***Execução (Running) -*** Quando o processo está em execução pela CPU.
- ***Pronto (Ready) -*** Quando o processo aguarda para ser executado.
- ***Espera (wait) -*** Quando o processo aguarda um evento externo ou por algum recurso para prosseguir seu processamento.

Um **processo muda de estado** durante seu processamento em virtude de eventos gerados pelo sistema operacional ou pelo próprio processo. Existem cinco estados que podem ocorrer a um processo:

- ***New (Novo):*** Processo é criado.
- ***Ready (Pronto):*** É quando o processo fica pronto para execução.
- ***Running (Executando):*** Processo executa suas instruções.
- ***Waiting (Suspenso):*** Esta etapa pode ou não ocorrer, o processo é interrompido e aguarda a ocorrência de algum evento para determinar se será cancelado ou se volta para o estado ***Running*** a fim de tentar uma nova execução.

- ***Terminated (Finalizado):*** O processo termina sua execução e passa para o estado de finalizado.



O que são Threads e Escalonador de #Processos

Um processo pode ter múltiplas ***threads*** (um pequeno programa que trabalha como um subsistema, sendo uma forma de ***um processo se autodividir em duas ou mais tarefas***).

As diversas ***threads*** que existem em um programa podem trocar dados e informações entre si e compartilhar os mesmos recursos do sistema, incluindo o mesmo espaço de memória ela é controlada pelo escalonador.

O ***escalonador (Scheduler)*** é um modulo do SO que escolhe qual será o próximo processo a ser executado, ele pode ser:

- ***Preemptivo:*** É quando um processo pode ser forçado a ser para que outro processo possa usar a CPU.

- ***Não preemptivo:*** Permite que o processo que está sendo executado continue executando até a sua finalização.

O **E*scalonador*** representa cada processo através de um ***PCB (Bloco De Controle De Processos)*** que contém em si:

- ***O estado do processo -*** Estado da execução do processo.
- ***Número do processo -*** PID de identificação do processo
- ***Contador de programa*** - Indica qual é a posição atual na sequência de execução de um processo.
- ***Registradores -*** Posições de memória dentro do processador.
- ***Localização da pilha de execução*** - Registrar o ponto em que cada sub-rotina ativa deve retornar quando termina de executar.
- ***Prioridade de execução –*** Qual a prioridade do processo.

Algoritmos de Escalonamento de Processos

Diversos mecanismos **(algoritmos)** foram sendo desenvolvidos ao longo dos anos; Cada um possui vantagens ou desvantagens.

Categorias de Escalonamento

- **Sistema Batch (Lote):** Processa uma grande quantidade de dados (lote de dados) durante a sua execução, minimiza o tempo entre submissão e o término do processamento, mantem a CPU ocupada o tempo todo.

  - ***First In, First out ou FIFO***: processos são executados na CPU seguindo a ordem de requisição.
    Desvantagem: Ineficiente quando há processos que demoram na sua execução.

  - ***Shortest Job First / SJF***: O processo com o menor tempo de execução ***(turnaround)*** é executado primeiro.
    Desvantagem: Se muitos processos curtos começarem a chegar, os longos podem demorar a serem executados

  - ***Shortest remaining Time Next / SRTF:*** O processo com a menor quantidade de tempo restante até a conclusão é selecionado para execução.
    Desvantagem: Processos que consomem mais tempo podem demorar muito para serem finalizados.

- ***Sistemas Interativos -*** Responde rapidamente as requisições, satisfaz as expectativas dos usuários.

  - ***Round-Robin:*** Os processos recebem um ***Quantum*** (tempo para a execução), quando esse tempo termina o processo é interrompido voltando para o final da fila, dando início ao próximo processo na ordem da fila.
    Desvantagem: O tempo de resposta em processos muito longo é comprometido.

  - ***Processo por Prioridade:*** Um processo é interrompido e substituído se um de mais alta prioridade surge na fila, nesse sistema os processos além da prioridade funcionam em conjunto com o algoritmo **Round-Robin.**
    Desvantagem: processos com menos prioridade podem nunca rodar ***(starvation).***

  - ***Múltiplas Filas (Multiple queues):*** Existe mais de uma fila de processos, em que a prioridade está associada à fila, não ao processo, pode ser aplicado em cada fila, diferentes algoritmos de escalonamento **(Round-robin, FIFO** etc.**)**

- ***Sistemas Tempo real -*** Evita a perda de dados, são algoritmos voltados para aplicações em sistemas críticos. **Hard Real Time** (atrasos não são tolerados ex - Aviões, usinas nucleares, hospitais). **Soft Real Time** (atrasos ocasionais são tolerados ex - Bancos, falha em multimidias).

  - ***Algoritmos Estáticos -*** Decisões de escalonamento são tomadas antes do sistema começar a rodar.
  - ***Algoritmos Dinâmicos -*** Decisões de escalonamento em tempo de execução.