

Processos e Threads

Rafaela Pavoski

Fatec São José do Rio Preto

Assembly Tecnologia

Processo é um nome abstrato dado para designar como os programas são executados no Sistema Operacional. E para gerar um processo, é preciso um código executável para o tipo de arquitetura (processador) da máquina utilizada. Em alguns casos é possível criar programas que são traduzidos para serem executados em diferentes sistemas computacionais e usa-se uma máquina virtual para isso.

Quando iniciado, o Sistema Operacional lê do disco seu código executável e o carrega na memória principal. Ele aloca um espaço na memória para fazer a carga de código e também para as áreas de dados. Além disso, muitas informações são mantidas pelo SO. Cria-se um PCB (process control block), que possui informações como PID (identificador do processo), o estado do processo, um contador de programa, valores de prioridade para escalonamento, credenciais do usuário, processo pai, ponteiros para processo filho, recursos alocados, tais como buffers, arquivos abertos, mecanismos de comunicação entre processos, e também o contexto. O processo pode assumir diversos estados após criado, inicialmente, fica em estado “pronto”, está pronto para ser executado. Como há outros processos para serem executados, forma-se uma fila de processos prontos, que são definidos pela sua prioridade. Se encontra em execução até que um evento faça essa mudança de estado, normalmente para bloqueado ou interrompido, onde finaliza a execução desse programa.

Pensando em resolver problemas com lentidão, facilitar o compartilhamento de dados e deixar a memória mais eficiente, foram desenvolvidos as chamadas *Threads*. Aplicações que executam diversas atividades simultâneas tendem a executar de forma mais rápida se no lugar de processos, utilizarem *threads*. Visando o desenvolvimento de paralelismo de execução, processos e *threads* são gerenciados pelo SO, contudo, ambos precisam ter informações de contexto que são mantidas no PCBs. O modelo *threads* é usado quando, contudo, criam-se *threads* associadas a um processo. Dessa forma, compartilham sua área de memória onde está o código executado, e também a área de dados, mesmo que o SO precise manter um PCB para cada *threads*, várias informações que estão no PCB do processo quando associadas são compartilhadas entre elas. Atualmente, é comum um processo possuir pelo menos uma *threads*.

Um bom exemplo de *Threads*, para melhor compreensão, é um simples editor de texto que normalmente é um programa multithreads. Enquanto uma *threads* coleta o que digitamos, outra *threads* faz a repaginação do texto e uma terceira faz ainda a gravação de segurança a cada intervalo de tempo. A maioria e mais importantes aplicações que utilizamos são multithreads, jogos interativos, servidores em geral e muitos outros. Portanto, as *threads* são ótimos amigos na execução de um programa com múltipla funcionalidade e auxiliam na agilidade e eficiência do mesmo.