

Capítulo 03: Processos

3.1 Conceito de Processo

3.1.1 O Processo

- **Processo:** programa em execução.
- **Seção de texto:** código de um programa.
- Um processo inclui, além da seção de texto, o **contador do programa** e o conteúdo dos registradores do processador.
- Geralmente, um processo também inclui a **pilha** de processo, além de uma **seção de dados** (que contém os dados globais) e um **heap**.



- Um programa por si só não é um processo:
 - Trata-se de uma entidade *passiva* (um **executável**)
- Um processo é uma entidade *ativa*, com um *program counter* e um conjunto de recursos associados.
- **Um programa se torna um processo quando um arquivo executável é carregado na memória.**
- Um programa pode estar associado a vários processos.

3.1.2 Estado do Processo

- Possíveis estados de um processo (os nomes são arbitrários):
 - **Novo:** o processo está sendo criado.
 - **Em execução:** instruções estão sendo executadas.
 - **Em espera:** o processo está esperando que algum evento ocorra (operação e I/O, por exemplo)
 - **Pronto:** o processo está esperando ser atribuído a um processador.
 - **Concluído:** o processo terminou sua execução.
- Quando um processo está sendo executado, ele muda de **estado**.
- **Somente um processo pode estar sendo executado em um processador a cada instante; porém, muitos podem estar prontos e em espera.**
- Possível caminho de estados:
 - novo -> pronto -> em execução -> em espera -> pronto -> em execução -> concluído
 - as mudanças de estados acima poderia ter sido, por exemplo:
 - * Criou-se um novo processo;
 - * Assume estado pronto, esperando para ser executado;
 - * Após ser executado, entra em estado de espera de I/O;
 - * Após a operação de I/O, assume estado pronto;
 - * Entra em execução e é encerrado.

3.1.3 Bloco de Controle de Processo

- Cada processo é representado no SO por um **bloco de controle de processo (PCB)**.
- Um PCB possui muitos trechos de informação associados a um processo específico, incluindo:
 - **Estado do processo**
 - **Contador do programa**: indica o endereço da próxima instrução
 - **Registradores da CPU**: junto com o contador do programa, as informações do estado devem ser salvas quando ocorrer uma interrupção.
 - **Informações da scheduling da CPU**: essas informações incluem a prioridade de um processo, ponteiros de filhas de scheduling, etc. [capítulo 5]
 - **Informações de gerenciamento da memória** [capítulo 8]
 - **Informações de contabilização**: incluem o período de tempo real e de CPU usados, os limites de tempo, números de jobs ou processos, etc.
 - **Informações de I/O**: incluem a lista de dispositivos de I/O alocados para o processo, uma lista de arquivos abertos, etc.
- Resumindo: **O PCB serve como repositório de qualquer informação que possa variar de um processo para outro.**

3.1.4 Threads

- Até o momento, consideramos apenas processos com um thread.
- Atualmente, muitos sistemas operacionais permitem que um processo tenha vários threads em execução.
- Neste caso, o PCB é estendido e possui informações sobre cada thread.
- Outras alterações no sistema como um todo também são necessárias, discutidas no *Capítulo 4*.

3.2 Scheduling de Processos

- O objetivo da multiprogramação é alternar processos com frequência tão alta que dá a impressão ao usuário que diversos processos ocorrem simultaneamente.
- Em um computador com uma única CPU, nunca haverá mais de um processo sendo executado ao mesmo tempo.
- O número de processos na memória é chamado de **grau de multiprogramação** (*degree of multiprogramming*).
- Um programa normalmente é caracterizado como **CPU-bound** ou **I/O-bound**:
 - **CPU-bound**: gera processos de I/O com baixa frequência
 - **I/O-bound**: gera processos que passam maior parte do tempo com operações I/O.

3.2.1 Filas de Scheduling

- Quando um processo entra em um sistema, são inseridos em uma **fila de jobs** (*jobs queue*), que contém todos os jobs do sistema.
- Se o processo está pronto, esperando para entrar em execução, é inserido na **fila de prontos** (*ready queue*), normalmente armazenada como uma fila encadeada.
- O cabeçalho de uma ready queue contém ponteiros para o primeiro e o último da PCBs da lista.
- **Fila de dispositivo**:
 - Trata-se de uma **fila de espera** (*wait queue*) de I/O de um dispositivo.
 - Cada dispositivo possui sua própria fila de dispositivo.
- **Diagrama de enfileiramento** (*queueing diagram*): Figura 01

3.2.2 Schedulers

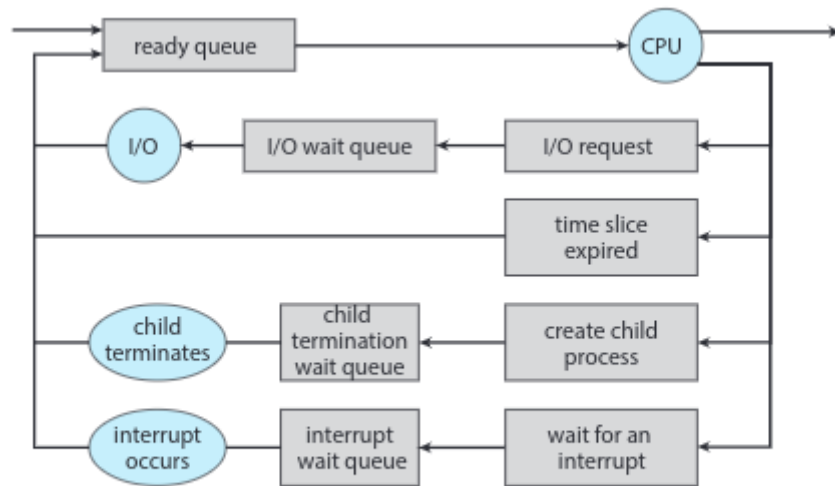


Figure 3.5 Queueing-diagram representation of process scheduling.

Figure 1: queueing diagram