Processos

Introdução

- O conceito de Processo é a base para a implantação de um sistema multiprogramável.
- O processador é projetado apenas para executar instruções, não sendo capaz de distinguir qual programa se encontra em execução.

Introdução

- A gerência de um ambiente multiprogramável é uma função exclusiva do SO, que deve controlar a execução dos diversos programas e o uso corrente do processador.
- Para isso, para ser executado, um programa deve estar sempre associado a um processo.

Introdução

- A gerência de processos é uma das principais funções de um SO. Através de processos, um programa pode alocar recursos, compartilhar dados, trocar informações e sincronizar sua execução.

Introdução

- Nos sistemas multiprogramáveis, os processos são executados concorrentemente, compartilhando, dentre outros recursos, o uso do processador, da memória principal e dos dispositivos de E/S.

Estrutura do Processo

- Um processo pode ser entendido inicialmente como um programa em execução, só que seu conceito é mais abrangente.
- Em um sistema multiusuário, cada usuário é associado a um processo.

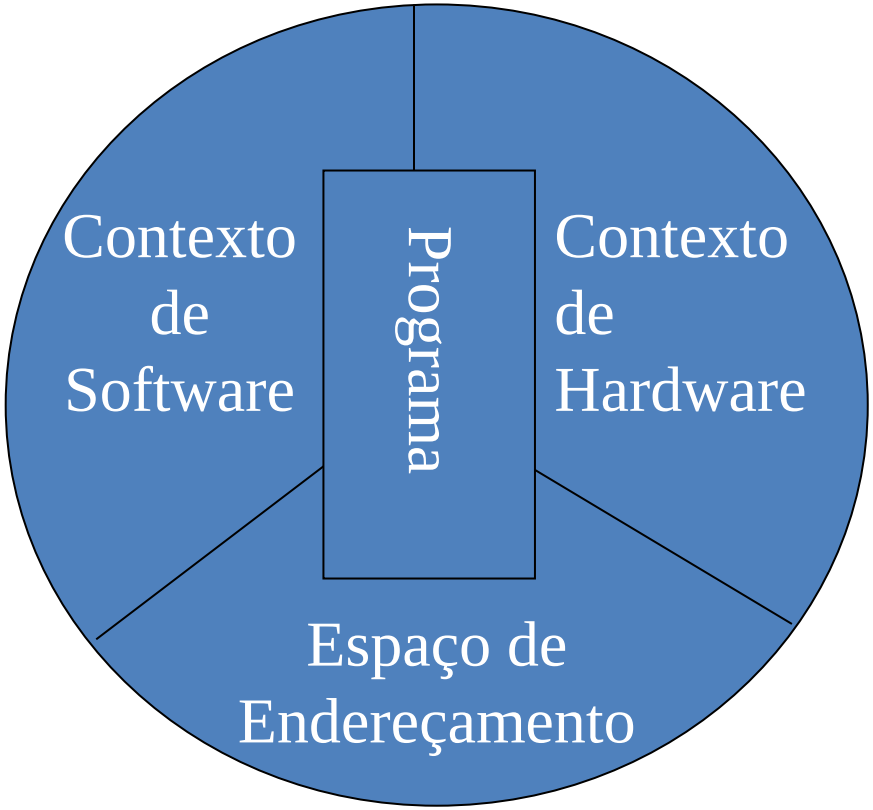
Estrutura do Processo

- Um processo também pode ser definido como o ambiente onde um programa é executado. Este ambiente, além das informações sobre a execução, possui também o quanto de recursos do sistema cada programa pode utilizar, como o espaço de endereçamento, tempo de processador e área em disco.

Estrutura do Processo

- Um processo é formado por três partes, conhecidas como contexto de hardware, contexto de software e espaço de endereçamento, que juntas mantêm todas as informações necessárias à execução de um programa.

Estrutura do Processo



Contexto de Hardware

- Armazena o conteúdo dos registradores gerais da UCP, além dos registradores de uso específico. Quando um processo está em execução, o seu contexto de hardware está armazenado nos registradores do processador. No momento em que o processo perde a utilização da UCP, o sistema salva as informações no contexto de hardware do processo.

Contexto de Hardware

- O contexto de hardware é fundamental para a implementação dos sistemas multiprogramáveis, onde os processos se revezam na utilização da UCP, podendo ser interrompidos e, posteriormente, restaurados.

Contexto de Hardware

- A troca de um processo por outro no processador, comandada pelo SO, é denominada “mudança de contexto”.
- A mudança de contexto consiste em salvar o conteúdo dos registradores do processo que está deixando a UCP e carregá-los com os valores referentes ao do novo processo que será executado.

Contexto de Software

- São especificadas características e limites dos recursos que podem ser alocados pelo processo, como o número máximo de arquivos abertos simultaneamente, prioridade de execução e tamanho do buffer para operações de E/S.

Contexto de Software

- O contexto de software é composto por três grupos de informações sobre o processo:
  - Identificação
  - Quotas
  - Privilégios

Contexto de Software: Identificação

- Cada processo criado pelo sistema recebe uma identificação única (PID – process identification) representada por um número.
- Através do PID, o SO e outros processos podem fazer referência a qualquer processo existente.

Contexto de Software: Identificação

- O processo também possui a identificação do usuário ou processo que o criou (owner). Cada usuário possui uma identificação única no sistema (UIS – user identification), atribuída ao processo no momento de sua criação.

Contexto de Software: Quotas

- As quotas são limites de cada recurso do sistema que um processo pode alocar.
- Caso uma quota seja insuficiente, o processo poderá ser executado lentamente, interrompido durante seu processamento ou mesmo não ser executado.

Contexto de Software: Quotas

- Alguns exemplos de quotas presentes na maioria dos SO são:
  - Número máximo de arquivos abertos simultaneamente.
  - Tamanho máximo de memória principal e secundária que o processo pode alocar.
  - Número máximo de operações de E/S pendentes.
  - Tamanho máximo do buffer para operações de e/s.
  - Número máximo de processos, subprocessos e threads que podem ser criados

Contexto de Software: Privilégios

- Os privilégios e direitos definem as ações que um processo pode fazer em relação a ele mesmo, aos demais processos e ao SO.

Espaço de Endereçamento

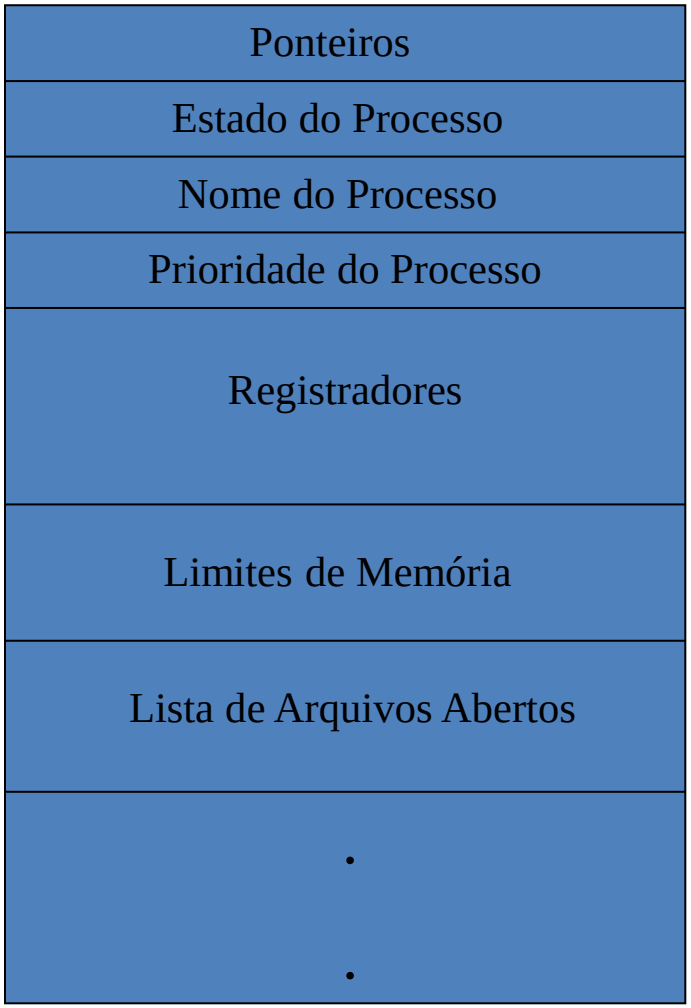
- É a área de memória pertencente ao processo onde as instruções e os dados do programa são armazenados para execução.
- Cada processo possui seu próprio espaço de endereçamento, que deve ser devidamente protegido do acesso dos demais processos.

Bloco de Controle de Processo

- O processo é implementado pelo SO através de uma estrutura de dados chamada “bloco de controle do processo” (Process Control Block – PCB).
- A partir do PCB, o SO mantém todas as informações sobre o contexto de hardware, contexto de software e espaço de endereçamento de cada processo.

Bloco de Controle de Processo

PCB



Bloco de Controle de Processo

- Os PCBs de todos os processos residem na memória principal em uma área exclusiva do SO. O tamanho desta área geralmente é limitado por um parâmetro do SO que permite especificar o número máximo de processos que podem ser suportados simultaneamente pelo sistema.

Bloco de Controle de Processo

- Toda a gerência dos processos é realizadas através de system calls, que realizam operações como criação, alteração de características, visualização, eliminação, sincronização, suspensão de processos, dentre outras.

Estados do Processo

- Os processos passam por diferentes estados ao longo do seu processamento, em função de eventos gerados pelo SO ou pelo próprio processo. Um processo ativo pode encontrar-se em três diferentes estados:
  - Execução (running)
  - Pronto (ready)
  - Espera (wait)

Estados do Processo

- Execução (running): um processo é dito no “estado de execução” quando está sendo processado pela UCP. Em sistemas com apenas uma UCP, somente um processo pode estar sendo executado em um dado instante.

Estados do Processo

- Pronto (ready): um processo está no “estado de pronto” quando aguarda apenas para ser executado. O SO é responsável por determinar a ordem e os critérios pelos quais os processos em estado de pronto fazem uso do processador. Este mecanismo é conhecido como “escalonamento”

Estados do Processo

- Espera (wait): um processo no “estado de espera” aguarda por algum evento externo ou por algum recurso para prosseguir seu processamento. Em alguns SO, o estado de espera pode ser chamado de bloqueado (blocked).

Mudanças de Estado de Processo

- Um processo muda de estado durante seu processamento em função de eventos originados por ele próprio (eventos voluntários) ou pelo SO (eventos involuntários). Basicamente, existem quatro mudanças de estado que podem ocorrer a um processo:

Mudanças de Estado de Processo

- Pronto → Execução
- Execução → Espera
- Espera → Pronto
- Execução → Pronto

Mudanças de Estado de Processo

- Um processo em estado de pronto ou de espera pode não se encontrar na memória principal. Esta condição ocorre quando não existe espaço suficiente para todos os processos na memória principal e parte do contexto do processo é levada para memória secundária. Uma técnica conhecida como “swapping” retira processos da memória principal e os traz de volta seguindo critérios de cada SO.

Criação e Eliminação de Processos

- A criação de um processo ocorre a partir do momento em que o SO adiciona um novo PCB à sua estrutura e aloca um espaço de endereçamento na memória para uso.

Criação e Eliminação de Processos

- Criação (new): Um processo é dito no “estado de criação” quando o SO já criou um novo PCB, porém ainda não pode colocá-lo na lista de processos do estado de pronto.
- Terminado (exit): Um processo no “estado de terminado” não poderá ter mais nenhum programa executado no seu contexto, porém o SO ainda mantém suas informações de controle presentes na memória.

Processos Independentes, Subprocessos e Threads

- Processos independentes, subprocessos e threads são maneiras diferentes de implementar a concorrência dentro de uma aplicação. Neste caso, busca-se subdividir o código em partes para trabalharem de forma cooperativa.

Processos Independentes, Subprocessos e Threads

- O uso de “processos independentes” é a maneira mais simples de implementar a concorrência em sistemas multiprogramáveis. Neste caso não exíste vínculo do processo criado com o seu criador.

Processos Independentes, Subprocessos e Threads

- Subprocessos são processos criados dentro de uma estrutura hierárquica. Neste modo, o processo criador é denominado “processo pai” enquanto o novo processo é chamado de “subprocesso” ou “processo filho”.

Processos Independentes, Subprocessos e Threads

- O conceito de “thread” foi introduzido na tentativa de reduzir o tempo gasto na criação, eliminação e troca de contexto de processos nas aplicações concorrentes, bem como economizar recursos do sistema como um todo.

Processos Independentes, Subprocessos e Threads

- Em um ambiente “multithread”, um único processo pode suportar múltiplos threads, cada qual associado a uma parte do código de aplicação. Neste caso não é necessário haver diversos processos para a implementação da concorrência. Threads compartilham o processador da mesma maneira que um processo, ou seja, enquanto um thread espera por uma operação de E/S, outro thread pode ser executado.

Processos Foreground e Background

- Um “processo foreground” é aquele que permite a comunicação direta do usuário com o processo durante o seu processamento. Neste caso, tanto o canal de entrada quanto o de saída estão associados a um terminal com teclado, mouse e monitor, permitindo, assim, a interação com o usuário.

Processos Independentes, Subprocessos e Threads

- Um “processo background” é aquele onde não existe a comunicação com o usuário durante o seu processamento.

Processos do Sistema Operacional

- Quando processos são utilizados para a implementação de serviços do sistema, estamos retirando código do seu núcleo, tornando-o menor e mais estável.

Processos do Sistema Operacional

- Alguns serviços que pode ser implementados por meio de processos:
  - Auditoria e segurança
  - Serviços de rede
  - Contabilização do uso de recursos
  - Contabilização de erros
  - Gerência de impressão
  - Gerência de jobs batch
  - Temporização
  - Comunicação de eventos
  - Interface de comandos (shell)

Processos CPU-Bound e I/O-Bound

- Um processo é definido como CPU-Bound (ligado a CPU) quando passa a maior parte do tempo no estado de execução, ou seja, utilizando o processador.

Processos CPU-Bound e I/O-Bound

- Um processo é classificado como I/O-Bound (ligado à E/S) quando passa a maior parte do tempo no estado de espera, pois realiza um elevado número de operações de E/S.

Sinais

- Sinais são um mecanismo que permite notificar processos de eventos gerados pelo SO ou por outros processos.
- O uso de sinais é fundamental para a gerência de processos, além de possibilitar a comunicação e sincronização entre processos.