

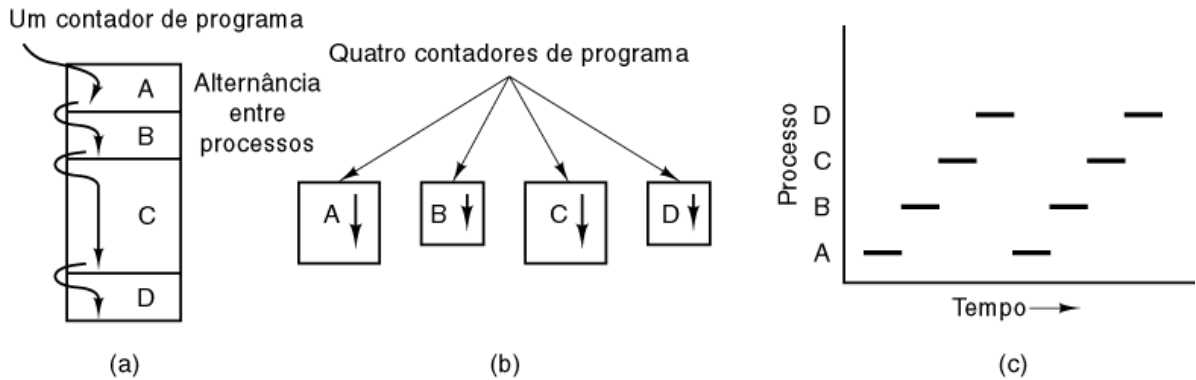
Sistemas Operacionais

Prof. Robson de Souza

Aulas 7 e 8

Conteúdo: Introdução aos processos.

Processo pode ser visto como um programa em execução, mas também pode ser visto como um conjunto necessário de informações para que o sistema operacional implemente a concorrência entre programas.

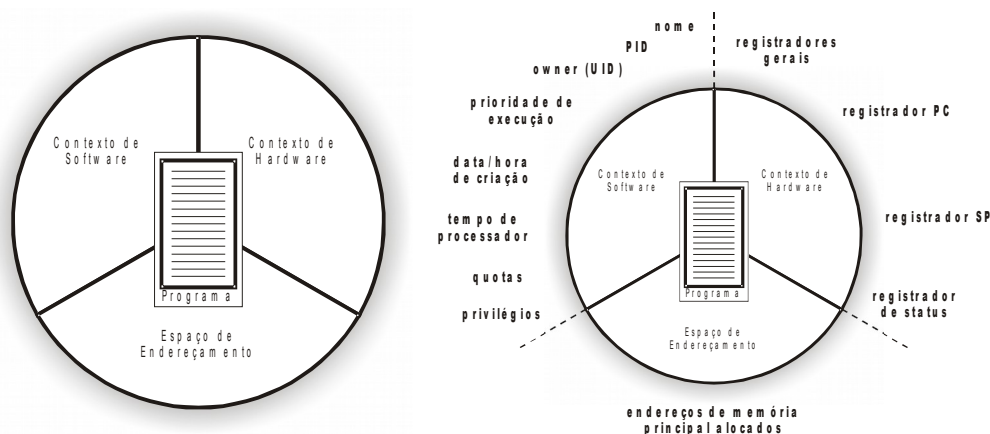


Na figura acima, temos (a) Um computador multiprogramado com quatro programas na memória, (b) quatro processos, cada um com seu próprio fluxo de controle executando de maneira independente um do outro e (c) a partir de um determinado tempo todos os processos fizeram progresso, mas em um dado instante só um processo está realmente executando.

Existe uma diferença entre processo e programa, os processos fazem parte do programa, podemos entender como se eles fossem atividades a serem realizadas.

Estrutura de um processo

Um processo é composto de 3 partes: contexto de hardware, contexto de software e espaço de endereçamento.



O **contexto de hardware** de um processo armazena o conteúdo dos registradores gerais da CPU, além dos registradores de uso específico, como o program counter (PC), stack pointer (SP) e o registrador de status.

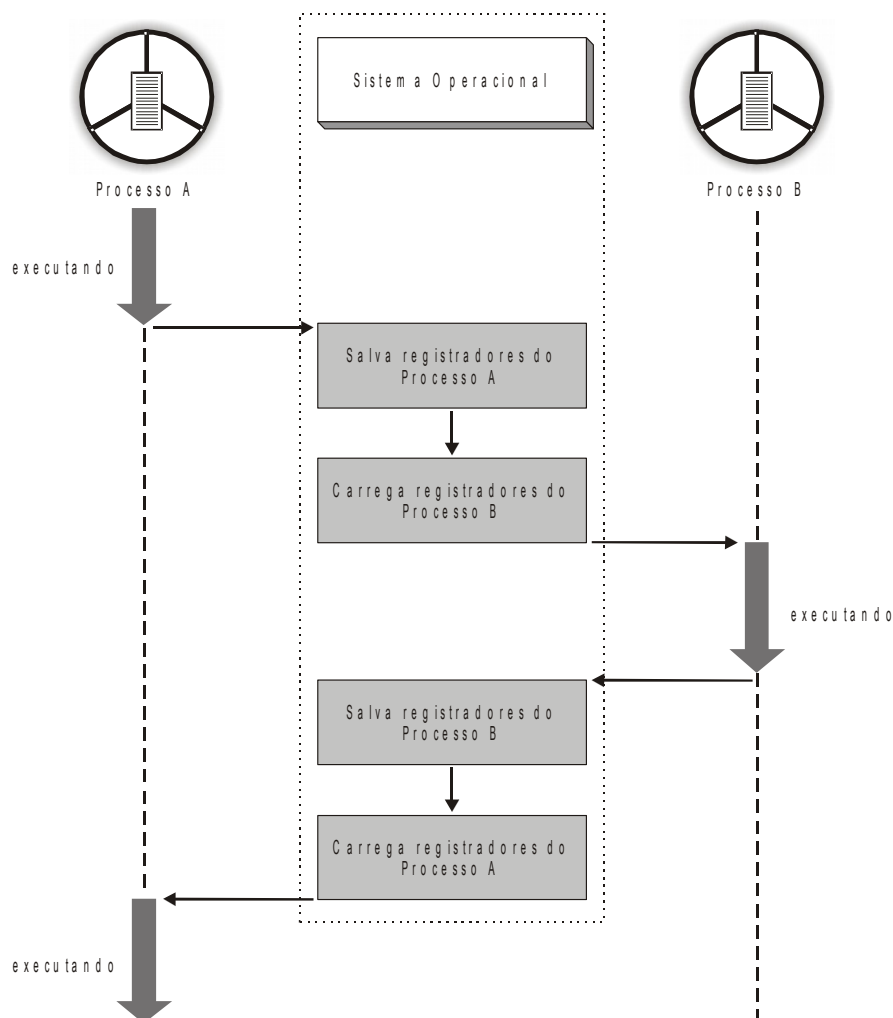
O **espaço de endereçamento** é a área de memória pertencente ao processo onde instruções e dados do programa são armazenados para execução. Cada processo tem o seu próprio espaço de endereçamento

No **contexto de software** de um processo são especificados limites e características de um processo divididos em 3 grupos de informações:

- Identificação → Cada processo recebe uma identificação única, PID e a identificação do usuário ou processo que o criou, UID.
- Quotas:
 - Número máximo de arquivos abertos simultaneamente;
 - Tamanho máximo de memória principal e secundária;
 - Número máximo de operações de E/S.
 - Tamanho máximo dos buffers;
 - Número máximo de processos e threads que podem ser criados.
- Privilégios → Podem ser modificados pelo administrador do sistema:
 - Unix via conta “root”
 - MS Windows via conta “administrator”
 - Open VMS via conta “system”

Mudança de contexto

Para que, em um determinado tempo t ocorra uma troca na execução de um programa, o conjunto de registradores e dados deverá ser salvo para posterior recuperação. A troca de um processo por outro no processador, comandada pelo sistema operacional, é denominada **mudança de contexto**. É desta forma que o sistema operacional implementa a gerência em um ambiente multiprogramável.



Hierarquia de processos

Os processos seguem uma hierarquia, de modo que um processo pode criar outro e assim ir gerando uma árvore de processos, como por exemplo um processo de inicialização a partir do qual todos os outros são gerados. Em alguns sistemas operacionais, quando um processo cria outro processo, é gerada uma cópia idêntica ao processo que o gerou.

De qualquer forma, os processos precisam de alguma maneira para criar novos processos quando necessário.

Estados de um processo

Existem estados nos quais um processo pode estar, isso é importante pois os processos precisam interagir entre si, um processo pode gerar um saída que um outro processo pode precisar utilizar como entrada. Conforme o processo executa, ele vai mudando de estado. Os estados de um processo são os seguintes:

novo: O processo está sendo criado

executando: As instruções estão sendo executadas

esperando: O processo está esperando que ocorra algum evento

pronto: O processo está esperando para ser atribuído a um processador

terminado: O processo terminou sua execução

O diagrama abaixo mostra a transição desses estados em um processo.



Pelo diagrama, percebe-se que um processo no estado “novo” só pode ir para o estado “pronto” e apenas um processo “executando” pode ir para o estado “terminado”, se o processo estiver “esperando”, antes de ir para o estado “executando” ele precisa ir para o estado “pronto”. Um processo no estado “pronto” pode ir para o estado “executando” e vice-versa.

Estado de execução → quando está sendo processado pela CPU.

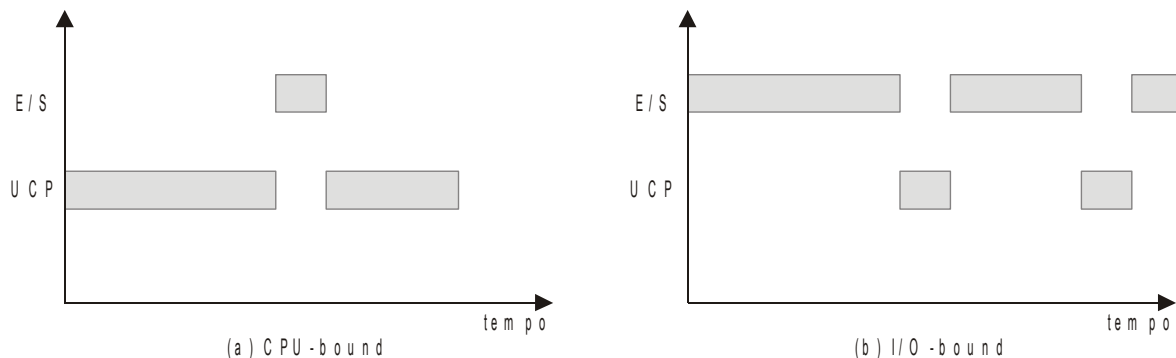
Estado de Pronto → quando aguarda apenas ser executado. O S.O. É responsável pelos critérios e ordem de execução. Esse mecanismo é conhecido por escalonamento.

Estado de espera → quando o processo espera algum evento externo ou espera algum recurso como por exemplo uma operação de E/S.

Os processos podem ser classificados como **processos CPU-bound** ou **processos I/O bound (E/S bound)**.

Um processo é definido como *CPU-bound* (ligado à CPU) quando passa a maior parte do tempo em estado de execução, utilizando o processador ou no estado de pronto, realizando poucas operações de E/S.

Um processo é definido como *E/S-bound* quando passa a maior parte do tempo em estado de espera, pois realiza muitas operações de E/S. Aplicações comerciais e aplicações interativas são exemplos de processos E/S-bound



Implementação de processos

Para implementar o modelo de processos, o S.O mantém uma tabela chamada **tabela de processos**, que é uma matriz de estruturas e contém uma entrada por processo.

A entrada contém as informações sobre o estado do processo, seu contador de programa, o ponteiro da pilha, alocação de memória, status de arquivos abertos, informações de contabilidade e escalonamento e tudo mais de informação sobre o processo que deva ser salvo no caso de uma interrupção para retomar a execução posteriormente.

Cada processo é implementado no S.O através de um **Bloco de Controle de Processo (PCB)**. Cada PCB mantém informações associadas a cada processo, estado do processo, contador de programa, registradores da CPU, informações de escalonamento da CPU, informação de gerencia de memória, entre outros.

Os PCBs de todos os processos ativos residem na memória principal numa área exclusiva do S.O.

Criação e término de processos

Existem alguns eventos que levam à criação de processos, tais como o início do sistema, execução de chamada ao sistema de criação de processos, solicitação do usuário para criar um novo processo, entre outras condições.

As condições que levam ao término de processos são:

- Saída normal (voluntária)
- Saída por erro (voluntária)
- Erro fatal (involuntário)
- Cancelamento por outro processo (involuntário)

Referências bibliográficas:

TANENBAUM, Andrew. 2ª ed. **Sistemas Operacionais Modernos**, Editora Pearson, 2003.

SILBERSCHATZ, Abraham. **Sistemas Operacionais com JAVA**, 6ª ed. Editora Campus

MACHADO, Francis B. **Arquitetura de Sistemas Operacionais**, 4ª ed, LTC, 2007.