




Plano de Ensino



- Introdução: Conceitos e Definições.
- **Processos, Comunicação e Sincronização de Sistemas Distribuídos.**
- Processamento Paralelo e Multiprocessadores.
- Tolerância a falhas.
- Aplicações Distribuídas: Socket.
- Objetos Distribuídos: RMI.



Livro-Texto



- Bibliografia Básica:
 - » TANENBAUM, Andrew S; STEEN, Maarten Van. Sistemas Distribuídos : Princípios e Paradigmas. 2ª Ed. São Paulo: Pearson Prentice Hall, 2007.
- Bibliografia Complementar:
 - » DANTAS, Mario. Computação Distribuída de Alto Desempenho : redes, clusters e grids computacionais. 1ª Ed. Rio de Janeiro: Axcel Books, 2005.
 - » SILBERSCHATZ, Abraham; GALVIN, Peter Baer; GAGNE, Greg. Sistemas Operacionais - Conceitos e Aplicações. 1ª Ed. Rio de Janeiro: Campus, 2001.

2. Processos, Comun. e Sincr. – SO



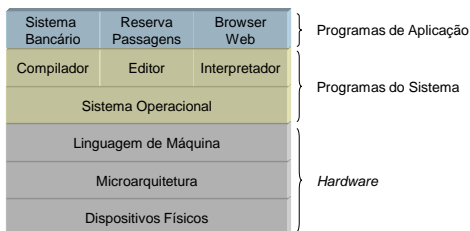
- “Um programa que atua como um intermediário entre o usuário e o hardware de um computador” (Silberschatz).
- “Um conjunto de programas que controla os recursos do computador e provê a base sobre a qual as aplicações são escritas” (Tanenbaum).

2. Processos, Comun. e Sincr. – Análise de SO



- É uma máquina estendida.
 - » Oculta os detalhes complicados que têm que ser executados.
 - » Apresenta ao usuário uma máquina virtual, mais fácil de usar.
- É um gerenciador de recurso.
 - » Cada programa tem um tempo com o recurso.
 - » Cada programa tem um espaço no recurso.
 - » Cada programa tem sua vez de usar o recurso.

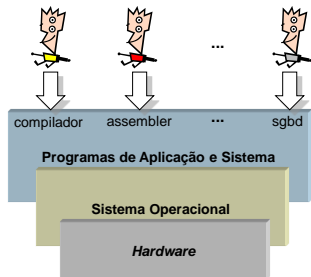
2. Processos, Comun. e Sincr. – Componentes



2. Processos, Comun. e Sincr. – Componentes



Visão Abstrata dos
Componentes de um
Sistema de
Computação



2. Processos, Comun. e Sincr. – Componentes



- **Hardware:** provê recursos básicos (CPU, memória, dispositivos de I/O, dentre outros).
- **Sistema Operacional:** controla a utilização dos recursos entre os usuários.
- **Programas de Aplicação:** definem o modo como os recursos são usados para solucionar problemas dos usuários.
- **Usuários:** pessoas e outros computadores.

2. Processos, Comun. e Sincr. – Objetivos SO



- Executar programas de usuário para solucionar seus problemas mais facilmente.
- Tornar o computador conveniente ao uso.
- Utilizar o *hardware* de maneira eficiente.
- Compartilhar os recursos de um sistema computacional entre vários usuários.

2. Processos, Comun. e Sincr. – Serviços SO



- Facilidade para criação de programas
 - » Editores, compiladores, depuradores e linkeditores
- Execução de programas
 - » Carga e inicialização de arquivos e E/S
- Acesso controlado a arquivos
 - » Formas e formatos específicos de E/S
- Acesso ao sistema
 - » Proteção de acesso aos recursos e dados
 - » Contenção no uso de recursos compartilhados
- Contabilização de uso
 - » Estatísticas de uso e parâmetros de desempenho

2. Processos, Comun. e Sincr. – Serviços SO



- Detecção de erros
 - » Erro de hardware: erro de memória, falha de dispositivo, ...
 - » Erro de software: estouro aritmético, acesso não permitido, ...
 - » Inabilidade do SO em garantir resposta
- Tratamento do erro
 - » Simplesmente reporta o erro ao aplicativo
 - » Repete a operação
 - » Aborta a operação

2. Processos, Comun. e Sincr. – Histórico SO



- 1ª Geração: 1945 – 1955
- 2ª Geração: 1955 – 1965
- 3ª Geração: 1965 – 1980
- 4ª Geração: 1980 – 1990
- 5ª Geração: 1990 – ????

2. Processos, Comun. e Sincr. – Histórico SO



1ª Geração: 1945-1955

- » Válvulas e Painéis
- » Não existia linguagem de programação
- » Único grupo: projeto, construção, operação e manutenção
- » Utilizado para cálculos numéricos repetitivos



UNIVAC (1951) com 5 mil válvulas (1000 instruções/s)

2. Processos, Comun. e Sincr. – Histórico SO



2ª Geração: 1955-1965

- » Transistores: os computadores tornaram-se confiáveis a ponto de serem comercializados (alto custo).
- » Batch (cartões perfurados): redução do tempo desperdiçado.
- » Fortran ou linguagem de montagem (cálculos científicos).



Típico Cartão-Perfurado (80 colunas) em linguagem Cobol

2. Processos, Comun. e Sincr. – Histórico SO



3ª Geração: 1965-1980

- » CIs: diminuição dos custos.
- » Multiprogramação
 - permite que enquanto um programa esperasse por uma operação de leitura/gravação o processador executasse outro programa
- » Spooling (*Simultaneous Peripheral Operation On-Line*).
- » Time-sharing: cada programa utiliza o processador em pequenos intervalos de tempo.



Altair (1975)

2. Processos, Comun. e Sincr. – Histórico SO



- 4ª Geração: 1980-1990
 - » PC, DOS.
 - » Microcomputadores.
 - » Redes de computadores.



Lisa (1983)

2. Processos, Comun. e Sincr. – Histórico SO



- 5ª Geração: 1990-????
 - » Processamento distribuído
 - » Funções espalhadas por vários processadores através de redes de computadores.
 - » Novas interfaces.
 - » Linguagens.
 - » Comunicação em Rede.



Apple TV (Apple), Core i7 (Intel), iPad (iOS) e Samsung S3 (Android)

2. Processos, Comun. e Sincr. – Tipos Sistemas



- Monoprogramação ou Monotarefa
 - » O processador, memória e os periféricos ficam dedicados a um único usuário.
 - » Enquanto o programa aguarda por um evento, como a digitação dos dados, o processador ficará ocioso sem realizar qualquer tarefa.



2. Processos, Comun. e Sincr. – Tipos Sistemas



▪ Monoprogramação ou Monotarefa

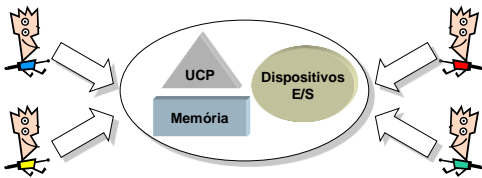
- » A memória é sub-utilizada caso o programa não a preencha totalmente.
- » Os periféricos, como discos e impressoras, estão dedicados a um único processo, nem sempre utilizados de forma integral.
- » Simples de Implementação (se comparados com outros sistemas).
- » Sem problema de proteção (só um processo).
- » *Exemplo: DOS.*

2. Processos, Comun. e Sincr. – Tipos Sistemas



▪ Multiprogramação ou Multitarefa

- » Vários processos dividem os mesmos recursos.
- » Mais complexos que os monoprogramáveis.



2. Processos, Comun. e Sincr. – Tipos Sistemas



▪ Multiprogramação ou Multitarefa

- » SO se preocupa em gerenciar o acesso concorrente: de forma ordenada e protegida.
- » Possibilita o aumento de produtividade dos programas.
- » Reduz custos de utilização do sistema.
- » *Exemplo: LINUX*

2. Processos, Comun. e Sincr. – Tipos SOs



- Sistemas em Lote (*Batch*).
- Sistemas Multiprogramados.
- Sistemas de Tempo Compartilhado.
- Sistemas de Computadores Pessoais.
- Sistema Paralelos.
- Sistemas de Tempo Real.
- Sistemas Distribuídos.

2. Processos, Comun. e Sincr. – Tipos SOs

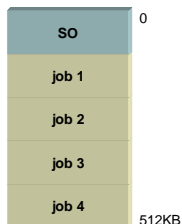


- Sistemas em Lote (*Batch/Off-line*)
 - » Operação via console.
 - » Entrada: Leitora de Cartões e Unidades de Fita.
 - » Saída: Impressoras de Linha, Unidades de Fita e Perfuradora de Cartões.
 - » Usuário não interage com o computador: ele prepara o *job* (tarefa) e entrega ao operador.
- Sistema Operacional era bastante simples
 - Transfere o controle de um *job* para outro.
 - Residente na memória.

2. Processos, Comun. e Sincr. – Tipos SOs



- Sistemas Multiprogramados
 - » Os Sistemas Operacionais começam a manter vários *jobs* na memória.
 - » Surge a Multiprogramação: onde há aumento de utilização da CPU pois sempre há um *job* sendo executado.



2. Processos, Comun. e Sincr. – Tipos SOs



- » Multiprogramação
 - Quando um *job* tem de executar uma operação de I/O a CPU ficaria ociosa.
 - Neste intervalo um outro *job* passa a ser executado, até que este por sua vez tenha de esperar I/O também, e assim por diante.
 - Quando o primeiro *job* termina a espera de I/O, ela tem a CPU de volta.
 - Desse modo a CPU nunca fica ociosa, desde que haja pelo menos um *job* para executar.

2. Processos, Comun. e Sincr. – Tipos SOs



- Sistemas de Tempo Compartilhado
 - » Não só o processador é compartilhado no sistema, a memória, periféricos, como discos e impressoras também.
 - » Cada processo tem um ambiente de trabalho próprio.
 - » Cada processo tem a impressão de que todo o sistema está dedicado exclusivamente a ele.
 - » Aumentam a produtividade dos seus usuários.
 - » Redução dos custos.

2. Processos, Comun. e Sincr. – Tipos SOs



- Sistemas de Computadores Pessoais
 - » O SO não tinha muita proteção contra o programa do usuário.
 - » Procuravam conveniência e capacidade de resposta.
 - » Não tinham eficiência de utilização de CPU e periféricos.
 - » Ex.: *MS-DOS* e *Apple Macintosh*.

2. Processos, Comun. e Sincr. – Tipos SOs



▪ Sistemas Paralelos

- » Mais de um processador que se comunicam entre si.
- » Compartilham barramento e relógio, e as vezes a memória e os periféricos.
- » São chamados de sistemas fortemente acoplados (*tightly coupled*).
- » Maior *throughput* – desempenho: vários processadores cooperam em uma mesma tarefa.
- » Compartilha periféricos, discos, e fontes de alimentação.
- » Aumenta a confiabilidade – tolerante a falhas.

2. Processos, Comun. e Sincr. – Tipos SOs



▪ Sistemas de Tempo Real

- » Limites de Tempo são bem definidos e fixos.
- » O sistema deve retornar o resultado correto a tempo de ser útil para o processo a que ele serve.
- » Exemplos:
 - Monitoramento de refinarias de petróleo.
 - Controle de tráfego aéreo.
 - Transações bancárias.

2. Processos, Comun. e Sincr. – Tipos SOs



▪ Sistemas Distribuídos

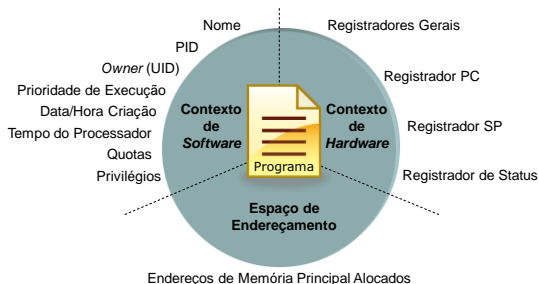
- » Sistemas Fracamente Acoplados (*Loosely Coupled*).
- » Coleção de processadores, conectados em rede, que não compartilham memória ou relógio.
- » Cada processador tem sua memória local.
- » Com a introdução da Web, a conectividade de rede torna-se essencial em um sistema de computação.

2. Processos, Comun. e Sincr. – Objetivos



- Os SOs foram desenvolvidos nos últimos 40 anos para alcançar dois objetivos:
 - » escalonar as atividades computacionais para garantir um bom desempenho do sistema de computação.
 - » fornecer um ambiente conveniente e amigável para o desenvolvimento e a execução dos programas.
- Para melhorar o desempenho surgiu a multiprogramação.
- Compartilhamento de tempo:
 - » muitos processos;
 - » muitos usuários e;
 - » interatividade.

2. Processos, Comun. e Sincr. – Estr. Processo



2. Processos, Comun. e Sincr. – PCB



- PCB (*Process Control Block*) ou também conhecido como Bloco de Controle de Processo:
 - » É a estrutura através da qual o processo é implementado pelo sistema operacional;
 - » Os PCBs de todos os processos ativos residem na memória principal em uma área exclusiva do S.O.

Ponteiros
Estado do processo
Nome do processo
Prioridade do processo
Registradores
Limites de memória
Lista de arquivos abertos
.
.
.

2. Processos, Comun. e Sincr. – Estado



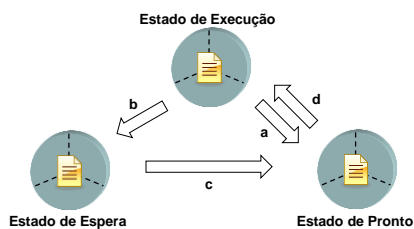
- Os processos passam por diferentes estados ao longo do seu processamento, em função de eventos gerados pelo SO ou pelo próprio processo;
- Um processo ativo pode encontrar-se em três diferentes estados:
 - » Execução (*running*);
 - » Pronto (*ready*);
 - » Espera (*wait*).

2. Processos, Comun. e Sincr. – Estado



- Um processo muda de estado durante o seu processamento em função de eventos:
 - » voluntários, originados por ele próprio;
 - » involuntários, originados pelo S.O.
- Basicamente, existem quatro mudanças de estado que podem ocorrer a um processo:
 - » Pronto → Execução;
 - » Execução → Espera;
 - » Espera → Pronto;
 - » Execução → Pronto.

2. Processos, Comun. e Sincr. – Estado

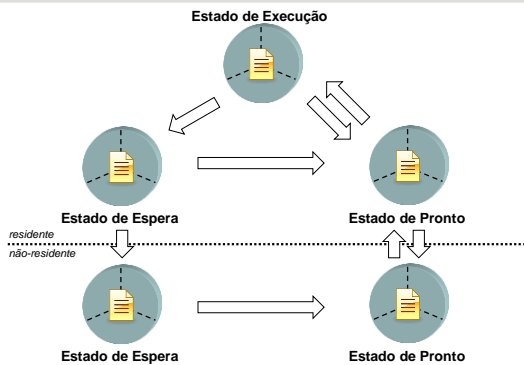


2. Processos, Comun. e Sincr. – Estado



- Um processo em estado de pronto ou de espera pode não se encontrar na memória principal;
- Isso ocorre quando não existe espaço suficiente para todos os processos na memória principal e parte do contexto é levado para a memória secundária;
- Essa técnica é conhecida como *swapping*;
- Neste caso, os processos em estado de espera e pronto podem estar residentes ou não-residentes na memória principal.

2. Processos, Comun. e Sincr. – Estado

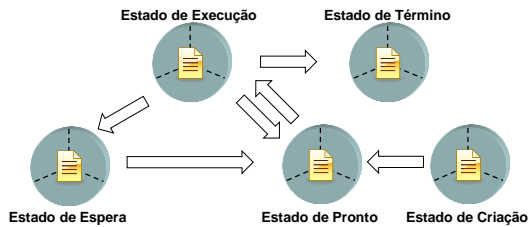


2. Processos, Comun. e Sincr. – Eliminação



- 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 para uso.
- No caso da eliminação, todos os recursos associados ao processo são desalocados e o PCB eliminado pelo SO.
- Além dos três estados vistos anteriormente, muitos SO estabelecem dois estados adicionais, para os momentos de criação e eliminação de um processo.

2. Processos, Comun. e Sincr. – Eliminação

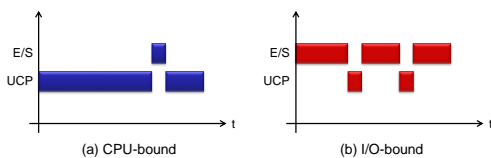


2. Processos, Comun. e Sincr. – Bound



- Um processo é definido como CPU-bound (ligado à CPU) quando passa a maior parte do tempo em estado de pronto ou execução, utilizando o processador.
 - » É encontrado em aplicações que efetuam muitos cálculos;
- Um processo é classificado como I/O-bound (ligado à E/S) quando passa a maior parte do tempo em estado de espera, pois realiza um elevado número de operações de E/S;
 - » É encontrado em aplicações comerciais, que se baseiam em leitura, processamento e gravação.

2. Processos, Comun. e Sincr. – Bound



2. Processos, Comun. e Sincr. – FG & BG



- Um processo *foreground* (FG) é aquele que permite a comunicação direta do usuário com o processo durante o seu processamento;
 - » É a base do processamento interativo, com teclado, mouse e monitor;
- Um processo *background* (BG) é aquele onde não existe a comunicação com o usuário durante o seu processamento;
 - » Neste caso, os canais de E/S não estão associados a nenhum dispositivo de E/S interativo, mas em geral a arquivos de E/S.

2. Processos, Comun. e Sincr. – FG & BG



Processo Foreground (FG)



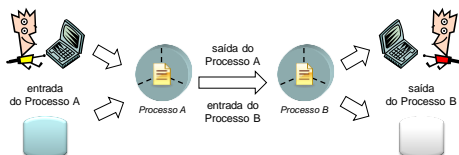
Processo Background (BG)



2. Processos, Comun. e Sincr. – Pipe



- É possível associar o canal de saída de um processo ao canal de entrada de outro.
- Neste caso, dizemos que existe um *pipe* ligando os dois processos.



2. Processos, Comun. e Sincr. – Proc. Independ.



- Processos independentes são maneiras diferentes de implementar a concorrência dentro de uma aplicação.
- Busca-se subdividir o código em partes para trabalharem de forma cooperativa.
- A concorrência proporciona melhoras no desempenho das aplicações e benefícios aos usuários.
- É a maneira mais simples de implementar a concorrência em sistemas multiprogramáveis.
- Não existe vínculo do processo criado com o processo criador.
- Exige a alocação de um PCB, possuindo contextos de hardware, contextos de software e espaços de endereçamentos próprios.

2. Processos, Comun. e Sincr. – Subprocessos

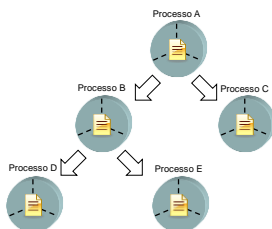


- São processos criados dentro de uma estrutura hierárquica.
- O processo criador é denominado *processo-pai* e o novo processo é chamado *subprocesso* ou *processo-filho*.
- Um subprocesso pode criar outras estruturas de subprocessos.
- Existe dependência entre o processo criador e o subprocesso.
- Os subprocessos possuem o seu próprio PCB.

2. Processos, Comun. e Sincr. – Subprocessos



Hierarquia de Processos e Subprocessos

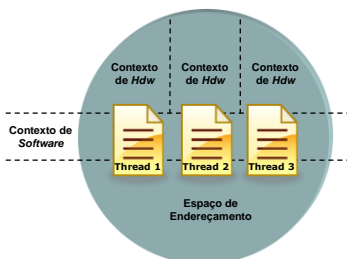


2. Processos, Comun. e Sincr. – Thread



- O conceito de *thread* foi introduzido na tentativa de reduzir o tempo gasto em criação, eliminação e troca de contexto de processos nas aplicações concorrentes.
- Economiza recursos do sistema como um todo;
- Um único processo pode suportar múltiplos *threads*, cada qual associado a uma parte do código da aplicação.
- Não é necessário haver vários processos para a implementação da concorrência.
- Todas as threads utilizam o PCB do processo que as inicializou.

2. Processos, Comun. e Sincr. – Multithread



2. Processos, Comun. e Sincr. – Multithread



- O compartilhamento de recursos entre processos pode ocasionar situações indesejáveis que podem comprometer a execução das aplicações.
- O SO deve oferecer mecanismos que sincronizem as execuções dos processos concorrentes, garantindo o processamento correto dos programas.



Sistemas Distribuídos – Aula 02

clayton.valdo@anhanguera.com