

# Introdução

- O que é um Sistema Operacional?
- Onde são (e foram) usados?
  - Computadores pessoais
  - Mainframes
  - Computadores Multiprocessados
  - Computadores Distribuidos
  - Clusters
  - Sistemas Computacionais de Tempo-Real
  - Sistemas portáteis – ARM

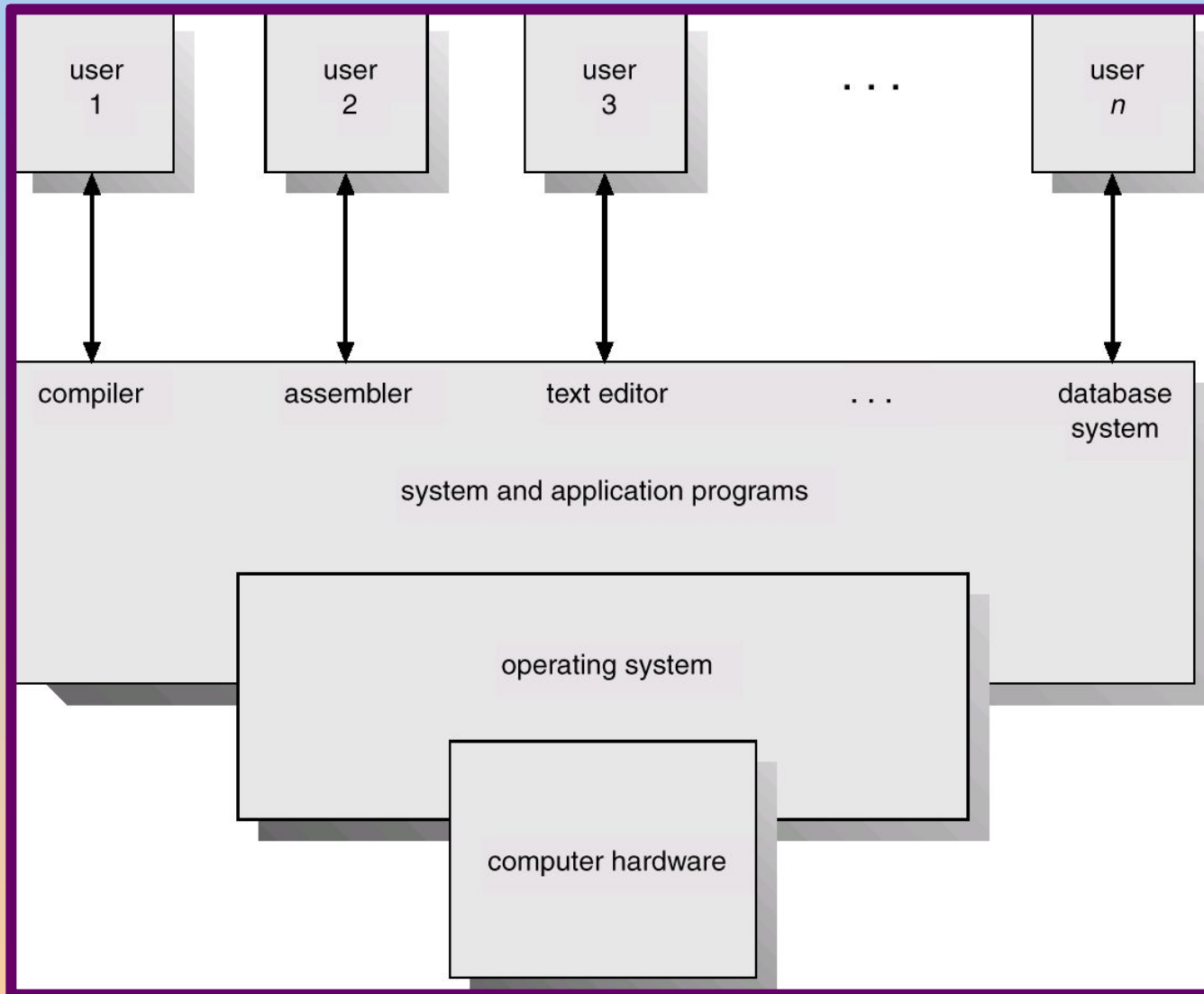
# O que é um Sistema Operacional?

- Um programa que age como um intermediário entre o usuário e o hardware
- Objetivos dos sistemas operacionais:
  - Executar os programas e resolver os problemas dos usuários mais simplesmente.
  - Tornar o sistema computacional mais conveniente de se usar.
  - Usar o hardware computacional de maneira mais eficiente

# Componentes de um Sistema Computacional

1. **Hardware** – provê recursos computacionais básicos (CPU, memória, dispositivos de E/S).
2. **Sistema Operacional** – controla e coordena o uso do hardware pelos vários programas e usuários.
3. **Aplicativos** – programas que definem como os recursos computacionais são usados para se resolver os problemas dos usuários (compiladores, bancos de dados, jogos, programas empresariais).
4. **Usuários** (pessoas, máquinas, outros computadores)

# Componentes do Sistema



# Definição de Sistema Operacional

- Gerenciador de Recursos – gerencia e aloca recursos de hardware (dispositivos de E/S, CPU, ...) e software.
  - BÔNUS: Relativa independência do hardware
- Controlador de Programas – controla a execução dos programas dos usuários.
- Fornece um ambiente (ou arquitetura) estendido. O usuário tem mais poderes sobre o hardware através do SO.

Núcleo (ou Kernel) – programa que está constantemente em execução e que executa as funções acima).

# Evolução dos Sistemas Computacionais

- A evolução do hardware foi influenciada pelos sistemas operacionais e vice-versa.
- O SO surgiu para facilitar o uso do hardware
- Criou-se uma arquitetura sem a qual não conseguiríamos utilizar os sistemas computacionais eficientemente
  - Arquivos
  - Programas e processos
  - Interfaces texto e gráficas

# SOs de Mainframes

- Computadores grandes e lentos
  - Se comparados com os atuais
- Dispositivos de E/S primitivos
  - Leitora e Perfuradora de cartões
  - Impressoras de linha
  - Fitas magnéticas
- Os programadores enviavam seus *jobs* para o operador que os organizava em lotes (*batches*) de jobs semelhantes.
  - Diminuir tempo gasto com setup da máquina
  - Ausência de interação com o usuário durante a execução

# SOs de Mainframes (2)

- Problemas

- Um job só poderia ser executado assim que o outro terminasse
- A CPU ficava muito ociosa esperando pelos dispositivos de E/S
  - Leitora de cartões/Impressora de linha □ Lentos

- Surgimento dos discos rígidos

- Jobs carregados dos cartões para o disco rígido para depois serem executados
- Saída dos jobs era colocada nos discos para serem impressas depois (CPU não espera pela impressora)
- SPOOL: **S**imultaneous **P**eripheral **O**peration **O**n-**L**ine
- Operação Simultânea de Periféricos On-Line



# SO de Mainframes (3)

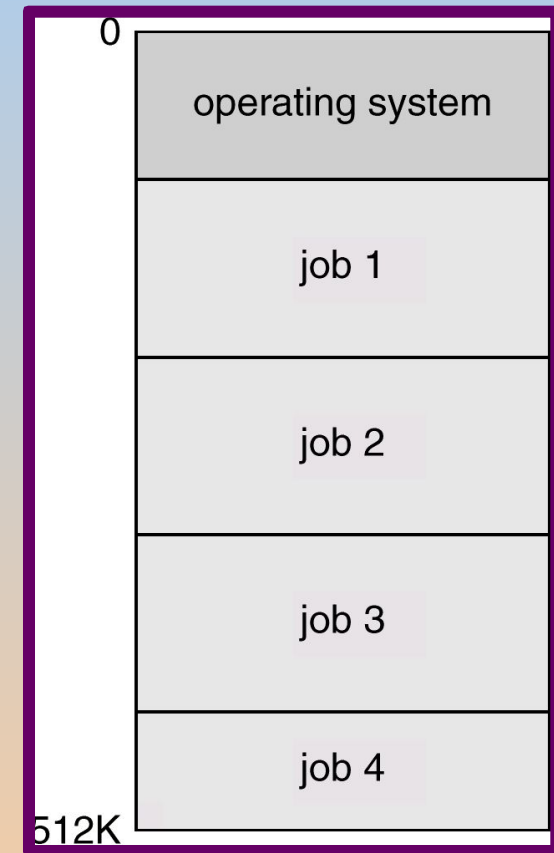
- Seqüenciamento automático de jobs
  - Devido ao surgimento do SPOOL
  - Ao custo de um pouco de espaço em disco, processador e periféricos são utilizados mais eficientemente
  - Primeiro SO rudimentar
- Monitor residente
  - Monitor controla a máquina inicialmente
  - Monitor transfere controle para job
  - Quando job termina, o controle volta para o monitor



Layout Memória de um Mainframe

# Sistemas Batch Multiprogramados

- Vários jobs são mantidos na memória e a CPU é multiplexada entre eles
  - Melhor aproveitamento do processador
  - Elimina espera ociosa por E/S
  - Surgem os seguintes conceitos:
    - Processo: Programa e todas as estruturas de dados associadas à sua execução
    - Troca de contexto: Processo pelo qual o processador deixa de executar um processo para executar outro



# Sistemas Batch Multiprogramados (2)

Serviços providos por SOs multiprogramados:


- Rotinas de E/S
- Gerenciamento de memória
- Agendamento de CPU: escolher entre vários processos prontos qual vai rodar.
- Alocação de dispositivos.

Primeiro tipo de SO onde decisões são tomadas pelos usuários

OBS: A troca de processo em execução é governada somente pelas requisições de E/S (importante)

# Sistemas de Compartilhamento de Tempo da CPU

- Visa permitir que múltiplos usuários interajam com o sistema computacional
  - Surge o uso do teclado e monitor, dos editores de texto e dos depuradores de programas.
  - Surge a necessidade de um sistema de arquivos
  - Início da década de 70
- O tempo da CPU é dividido entre os processos
- Um processo pode perder CPU se:
  - Ele efetuar uma requisição de E/S ou
  - Se o seu *quantum* se acabar



Quantidade máxima de tempo que um processo pode executar sem ser interrompido

# Sistemas de Compartilhamento de Tempo (2)

- Surge o conceito de memória virtual:
  - Extensão da memória principal (ou *core*)
  - Os processos podem não estar totalmente na memória principal ao serem executados
  - O SO decide quando um processo deve estar ou não na memória principal
- Além dos serviços do SO multiprogramado, o SO de tempo compartilhado deve prover:
  - Comunicação entre processos
  - Mecanismos de sincronização e de proteção à memória

# Sistemas de Computadores Pessoais

- Surgiu do final da década de 70
- Baixo poder computacional: Sistemas operacionais simples
  - Ausência de mecanismos de proteção, não possuíam suporte à multitarefa nem multiusuário.
- Atualmente existem sistemas operacionais para PCs com as mesmas características encontradas nos SOs para máquinas de grande porte
  - Atualmente estas características são importantes devido ao alto grau de difusão da Internet.

# Sistemas de Computadores Pessoais

- Características importantes para SOs de PCs
  - Conveniência ao usuário
  - Responsividade (alta interatividade)
  - Suporte à uma quantidade elevada de dispositivos de E/S
  - Fácil utilização

# Sistemas de Tempo Real (RTOS)

- Estão sob restrições de tempo rígidas e bem definidas.
- Ex: VxWorks (<http://www.windriver.com>)
- Podem ser instalados em computadores ou fazer parte do equipamento (software embarcado)
- Utilizados para se controlar dispositivos em aplicações específicas como sistema de controle industrial (robôs), equipamentos de comunicação (celulares), sistemas de coleta de dados e manutenção da vida.
- São específicos para determinados tipos de hardware.



# Dispositivos Móveis

- Smartphones
- Processadores ARM
- Ex: Android, IOS, FirefoxOS
- Problemas:
  - Memória limitada
  - Processadores mais lentos
  - Displays pequenos
  - Economia de bateria

# Sistemas Paralelos

- São sistemas com mais de um processador
- Sistemas altamente acoplados
  - Compartilhamento de memória e relógio
  - Comunicação através de memória compartilhada
- Vantagens do sistema paralelo:
  - Maior performance
  - Economia
    - Espaço, custos de manutenção,...
    - *Total Cost of Ownership (Custo Total de Propriedade)*
  - Maior confiabilidade
    - Degradação graciosa
    - Sistemas tolerantes a falhas

# Sistemas Paralelos (2)

- *Multiprocessadores Simétricos (SMP)*
  - Kernel na memória compartilhada.
  - Cada processador executando alguma funcionalidade do SO ou processo do usuário.
  - Vários processos em execução sem degradação
  - SOs modernos suportam SMP (Unix, Linux, Windows,...)
    - Kernel do linux suporta NUMA
- *Multiprocessamento Assimétrico*
  - Também conhecido como “Processor Farms”
  - Cada processador cuida de uma tarefa específica.
  - Processador mestre aloca processos para trabalhadores.
  - Mais comum em sistemas grandes

# Sistemas Distribuídos

- Computação distribuída entre processadores fisicamente dispersos.
- Computação em “Grade”: **Grid Computing**
- *Sistemas Fracamente Acoplados* – cada processador tem sua memória local própria. A comunicação pode se dar através de redes, linhas telefônicas ou barramentos.
- Vantagens
  - Compartilhamento de Recursos
  - Aumento de performance
  - Confiabilidade

# Sistemas Distribuídos (2)

- Requer infraestrutura de rede.
- Relação entre processadores:
  - Cliente-Cliente (Peer-to-Peer)
  - Cliente-Servidor (Client-Server)
- Difícil se conseguir ganhos de performance
  - Taxas de transferência e latência das redes não são suficientes
  - Depende da granularidade dos processos e da
  - Relação Computação/Comunicação

# Clusters

- Vários processadores que compartilham armazenamento e estão conectados por uma rede de alta velocidade.
- Alta confiabilidade.
- Ex: OpenMOSIX (<http://openmosix.sf.net>), Linux – qualquer distribuição (<http://www.linux-ha.org>), Windows Server
- *Cluster Assimétrico*: um servidor em operação e outro em espera (standby).
- *Cluster Simétrico*: todos os N servidores executando a aplicação.