BCC722

Programação de Sistemas em Tempo Real

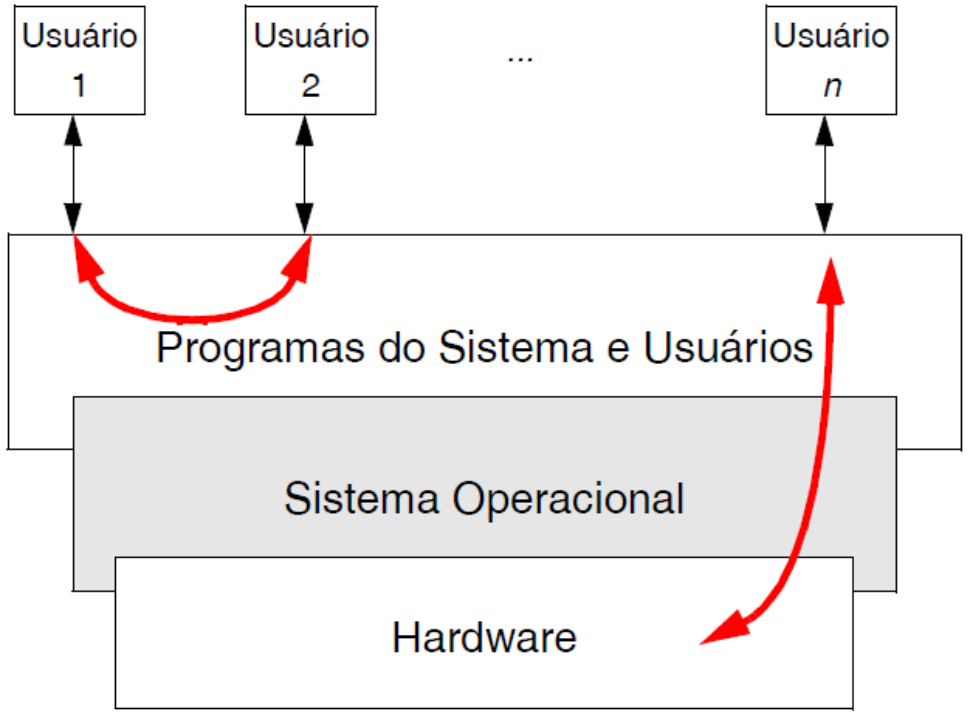
Introdução

Prof. Charles Garrocho

O que é um sistema operacional?

Um programa governo:

- Controla a interação entre:
  - Usuários e hardware
  - Os diversos usuários



O que é um sistema operacional?

Um programa ilusionista:

- Prove abstrações:
  - A interface do usuário não corresponde ao hardware existente.
  - Permite executar o mesmo programa em vários ambientes diferentes.
- Coordena recursos:
  - Controla o acesso a recursos de modo a maximizar seu uso e a proteger usuários uns dos outros:
  - Memória
  - CPU
  - Dispositivos de E/S

O que é um sistema operacional?

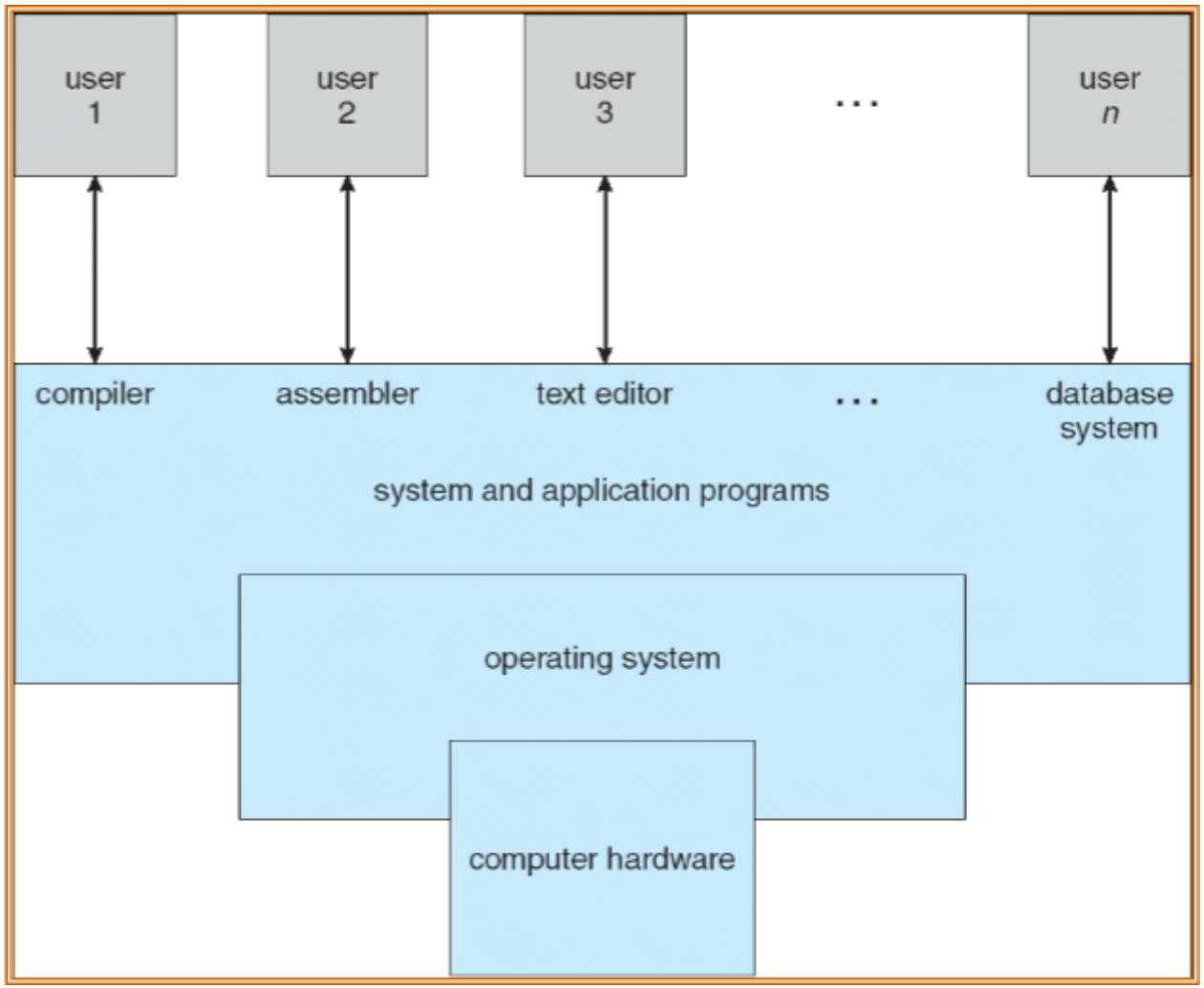
- Um programa que age intermediário entre o usuário e o hardware do computador

- Objetivos:
  - controlar a execução dos programas do usuário;
  - tornar os sistemas computacionais mais simples de serem usados.
  - Fazer uso do hardware de forma “eficiente”

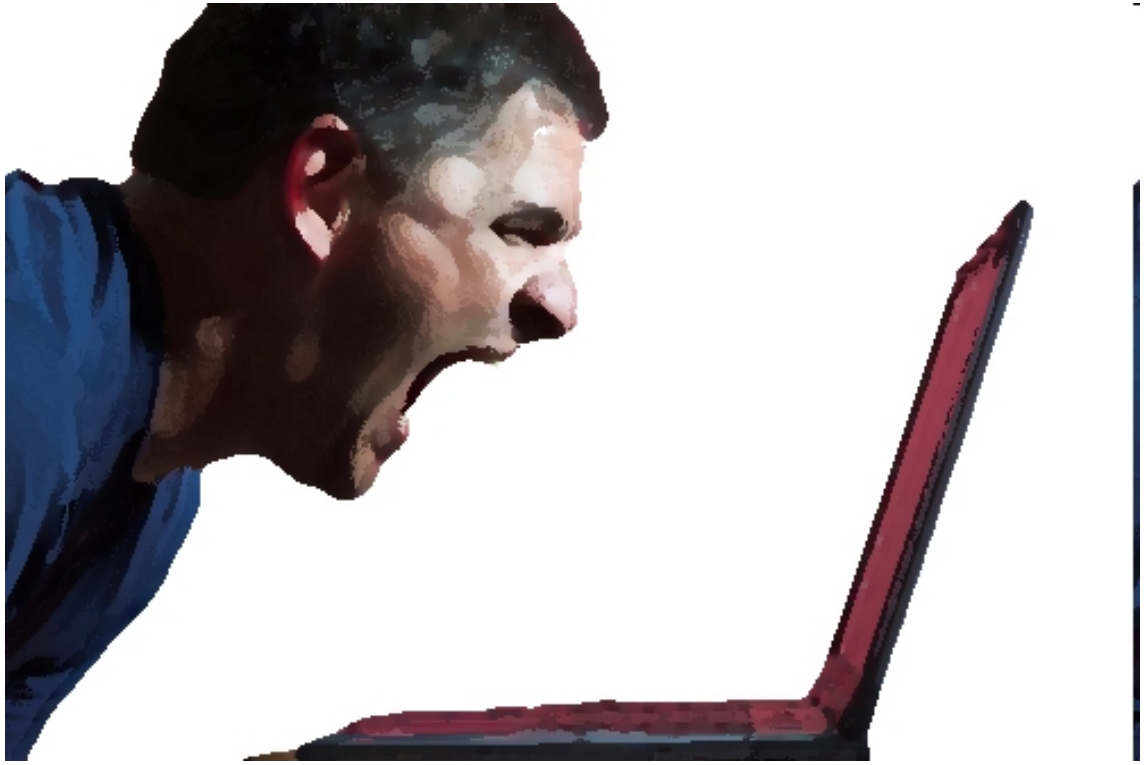
Componentes de um sistema computacional

- Hardware – recursos básicos de computação (CPU, memória, dispositivos de E/S)
- Sistema Operacional – controla e coordena o uso do hardware pelos vários programas aplicativos dos vários usuários
- Programas aplicativos – definem como os recursos são usados para solucionar os problemas dos usuários
- Usuários (pessoas, máquinas, outros comp.)

Componentes de um sistema computacional



SO do ponto de vista do usuário



Usuário enxerga a interface com o sistema, velocidade, capacidade de armazenamento e funcionalidades de periféricos

SO do ponto de vista do Hardware

- Alocador de recursos – gerencia e distribui recursos do hardware
- Programa de controle – controla a execução dos programas dos usuários e a operação dos dispositivos de E/S
- *Kernel* – o programa essencial que executa todo o tempo (o restante seriam aplicativos)

Funções Principais

- Coordenação: Permite múltiplas aplicações/usuários trabalhar simultaneamente de forma eficiente.
- Concorrência
- Proteção de memória
- Arquivos
- etc.

- Padronização de Serviços: Padronização da interface a fim de simplificar a programação e uso.

Por que estudar SO?

- Entender SOs = entender computadores:
  - Ajuda a usá-los.

- SO conectam hardware e software:
- Entendimento releva limitações e pontos fortes:
  - Exemplo, porque não se pode usar Windows ou X-Windows para se controlar uma usina nuclear?

- SOs combinam conceitos de praticamente todas as áreas da computação. Seu conhecimento fornece base sólida para construção de sistemas complexos
  - Linguagens
  - Hardware
  - Estruturas de dados
  - Algoritmos
  - Teoria

História de SOs

Fase 0: Não existem SOs.

- Programação através de chaves no painel.
- Conjuntos de cartões manualmente carregados para executar os programas
- Usuário presente todo o tempo; toda atividade é seqüencial:
  - nenhuma sobreposição entre computação, E/S e tempo de pensar do usuário
- Problema: muita espera.
  - usuário tem que esperar pela máquina
  - máquina tem que esperar pelo usuário
  - todos têm que esperar pela leitora de cartões

História de SOs

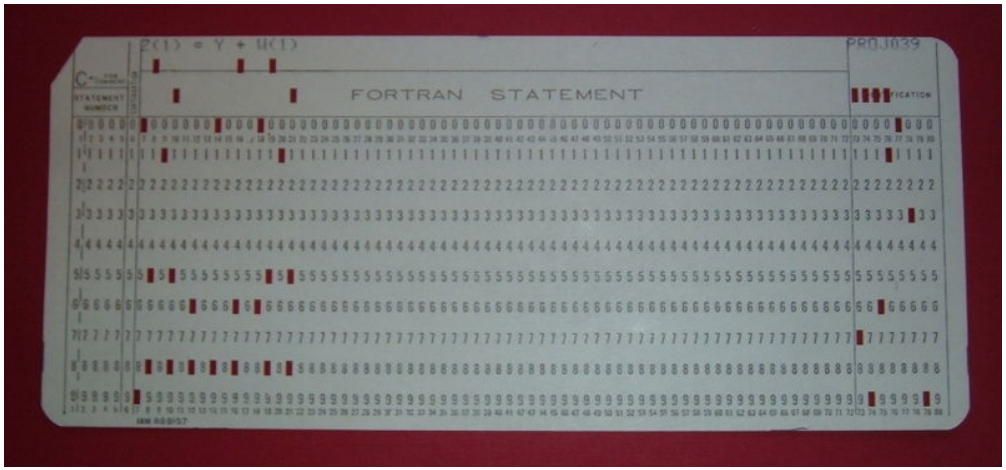
- Primeiros sistemas desenvolvidos: **Mainframes**
- Voltados para a execução de aplicações comerciais e científicas
- Diversos tipos ao longo do tempo:
  - programação direta (sem S.O.)
  - monitores de execução (quasi-S.O.)
  - batch
  - multiprogramados
  - tempo compartilhado

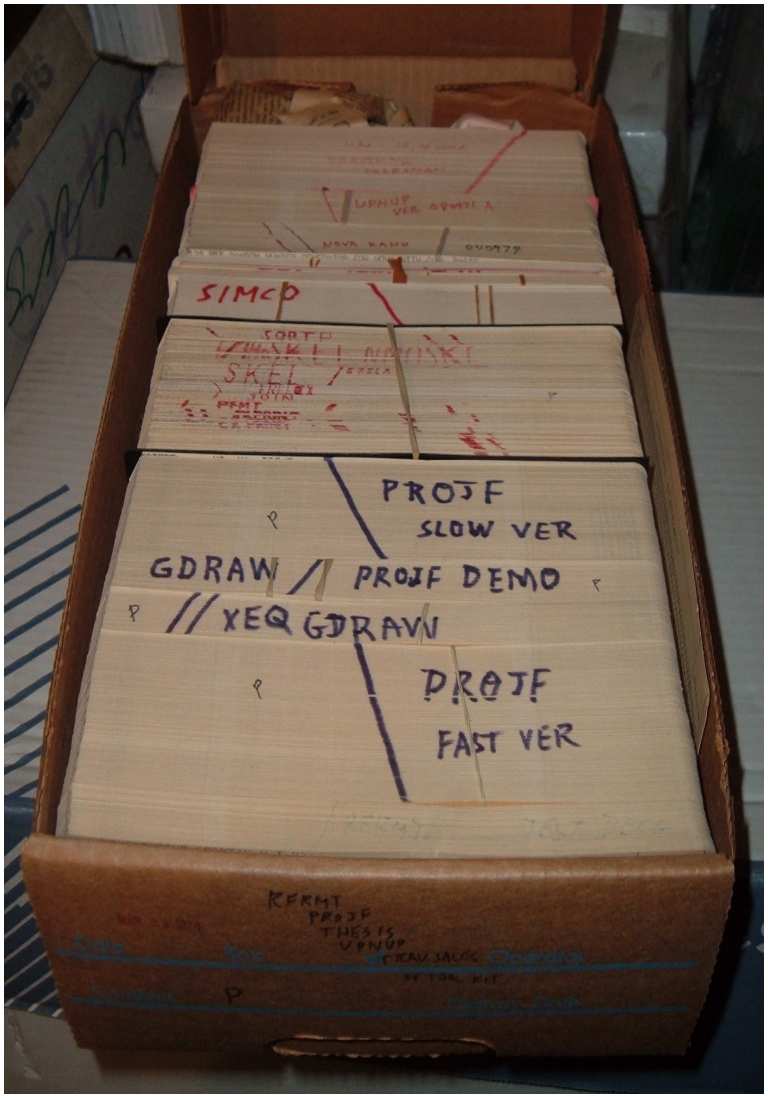
História dos SOs

Programação direta: Primeiros computadores com controle direto do hardware com chaves e cabos (nenhum SO)

História dos SOs

Programação passou a usar cartões perfurados



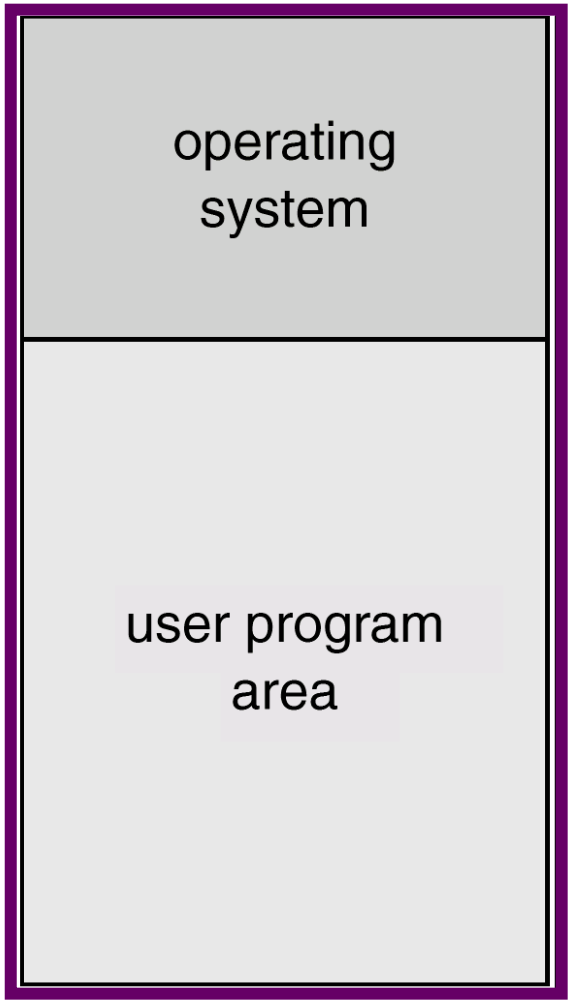


- O programador controlava o computador diretamente
  - Sem SO: apenas um "monitor" mínimo

História de SOs

Sistemas Batch

- Primeiros sistemas operacionais, rudimentares
- “Monitor residente”:
  - inicializa o sistema, controla o HW;
  - aloca recursos;
  - transfere controle para uma tarefa;
  - ao seu final, reassume o controle
- Problema: grande ociosidade



História de SOs

Sistemas Batch

História de SOs

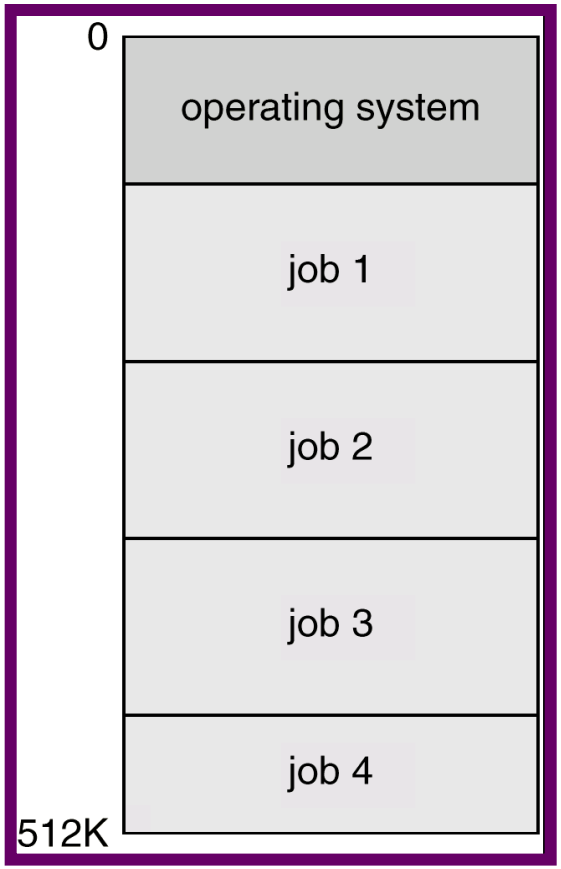
Fase 1: Primeiros SO – Multiprogramação

- Vários programas são carregados na memória simultaneamente
  - O OS escolhe um deles e executa este trabalho an CPU
- Utilização mais eficiente do computador:
  - Permite a sobreposição de computação, E/S, tempo do usuário.
- Complicado!!!
  - OS/360 liberado com 1000 bugs!
  - Escalonamento de jobs;
  - Escalonamento de CPU;
  - Proteção;
  - Gerenciamento de memória
- Problema:
  - Usuários ainda esperam pelo computador.

História de SOs

Sistemas Multiprogramados

- Tentam resolver o problema da ociosidade carregando vários programas ao mesmo tempo na memória
- Quando um programa pára esperando por dados, outros podem ser executados



História de SOs

Sistemas Multiprogramados

IBM 1401, 1961

IBM 7094, 1962

História de SOs

Características do SO para permitir multiprogramação

- Rotinas de E/S fornecidas pelo SO
- Gerência de memória: o SO deve alocar memória para vários programas (jobs)
- Escalonamento da CPU: o SO escolhe entre os jobs disponíveis qual deve executar
- Alocação de dispositivos: o SO define quem pode acessar qual dispositivo

História de SOs

Fase 2: Time Sharing -- Interatividade

- Uma extensão da multiprogramação
  - Nenhum processo executa mais do que uma faixa de tempo, chamada de quantum.
  - Um temporizador é setado a cada momento que um processo entra na CPU
  - Se o temporizador termina, outro processo é escolhido
- A CPU é compartilhada por vários jobs mantidos na memória (multiprogramação)
- Comunicação on-line entre usuário e sistema: S.O. procura por comandos do usuário

História de SOs

Fase 2: Time Sharing -- Interatividade

IBM 3278, 1972

IBM 3032, 1977

História de SOs

Fase 2: Time Sharing -- Interatividade

- Mais complicados!!!
  - Multics anunciado em 63, concluído em 69!
  - Unix foi ``simplificado'' a partir de Multics
- Escalonamento e proteção passam a ser críticos:
  - Tempo de resposta
  - Hackers
- Dados tem que poder ser acessados continuamente
  - Sistema de arquivos
- Base dos SOs modernos

História de SOs

Fase 3: Computadores pessoais:

- Inicialmente baratos mas com pouca potência
- SOs foram simplificados correspondentemente
- DOS/Windows: proteção totalmente eliminada!
  - Problema não muito crítico:
    - Sistema normalmente tem um só usuário.
- Eventualmente as coisas melhoraram:
  - Windows NT, etc.
  - Linux, FreeBSD, etc.
  - Macs

SOs Modernos

- Enormes:
  - Centenas de milhares de linhas de código;
  - 100..1000 homens-ano de desenvolvimento!

- Complexos:
  - Idiossincrasias de hardware;
  - Tipos diferentes de usuários;
  - Desempenho

- Mal compreendidos:
  - Muito grandes para serem compreendidos por uma pessoa
  - Nunca estão completamente sem erros
  - Comportamento difícil de prever

História de SOs

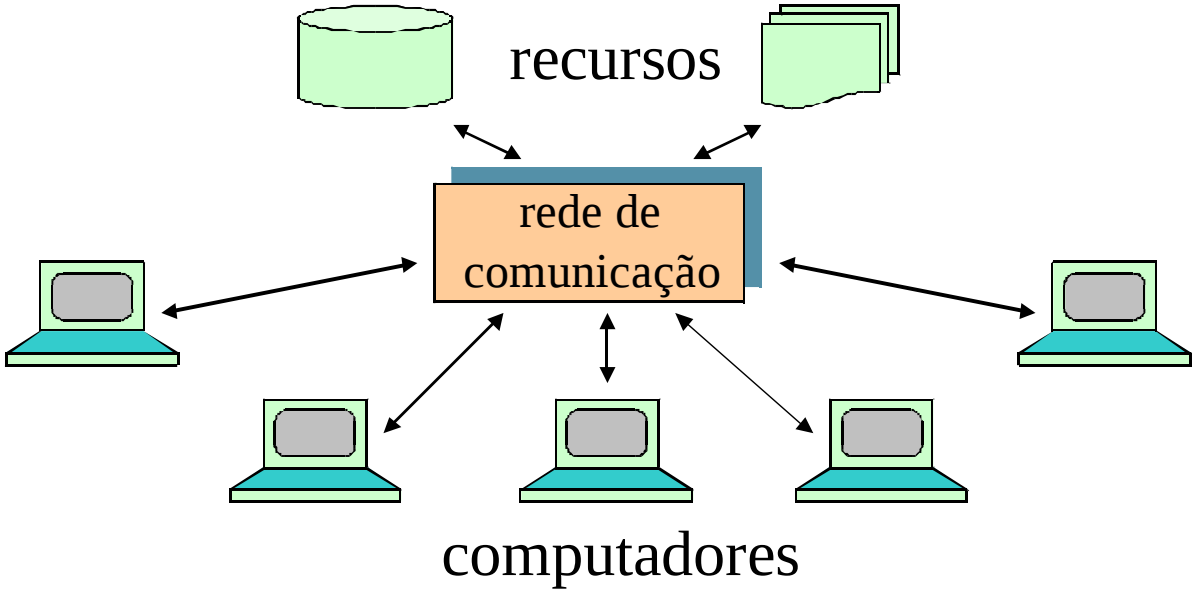
- **Fase 4: Sistemas paralelos e distribuídos**

- Mas o mundo não é mais o mesmo:
  - Menos de um usuário por sistema!
  - Multiprocessadores
  - Acesso à rede torna-se importante:
    - Os recursos usados freqüentemente não são locais e são acessados via rede:
    - Arquivos
    - Impressoras

- Você sabe que está usando um sistema distribuído quando:
  - Não consegue trabalhar porque um computador que você não está usando e do qual nunca ouviu falar está fora do ar.

Sistemas Distribuídos

- Distribuem a computação entre diversos processadores fisicamente independentes
- Podem se organizar como cliente-servidor ou par-a-par (peer-to-peer)



Clusters

- Sistemas distribuídos com uma rede local rápida e uniformidade de operação (SW)
- Diversos processadores compartilham recursos
- Compartilhamento de dispositivos de armazenamento (dados)

Sistema de tempo real

- É um sistema destinado à execução de múltiplas tarefas onde o tempo de resposta a um evento (externo ou interno) é pré-definido.

- Por exemplo, em um hospital, o sistema que monitora os batimentos cardíacos de um paciente deve alarmar os médicos caso haja alteração nos batimentos.

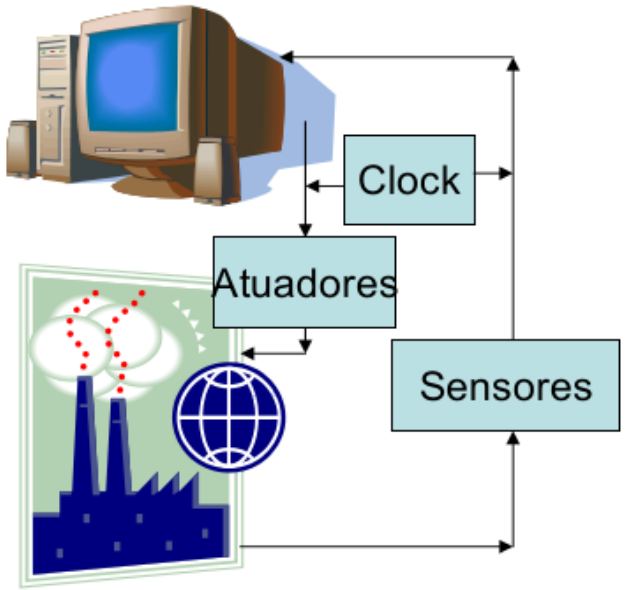
Os STR possuem duas grandes

- Classe A: Quando o tempo de resposta variável é admissível (sistema on-line)
  - Reserva de passagem, automação bancária.

Os STR possuem duas grandes

– Classe B: Quando estimulados por um evento, deve fornecer uma resposta em tempo finito e especificado.

- Controle e supervisão de processos industriais.



Sistemas de Tempo Real Críticos e Não-Críticos

Os STR são classificados, basicamente, em:

- **Críticos (hard RTS - também chamados de rígidos):** é aquele que tem um comportamento determinístico, ou seja, o prazo para execução de uma tarefa (deadline) não pode ser violado. Se o sistema de um freio ABS, por exemplo, falhar ou demorar demais para responder, uma pessoa poderá se machucar.

Sistemas de Tempo Real Críticos e Não-Críticos

Os STR são classificados, basicamente, em:

- **Não-Críticos (soft RTS - também chamados de moderados):** é aquele que também tem o tempo como parâmetro fundamental, mas uma falha é aceitável. O sistema que funciona em um leitor de DVD não é crítico, pois o não cumprimento de uma tarefa em resposta e um evento em um determinado intervalo de tempo não provoca danos irreversíveis

Sistemas pessoais/de mão

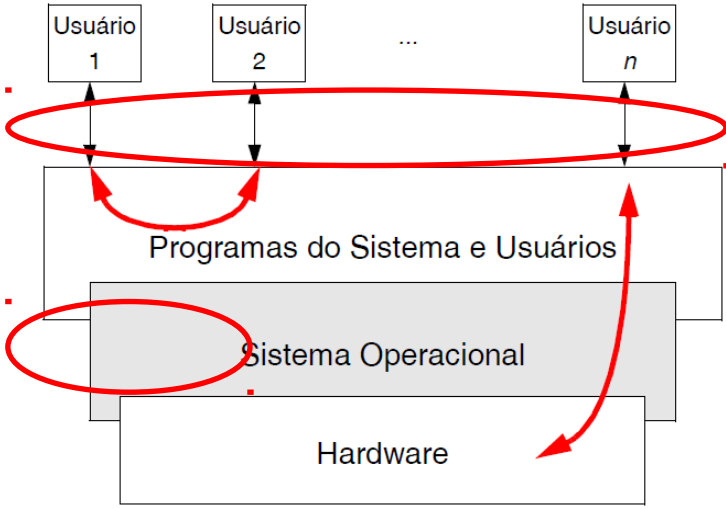
- Personal Digital Assistants (PDAs)
- Telefones celulares (ex. iPhone, Blackberry, Android)
- Já não são tão diferentes dos desktops:
  - Boa memória e capacidade de expansão;
  - Processadores relativamente rápidos;
  - Diferentes tamanhos de telas.

Componentes de SO

- Os pontos principais de um SO são:
  - Gerência de processos
  - Comunicação entre processos
  - Gerência de memória
  - Sistema de arquivos
  - Sistema de Entrada/Saída
  - Sistema de Proteção

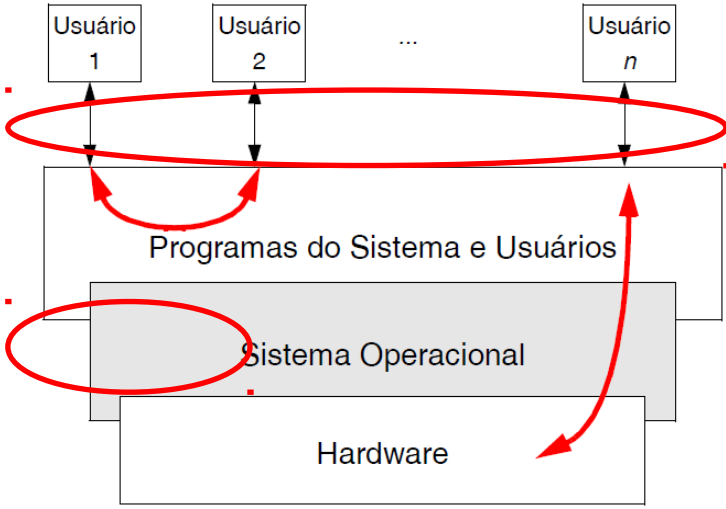
Gerência de processos

- Permite a execução simultânea de vários programas;
- Gera a ilusão de que cada programa roda sozinho;
- Um processo consiste de todo o ambiente de execução de um programa:
  - memória, registradores, etc.
- O escalonador é o programa que a cada instante determina qual processo estará executando
  - O programa mais importante do SO!

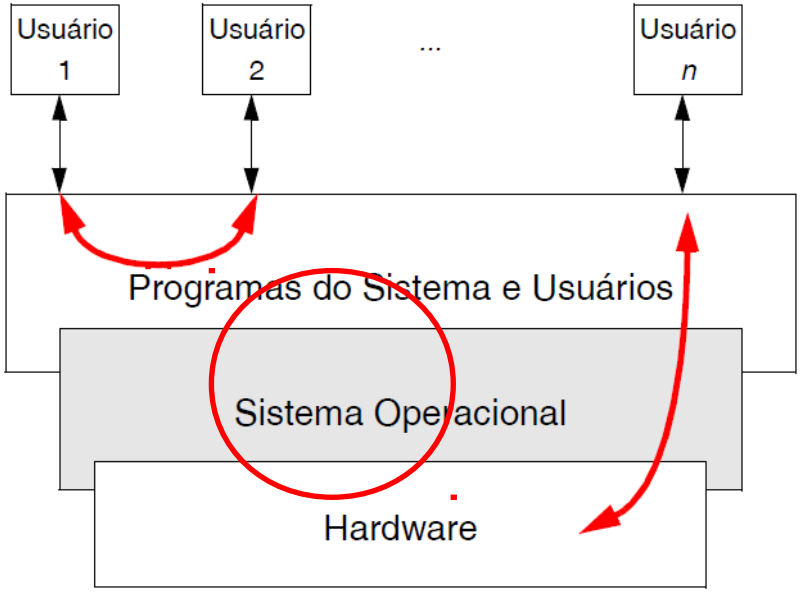


Comunicação entre Processos

- Permite que processos do usuário e do SO se comuniquem e sincronizem
- Extremamente complicado:
  - Comunicação/sincronização são a origem da maioria dos erros em SOs!
- Diversas primitivas existem:
  - Semáforos
  - Memória compartilhada
  - Troca de mensagens
  - ...

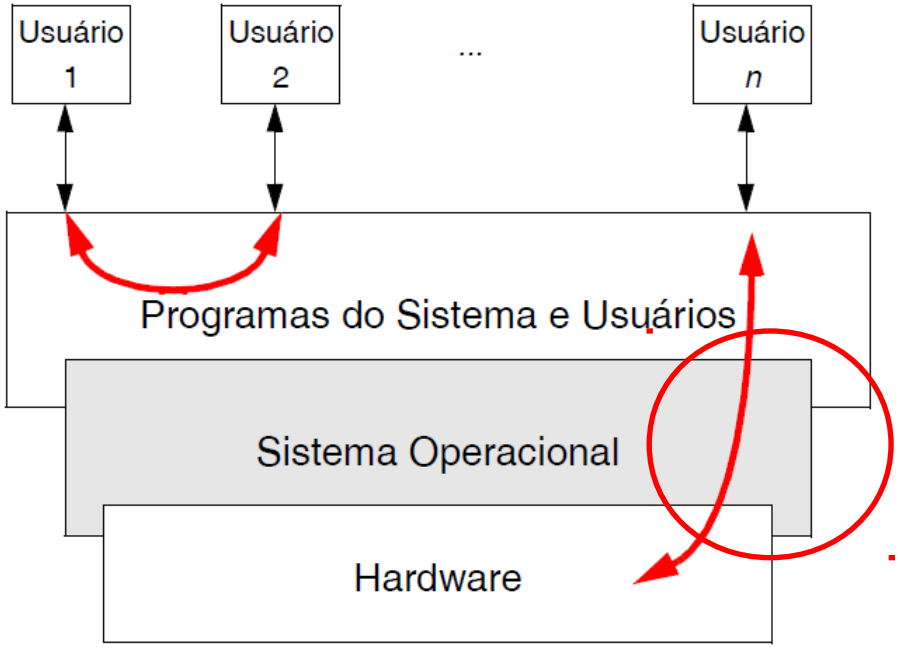


Gerência de Memória



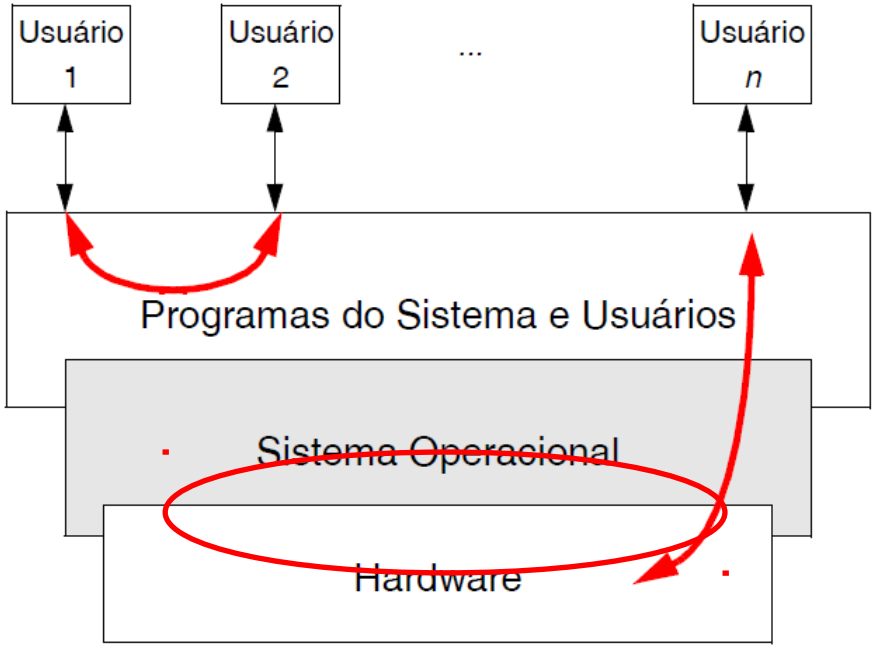
- Prove a ilusão de que a memória é infinita:
  - O programa endereça memória virtual;
  - A memória real é dividida em páginas;
  - Usa-se tabela páginas virtuais ↔ reais;
  - A memória virtual é muito maior do que a memória real.

Sistemas de Arquivos



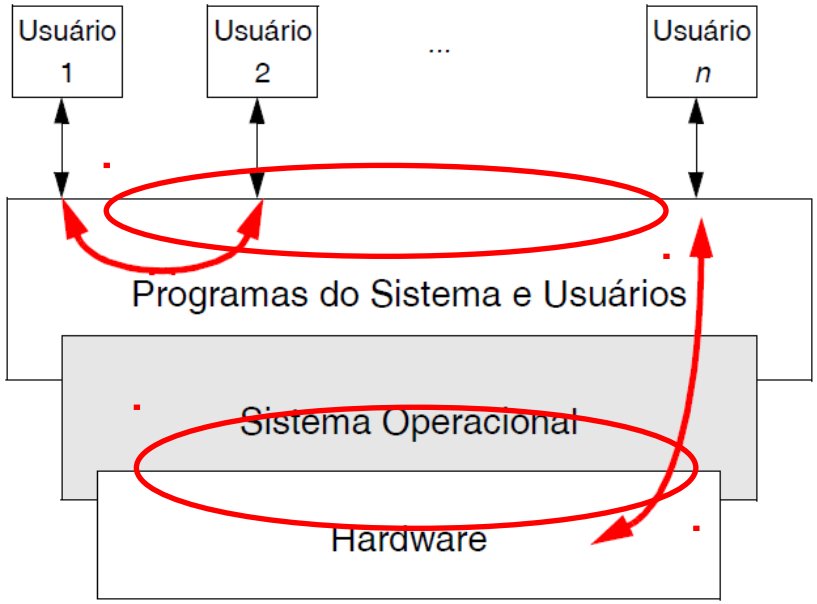
- Permite a utilização interativa
- Estrutura hierárquica de diretórios

Sistemas de Entrada e Saída



- Controla:
  - Discos;
  - Impressoras;
  - Acesso a rede;
  - ...
- Todo acesso é através do SO.

Sistemas de Proteção



- Impede que usuários se atrapalhem. Operações erradas podem afetar outros usuários e o SO:
  - Usuários não acessam E/S
  - Usuários não acessam memória fora da sua memória virtual

- Dois modos de operação:
  - SO acessa todos os recursos
  - Usuários têm acesso restrito