



# SISTEMAS OPERACIONAIS

**Roberto Willrich**

**INE- CTC-UFSC**

**E-Mail: [willrich@inf.ufsc.br](mailto:willrich@inf.ufsc.br)**

**URL: <http://www.inf.ufsc.br/~willrich>**

**Versão 2004.1/Profa. Lúcia**

# Introdução

- **Porquê existem os sistemas operacionais?**

- **No computador**

- Diversas atividades competem os recursos da máquina
      - uma máquina pode ser conectada a diversos terminais atendendo a diversos usuários
      - um PC pode realizar diversas atividades ao mesmo tempo:
        - » impressão, edição, anti-virus, ...
    - Competição exige um alto grau de coordenação
      - para evitar que atividades independentes interfiram umas com as outras
      - garantindo a comunicação eficiente e confiável entre atividades interdependentes

# O que é um sistema operacional?

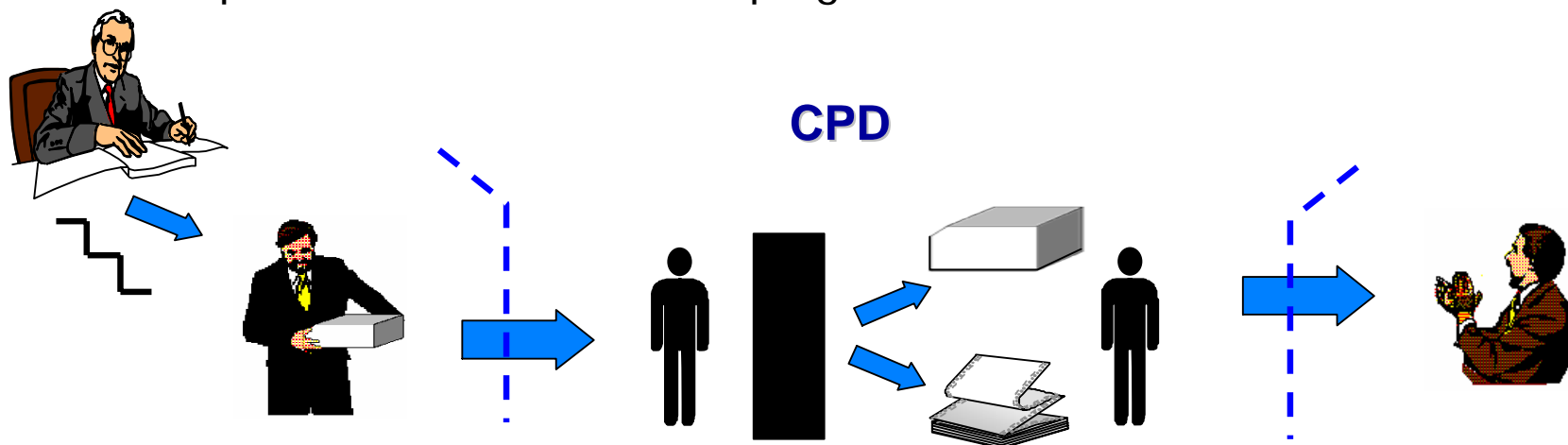
- **Visão top-down:**
  - uma camada de software que apresenta para o usuário uma visão modificada do hardware
- **Visão bottom-up:**
  - um gerenciador de recursos
  - sob esta ótica deve prover acesso controlado a:
    - processadores, memória, dispositivos de E/S, etc
- **O Sistema Operacional tem as seguintes características:**
  - Definição de interface com o usuário;
  - Compartilhamento do hardware entre os usuários;
  - Permite aos usuários compartilhar dados entre si;
  - Escalonamento de recursos entre usuários;
  - Facilidades de Entrada e Saída;
  - Recuperação de erros.



# **GERAÇÕES DE SISTEMAS OPERACIONAIS**

# Segunda Geração 1955-65

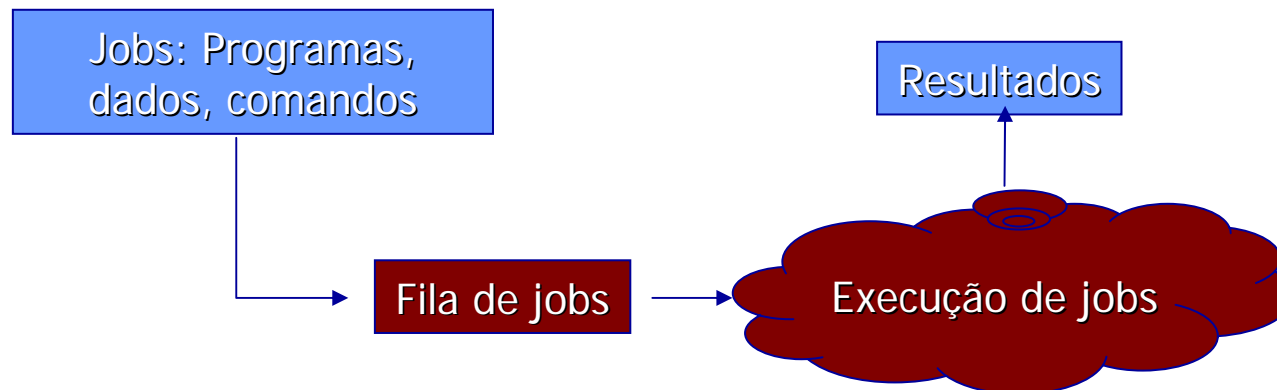
- **Comutadores executavam um job (programas) por vez. Para executar um job:**
  - Programador primeiro escrevia em papel (Fortran ou assembly)
  - Programa é transformado em cartões perfurados
  - Programador levava os cartões para um operador
    - que colocava em uma unidade de leitura
  - **Após o job**
    - operador levava a saída ao programador



# Sistemas batch (em lote)

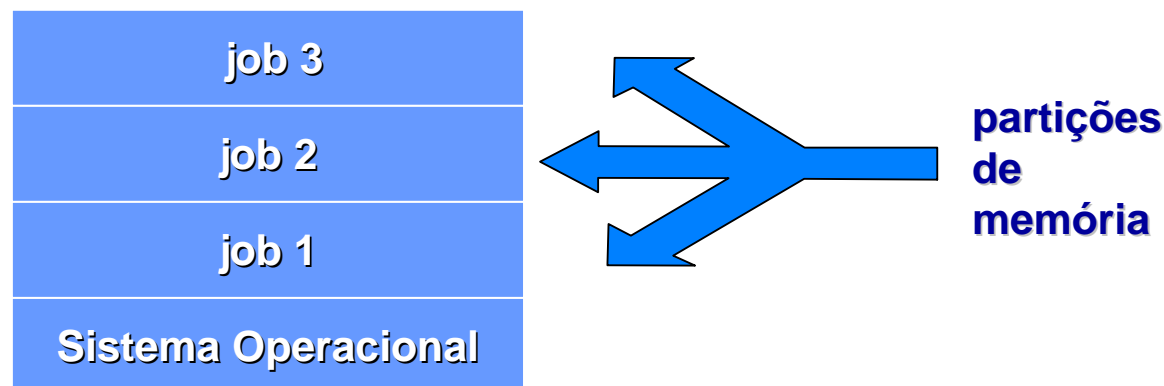
- **Sistema Operacional**

- simplifica a instalação de programas e torna suave a transição de um job para outro
- não há interação com o usuário
- jobs eram armazenadas em fita aguardavam para serem executados em uma fila de jobs
  - fila FIFO (First In, First-out)
  - maioria tinham esquemas de prioridade



# Multiprogramação

- **Solução para o problema de tempo de resposta enquanto jobs esperavam por E/S**
  - particionar a memória
  - instalar um job em cada partição
  - enquanto um job esperava por uma operação de E/S (demorada), outro job poderia usar a CPU
  - se fosse possível manter na memória todos os jobs de uma só vez, a CPU poderia ter uma taxa de utilização de 100%
  - **múltiplos jobs necessitam de proteção de memória**
    - 3a geração implementava em hardware



# Multiprogramação

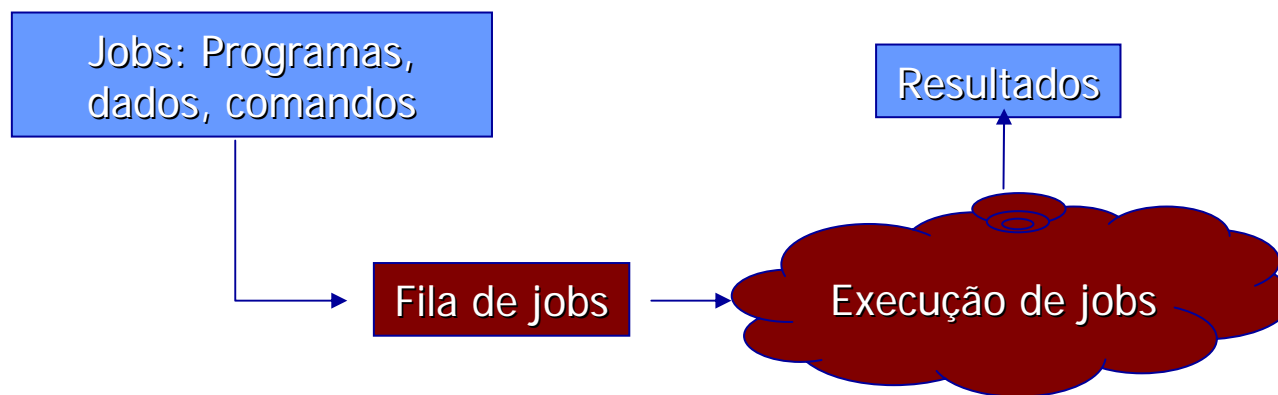
- Apesar de bem aceitos para aplicações científicas e comerciais, constituíam-se em sistemas batch
  - tempo de resposta comprometido
  - ex: procedimento de compilação
    - procedimento de submeter um programa para compilação, receber o resultado, corrigir algum erro, resubmetê-lo novamente poderia consumir várias horas
- Necessidade de melhores tempos de resposta conduziram a sistemas time-sharing
  - variação de multiprogramação onde cada usuário tem um terminal on-line



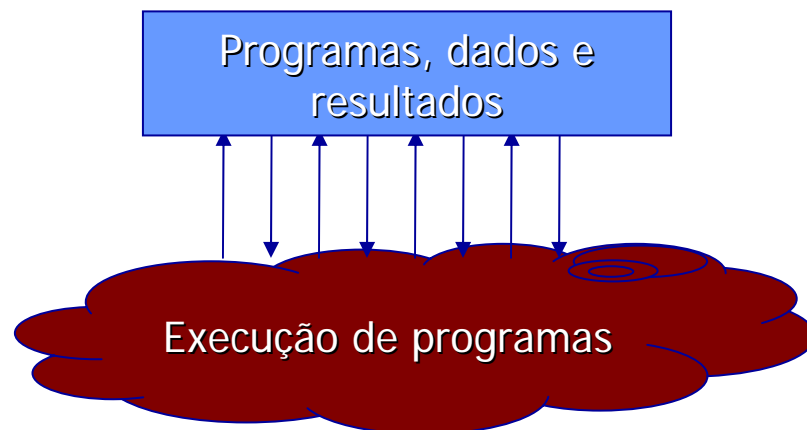
# Processamento em batch e interativo

- **Processamento em batch**

- tem falta de interatividade com os usuários



- **Processo interativo: time-sharing**



# Quarta Geração (1980-hoje)

- **Característica : PC's e estações de trabalho**
  - desenvolvimento de chips em LSI (Large Scale Integration) permitiu o desenvolvimento de computadores menores
  - conectados via redes de computadores
- **2 sistemas operacionais são dominantes**
  - **MS-DOS - Microsoft**
    - arquitetura Intel (IBM PC)
    - sucessor é o Windows'9x
      - windows 3.x eram apenas um shell do sistema MS-DOS
  - **UNIX**
    - arquitetura motorola 68000, arquitetura Intel e RISC's
    - variantes: LINUX, Solaris, AIX

# Sistemas Multiprocessados

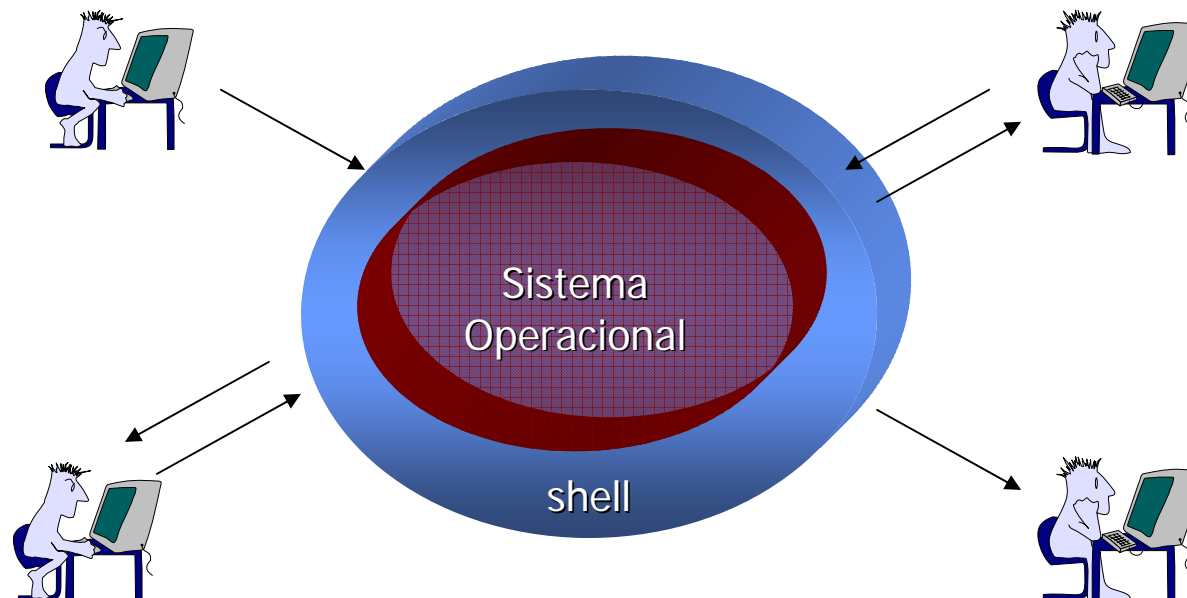
- **Computadores com vários processadores**
  - **sistema operacional não apenas coordenará a competição entre as várias tarefas que são executadas de fato simultaneamente**
    - também controlará a alocação de tarefas aos diversos processados
    - envolve problemas de balanceamento de carga
      - garantir que os processadores sejam utilizados de modo eficiente
    - envolve também escalação
      - divisão das tarefas em várias sub-tarefas de número compatível com o número de processadores disponíveis



# **ARQUITETURA DOS SISTEMAS OPERACIONAIS**

# Arquitetura de um Sistema Operacional

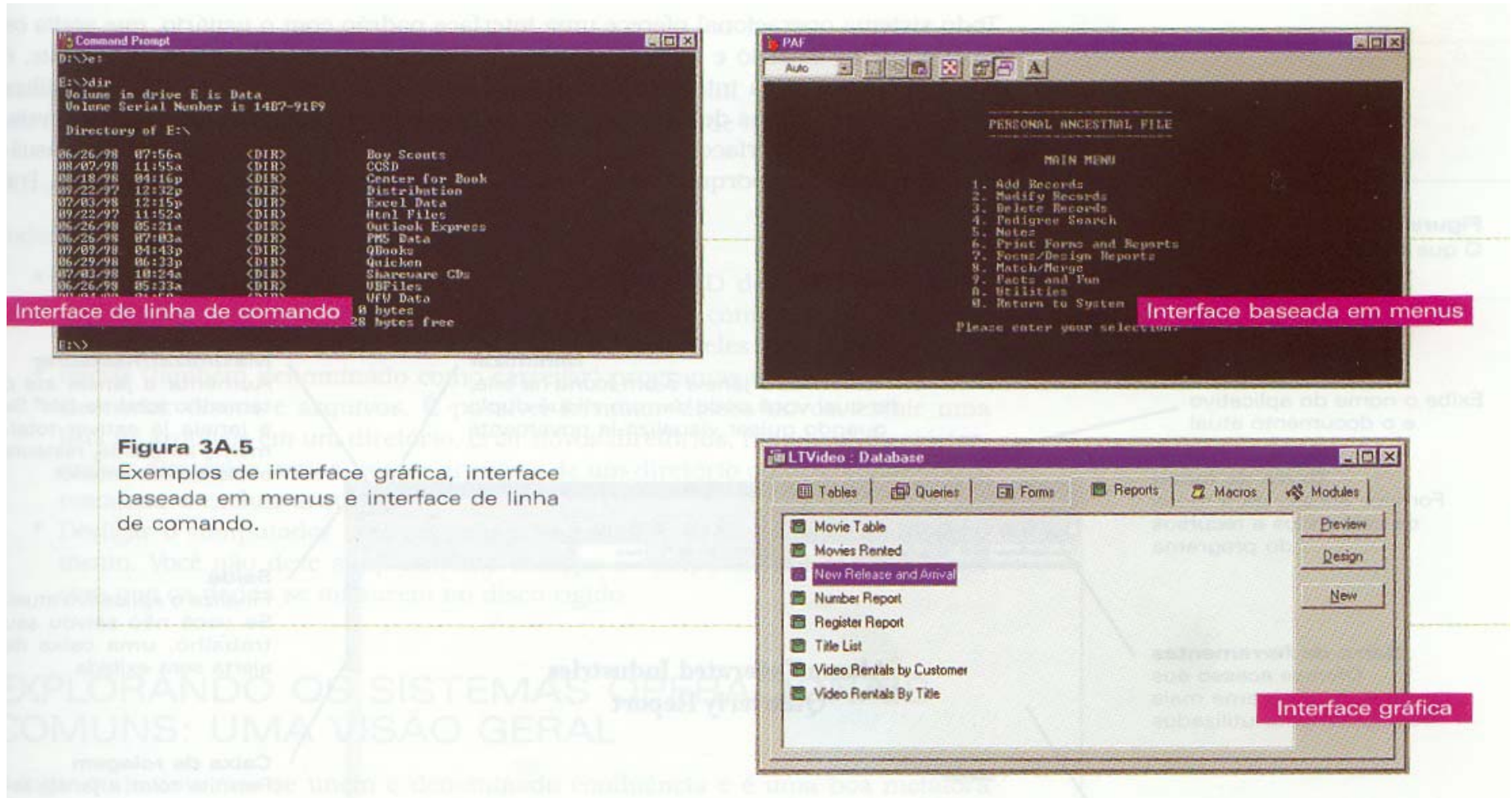
- **Basicamente dividido em**
  - **shell**
    - é a interface entre o usuário e o sistema operacional
    - é um interpretador de comandos
      - possui embutido uma linguagem de programação
  - **núcleo (kernel)**
    - funções básicas necessárias à utilização do computador



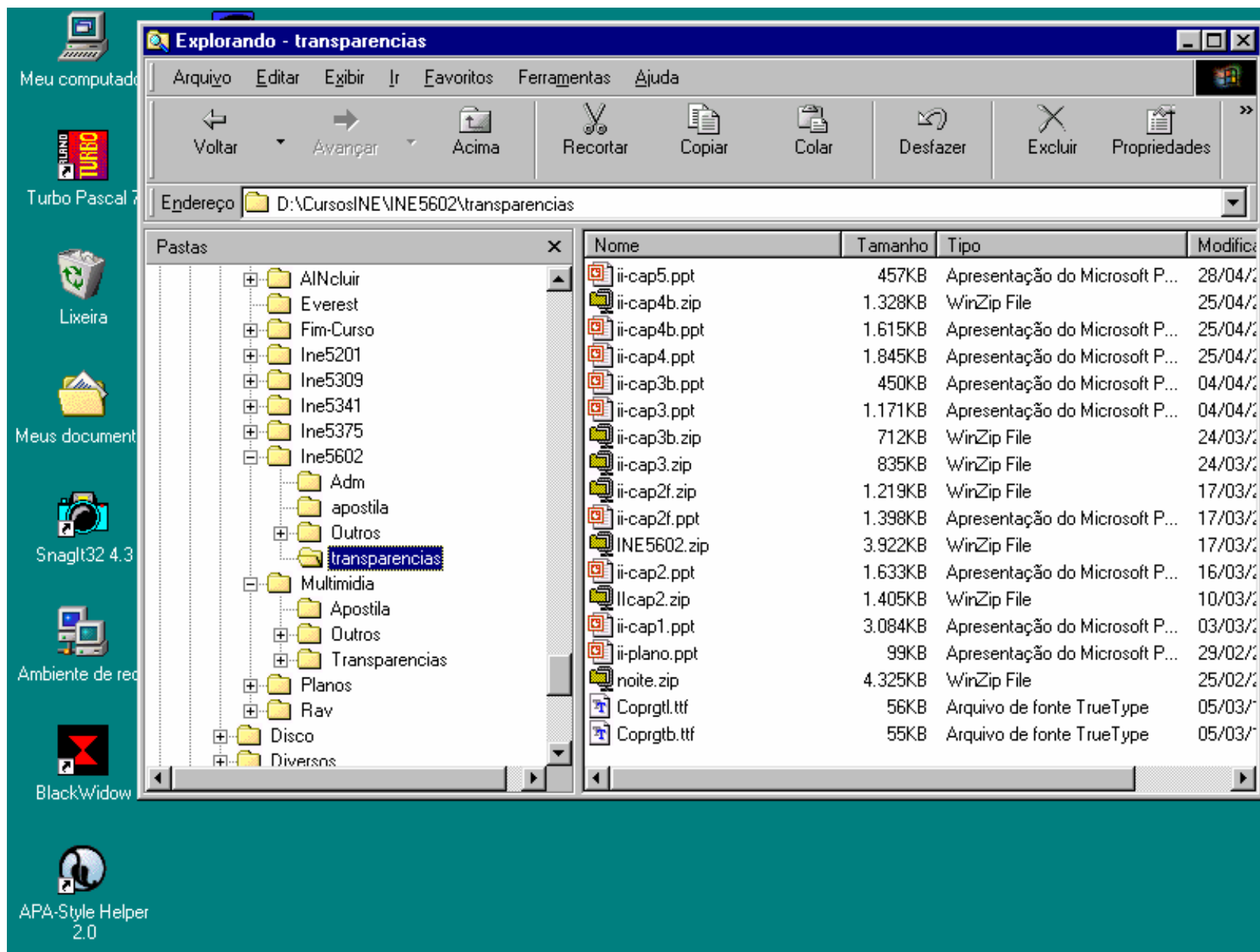
# Shell

- **Define uma interface entre o sistema operacional e seus usuários**
  - fornece uma comunicação natural com os usuários do computador
  - *shells* antigos se comunicavam com os usuários, via teclado e monitor, por mensagens textuais
  - *shells* modernos fornecem uma interface gráfica com o usuário
    - GUI – Graphical User Interface
    - na qual objetos a serem manipulados (arquivos e programas) são representados graficamente através de ícones na tela
    - permite ao usuário executar um comando do sistema apontando e deslocando ícones

# Shell



# Shell: Gráfico





# Shell

- **Distinção entre *shell* e as partes internas do S.O.**
  - enfatizada pelo fato de alguns S.O. permitirem ao usuário selecionar, entre diversos shells, aquele que lhe for mais adequado
    - UNIX: Borne Shell, C shell e Korn shell
    - Versões mais antigas do Windows
      - eram shells de substituição ao MS-DOS
  - **Sistema Operacional permanece o mesmo**
    - exceto quanto à forma em que se comunica com o usuário
- ***Shells* com características similares podem ser utilizados em conjunto com sistemas operacionais de diferentes estruturas internas**
  - propicia uniformidade na interface homem-computador em uma variedade de computadores
    - exemplo: Computadores IBM da série System/360

# Núcleo (kernel)

- **Conteúdo**

- **Componentes de software que executam as funções básicas necessárias ao funcionamento de cada instalação computacional em particular**
  - gerenciador de arquivo
  - acionadores de dispositivo (device drivers)
  - gerenciador de memória
  - Gerenciador de processos:
    - escalonador (scheduler)
    - despachante (dispatcher)

# Núcleo (kernel)

- **Gerenciador de Arquivo**

- **Parte mais visível do sistema operacional**
  - Aplicações necessitam armazenar e recuperar informações, organizadas como **arquivos**
- **Função é coordenar o uso dos recursos de armazenamento de massa do computador (memória secundária)**
  - mantém informações sobre a localização de cada arquivo
  - sobre os usuários autorizado a acessar os diversos arquivos
  - sobre as áreas disponíveis no armazenamento de massa
    - para novos arquivos ou expansão de arquivos já existentes

# Núcleo (kernel)

- **Gerenciador de Arquivo**

- **Permite que os arquivos sejam agrupados em conjunto chamados pastas ou diretórios (folders)**
  - permite ao usuário organizar seus arquivos de acordo com as respectivas finalidades
  - permite criar uma organização hierárquica
    - possibilitando que cada diretório possa conter, por sua vez, sub-diretórios
  - sequência de aninhamentos de níveis de diretório é denominado trajetória (path)
    - /home/willrich/public\_html/
    - D:\CursosINE\INE5602\

# Núcleo (kernel)

- **Gerenciador de Arquivo**

- **Qualquer acesso a arquivo por parte de um módulo de software**

- inicialmente é solicitado ao gerenciador de arquivos para fazer acesso ao arquivo
      - procedimento chamado “abrir arquivo”
    - se o gerenciador aceitar o pedido
      - ele fornecerá a informação necessária para encontrar e manipular o arquivo
        - » informação é mantida em uma área da memória principal (descritor de arquivo)
    - com base na informação contida neste descritor de arquivo que operações elementares individuais são executadas sobre o arquivo

# Núcleo (kernel)

- **Gerenciador de Arquivo (Sistema de Arquivos)**

- Ao nível do *shell* o Sistema de Arquivos oferece chamadas de sistema, com interface única para dispositivos
- Operações mais comuns:
  - Criar; excluir (create; delete)
  - Ler; gravar (read; write)
  - Abrir; fechar (open; close)
  - Alterar nome (rename)
  - Ler e escrever atributos (get/set attributes)

# Núcleo (kernel)

- **Acionadores de dispositivos (*device drivers*)**
  - **são os módulos de software que executam a comunicação com os controladores**
    - ou as vezes diretamente com os dispositivos
    - solicitando a realização de tarefas aos periféricos
    - um *driver* é projetado para um dado tipo de controlador ou dispositivo
      - impressora, unidade de disco, unidade de fita, monitor, ..
    - Oferece uma interface padrão independente do modelo do dispositivo
  - **drivers convertem solicitações de alto nível em comandos mais elementares**
    - diretamente reconhecíveis pelos controladores ou dispositivos associados àquele *driver*
    - detalhes técnicos associados aos dispositivos são confinados aos *drivers*
      - ficando transparentes aos demais módulos de software

# Núcleo (kernel)

- **Acionadores de dispositivos (*device drivers*)**
  - **Exemplo: gravação em disco**
    - *driver* para uma unidade de disco é capaz de converter um pedido de gravação de um trecho de arquivo em disco para uma sequência de passos
      - referenciando trilhas e setores, e transferir toda essa informação para o controlador apropriado
      - baseado nas informações extraídas do descritor de arquivo
    - controlador se responsabiliza pelo posicionamento do cabeçote de leitura e gravação e pelo acompanhamento do processo de gravação
  - **Exemplo: impressão**
    - acionador para uma impressora efetua a conversão de um pedido e impressão de um trecho de arquivo para um conjunto de operações básicas
      - envolvendo transferência de caracteres, tipos de letras e controles de impressão
      - variam de impressora para impressora



# Núcleo (kernel)

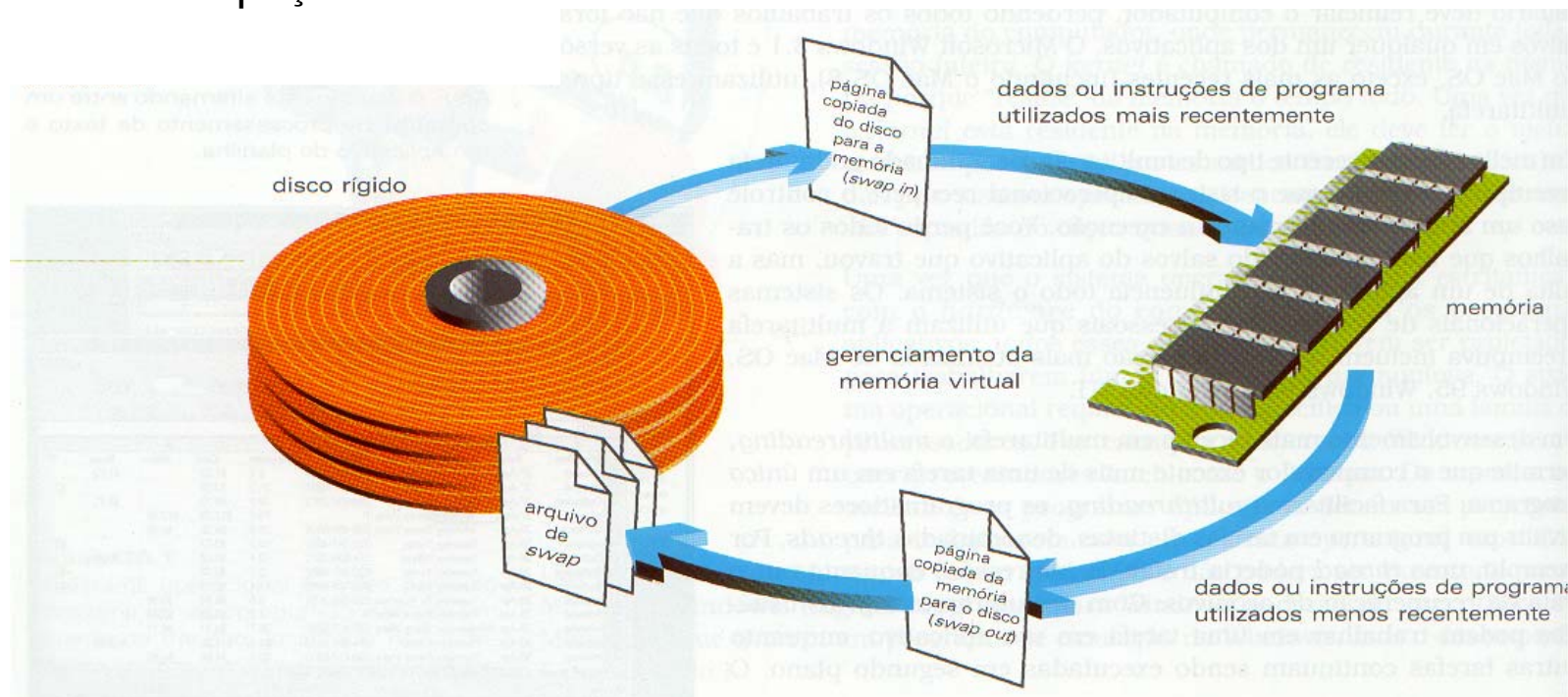
- **Gerenciador de Memória**

- **Parte do SO que gerencia a utilização da memória principal**
- **Em computadores monoprogramados é trivial**
  - programa a executar é colocado na memória principal
  - após a execução ele é substituído pela tarefa seguinte
- **Em ambiente multi-usuários ou multi-tarefas é complexo**
  - máquina se encarrega de executar diversas tarefas ao mesmo tempo
  - muitos programas e blocos de dados coexistem na memória principal
    - cada qual em sua área de memória própria determinada pelo gerenciador de memória
  - na medida das necessidades das diferentes atividades o gerenciador vai providenciando as áreas de memória RAM necessárias
    - mantendo um mapa das regiões de memória não utilizadas

# Núcleo (kernel)

- Gerenciador de Memória

- Sua tarefa se torna mais complexa quando a área total de memória principal solicitada excede o espaço realmente disponível na máquina
  - gerenciador pode criar a ilusão de espaço adicional alterando os programas e dados entre a memória principal e o disco
  - espaço ilusório é chamado de **memória virtual**



# Núcleo (kernel)

- **Gerenciador de Memória**

- **suponha que seja solicitada uma área de 64MB e somente 32MB estejam disponíveis**

- para criar a ilusão de um espaço maior de memória, o gerenciador de memória divide a área solicitada em partes chamadas **páginas** e armazena em disco o conteúdo destas páginas
      - um tamanho típico de página não é superior a 4KB
    - na medida que estas páginas forem sendo solicitadas
      - gerenciador de memória pode armazená-las na memória física em substituição a outras que já não sejam mais necessárias
        - » arquivo de swap
      - de modo que as demais partes do software possam ser executadas

# Núcleo (kernel)

- **Gerenciador de processos: Escalonador (scheduler) e Despachante (dispatcher)**
  - **Necessário em um sistema de tempo compartilhado**
  - **Escalonador**
    - determina quais atividades serão executadas
  - **Despachante**
    - controla a distribuição de fatias de tempo para tais atividades

# Coordenação das atividades da máquina

- **Gerenciador de Processos**
- **Processo**
  - **Um dos conceitos mais fundamentais dos SOs modernos**
    - é uma distinção entre um programa e a atividade de executá-lo
    - programa
      - um conjunto estático de instruções
    - execução do programa
      - uma atividade dinâmica
      - propriedades mudam à medida que o tempo avança
      - atividade conhecida como **processo**

# Coordenação das atividades da máquina

- **Processo**

- **Leva em conta a situação corrente da atividade**
  - conhecida com o **estado do processo**
- **Estado do processo, inclui**
  - posição do programa que está sendo correntemente executada (contador de instruções)
  - valores contidos nos registradores do processador
  - posições associadas de memória

# Coordenação das atividades da máquina

- **Um único programa pode ser associado a mais de um processo**
  - **em sistemas multi-usuários de tempo compartilhado**
    - dois usuários podem editar textos ao mesmo tempo usando um mesmo programa
    - cada qual caracteriza um processo separado
      - com seus próprios conjunto de dados e sua própria taxa de progresso
  - **SO pode manter na memória principal uma só cópia do programa editor e permitir que cada processo o utilize à sua maneira (durante a fatia de tempo que lhe couber)**

# Coordenação das atividades da máquina

- **Os processos concorrem por fatias de tempo de execução**
  - estes processos pode ser associados a aplicativos, utilitários e porções do próprio SO
  - SO deve coordenar todos estes processos
  - **atividade de coordenação inclui**
    - garantir que cada processo tenha acesso aos recursos de que necessita
      - dispositivos periféricos, área na memória principal, acesso a dados e acesso ao processador
    - que processos independentes não interfiram uns com os outros
    - que processos que se intercomunicam tenham a possibilidade de trocar informações entre si (comunicação interprocessos)



# Coordenação das atividades da máquina

- **Gerenciador de Processos**

- Escalonador (scheduler) e Despachante (dispatcher)
  - realizam as atividades de coordenação de processos
  - partes integrantes do núcleo do SO

- **Escalonador**

- mantém um registro dos processos presentes no sistema computacional
- inclui novos processos neste conjunto
- remove processos que já completaram sua missão

# Coordenação das atividades da máquina

- **Escalonador mantém uma tabela de processos**
  - região na memória principal utilizada para coordenar os processos
  - mantém vários indicadores para cada processo:
    - área de memória designada para o processo (obtida pelo gerenciador de memória)
    - prioridade do processo
    - indicador se o processo está pronto para ser executado ou à espera de um evento
      - pronto: estado em que o processo está pronto para prosseguir
      - estado de espera: aguardando um evento (acesso a disco ou recebimento de uma mensagem)

# Coordenação das atividades da máquina

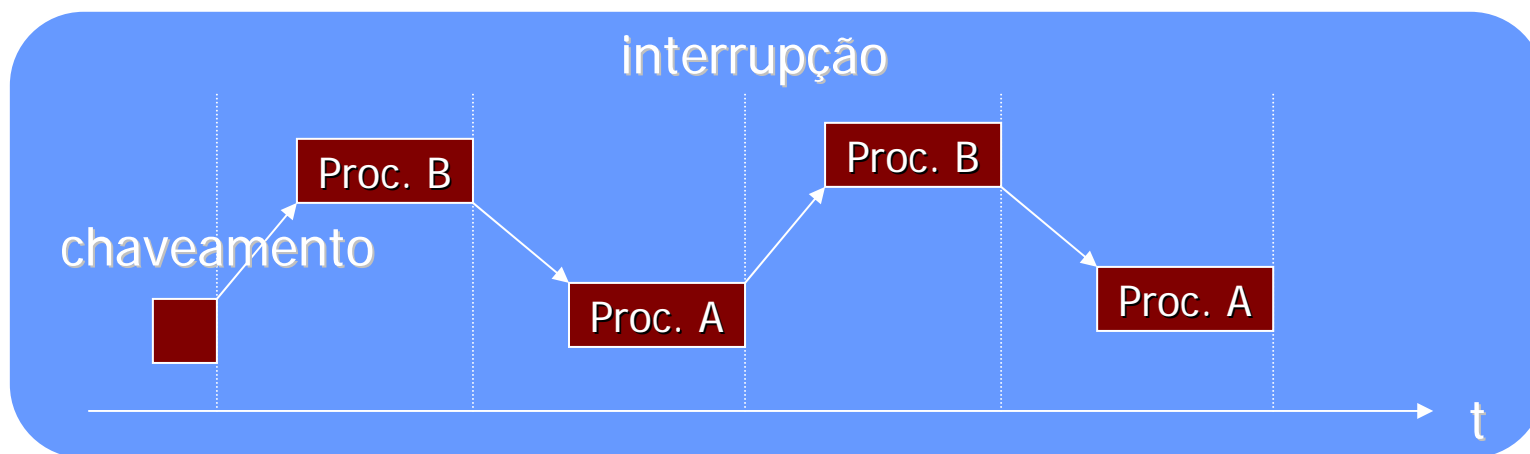
- **Escalonador**

- mantém uma tabela de processos atualizada à medida que o processo vai progredindo
- Na medida que os processos evoluem
  - irá alternar de estado pronto e estado de espera
  - terá sua prioridade variada ao longo da execução
  - irá ser removido da tabela no final da sua execução

# Coordenação das atividades da máquina

- **Despachante**

- Módulo do núcleo do SO cuja função é a de assegurar que os processos escalonados sejam de fato executados
- Atividade realizada dividindo-se o tempo físico em pequenas fatias (cerca de 50ms)
  - denominada *quantum* (time slice)
- Atenção do processador é revezada entre os processos
  - a cada qual é concedido um intervalo de tempo não superior a duração do quantum
  - procedimento de alternar o processador de um processo para outro é chamado de **chaveamento de processos**



# Interrupção, exceção e trap:

- **são eventos de hardware ou software que movimentam e dirigem os sistemas operacionais**
  - pois um sistema operacional só recebe o controle da execução quando ocorre alguma interrupção, exceção ou trap
- **Interrupção e exceção**
  - São associados a sinais de hardware que fazem com que o processador interrompa a execução do programa que vinha executando para atender uma outra tarefa
    - guarda informações para poder continuar, mais tarde, a execução desse programa
    - e passa a executar uma rotina específica que trata da interrupção
    - A diferença entre exceções e interrupções depende do hardware do processador

# Interrupção, exceção e trap

- **Trap**

- é uma instrução especial que, quando executada pelo processador, origina as mesmas ações ocasionadas por uma interrupção ou exceção (chamada de serviços do SO por software)
  - Há o salvamento de informações para poder continuar, mais tarde, a execução do programa e desvio para uma rotina específica que trata do *trap*
- um trap é uma interrupção ou exceção ocasionada por software
- é uma interrupção prevista
  - programada no sistema pelo próprio programador

# Interrupção, exceção e trap

- **Origem de uma interrupção**

- podem ser originadas pelos vários dispositivos periféricos (terminais, discos, impressoras, etc.), pelo operador (através das teclas do console de operação) ou pelo relógio do sistema

- **As exceções estão relacionadas a:**

- Falhas durante o processamento (divisão por zero, falta de página na RAM, erros nos barramentos etc.)

- **relógio (timer)**

- **é um dispositivo de hardware que**
  - decrementa automaticamente o conteúdo de um registrador ou posição de memória, com uma frequência constante
  - e interrompe a CPU quando o valor decrementado atinge zero
- **SO garante que ocorrerá pelo menos uma interrupção dentro de um intervalo de tempo  $t$ , colocando no relógio um valor que demore  $t$  unidades de tempo para ser decrementado até zero**
  - esta atribuição de valor ao relógio é feita imediatamente antes do sistema operacional entregar a CPU para um programa de usuário.

# Interrupção, exceção e trap

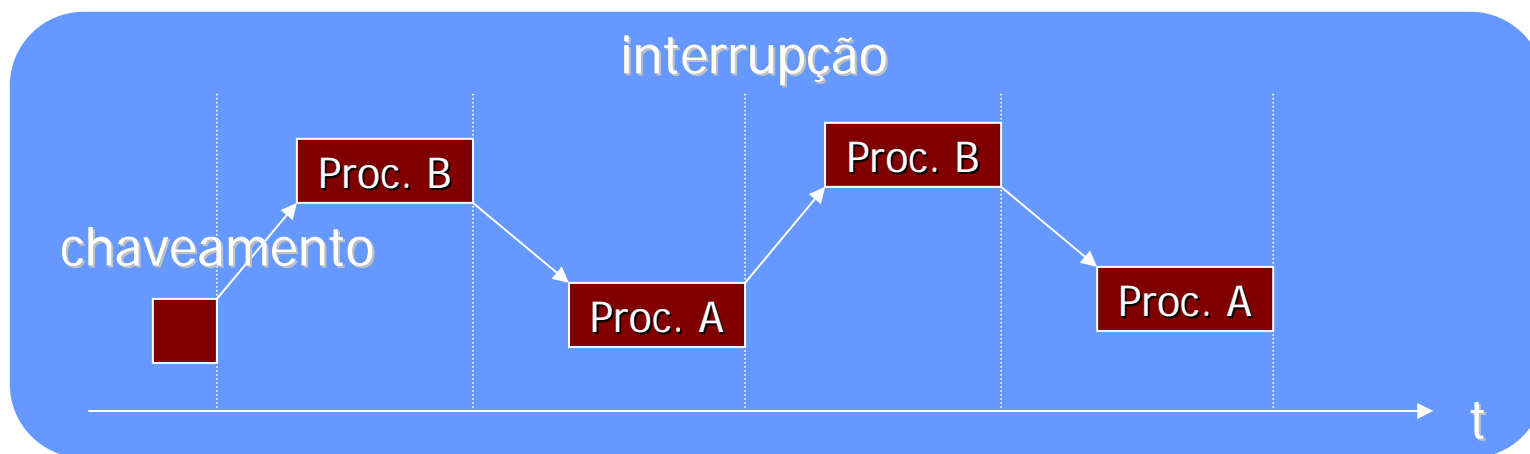
- **Uma interrupção não afeta a instrução que está sendo executada pela CPU no momento em que ela ocorre**
  - CPU detecta interrupções apenas após o término da execução de uma instrução (e antes do início da execução da instrução seguinte)
- **Habilitação e desabilitação de interrupções**
  - computadores possuem instruções para mascarar (desabilitar, inibir) o sistema de interrupções
  - enquanto as interrupções estão mascaradas elas podem ocorrer, mas não são sentidas pelo processador
    - as interrupções ficam pendentes (enfileiradas) e só serão sentidas quando uma instrução que desmascara as mesmas é executada



# Coordenação das atividades da máquina

- **Despachante**

- **Cada vez que um processo inicia o uso de sua fatia de tempo**
  - despachante dispara um circuito temporizador encarregado de medir o quantum
- **Ao final do quantum o temporizador gera uma interrupção ou exceção**
  - processador interrompe o que está realizando
    - bloqueio do processo em andamento
  - registra o ponto da tarefa no qual se foi interrompido
  - passa a tratar a interrupção (rotina de tratamento de interrupção)
    - rotina faz parte do despachante

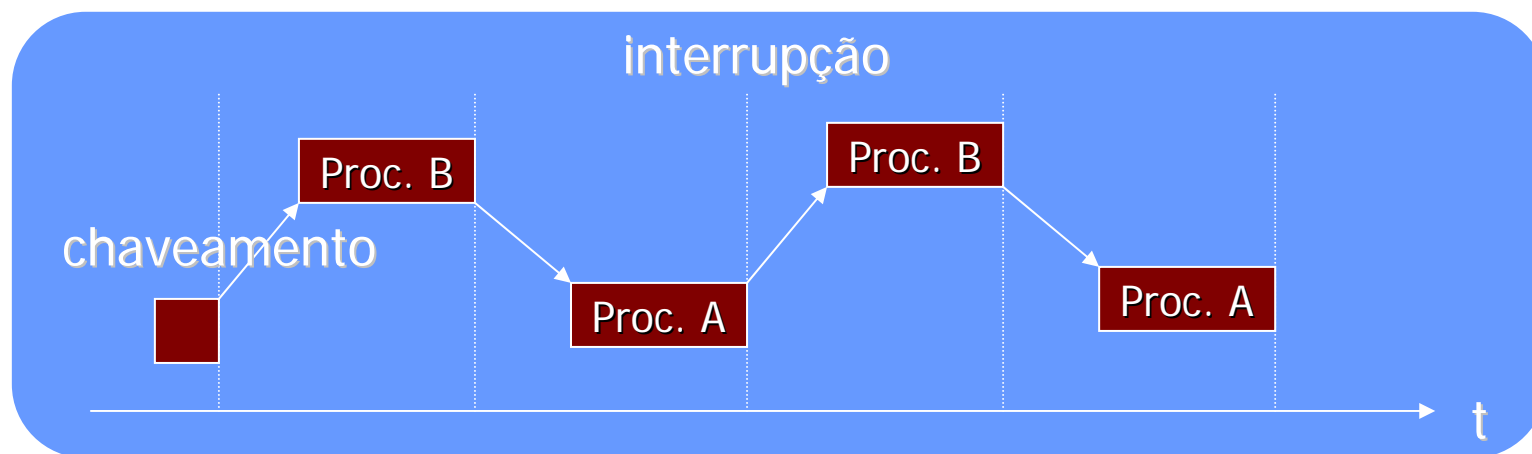


# Coordenação das atividades da máquina

- **Despachante**

- **No tratamento da interrupção/exceção**

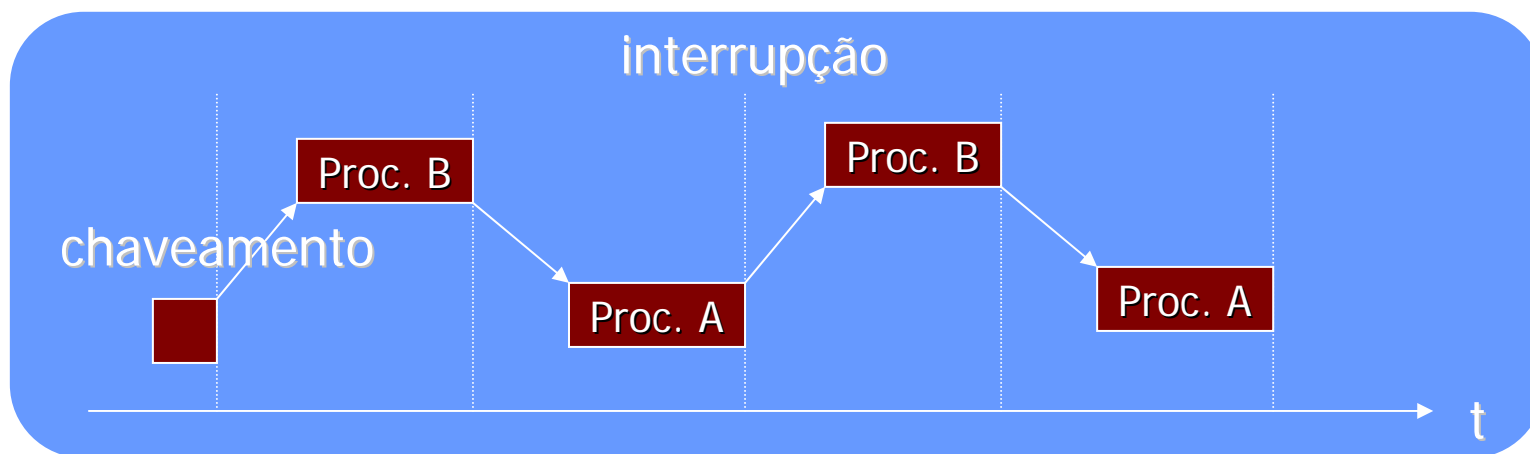
- despachante permite que o escalonador atualize a tabela de processos
      - prioridade do processo interrompido poderá ser reduzida e prioridades dos demais aumentadas
    - despachante seleciona o processo de maior prioridade dentre os que se encontram prontos
    - reinicia a operação do temporizador



# Coordenação das atividades da máquina

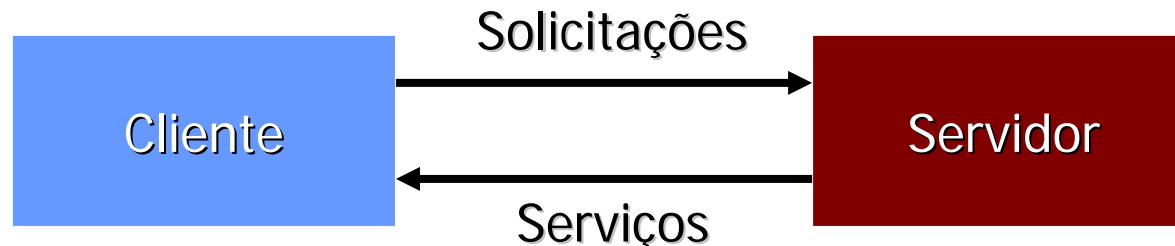
- **Despachante**

- **No chaveamento de processos**
  - Atualizar o estado do processo
    - de maneira que ele possa ser retomado posteriormente
- **Processadores projetados para operarem em sistemas de tempo compartilhado incluem recursos para guardarem tal estado a cada ocorrência de interrupção**
  - possuem instruções para recuperar os estados anteriores
  - simplificam a implementação do despachante
  - projeto do sistema influencia no projeto do processador



# Modelo Cliente/Servidor

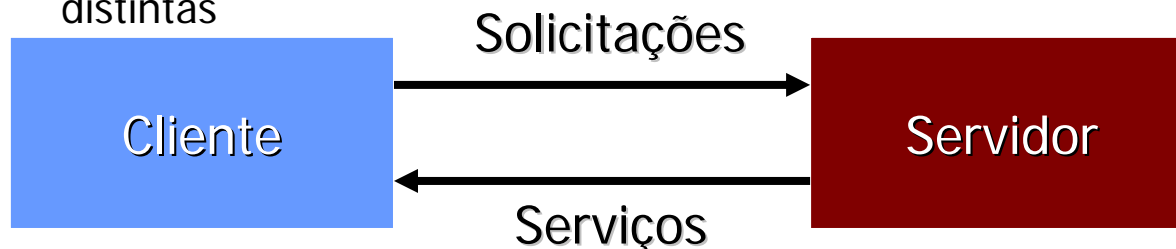
- **Diversas unidades internas a um SO funcionam como processos independentes**
  - que também competem por uma fatia de tempo partilhado
  - se comunicam para coordenar suas atividades
    - para escalar um novo processo, o escalonador solicita espaço de memória ao gerenciador de memória
    - para acessar um arquivo em disco, o processo deve obter informação do gerenciador de arquivo
- **Modelo Cliente/Servidor**
  - simplifica a comunicação entre os processos



# Modelo Cliente/Servidor

- **Modelo Cliente/Servidor**

- **Cliente envia solicitações a outras unidades**
  - Processo solicita leitura de arquivo
- **Servidor satisfaz as solicitações recebidas dos clientes**
  - Gerenciador de arquivo pode funcionar como um servidor
    - fornecendo acesso a arquivos
- **Padronizam os tipos de comunicação existentes em um sistema**
  - amplamente adotado em sistemas operacionais
  - se componentes de um SO forem projetados como clientes e servidores, a forma de comunicação entre eles é idêntica
    - quer seja entre componentes dentro de uma máquina ou em máquinas distintas



# Tratamento da Competição entre Processos

- **Distribuição dos recursos da máquina entre os processos**
  - **uma tarefa comum dos componentes do núcleo de um SO**
    - gerenciador de arquivo
      - autoriza o acesso a arquivos existentes e a regiões vazias de disco para a construção de novos arquivos
    - gerenciador de memória
      - aloca espaço de memória
    - escalonador
      - aloca espaço na tabela de processos
    - despachante
      - aloca fatias de tempo
  - **é uma tarefa complexa**
    - envolve vários problemas que devem ser tratados para não ocorrer falha no sistema
  - **para construir um sistema confiável**
    - é necessário levar em conta cada possível detalhe do sistema independente de quão insignificante possa parecer

# Tratamento da Competição entre Processos

- **Semáforo**

- **Implementação adequada do sinalizador**

- mesma idéia do semáforo ferroviário
    - mecanismo de sincronização entre processos

- **Deadlock**

- **paralisação completa**

- situação na qual dois ou mais processos ficam impedidos de prosseguir suas execuções devido ao fato de cada um estar aguardando acesso a recursos já alocados a outro