

CURSO SUPERIOR EM TECNOLOGIA EM REDES DE  
COMPUTADORES

# SISTEMAS OPERACIONAIS

Aula 02

PROFESSOR ANTÔNIO ROGÉRIO MACHADO RAMOS

# ARQUITETURA DE COMPUTADORES

- Os computadores possuem uma arquitetura derivada de Von Neumann e Alan Turing
- Todo o hardware (entrada, processamento, memória e saída) são de Von Neumann
- Todo o formalismo para construção de linguagens, protocolos e fluxo de processos são de Turing.

# CAIXINHA DE MÚSICA

- Tambor com os pinos - programa armazenado.
- Palhetas - unidade de controle e atuadores.
- Para mudar a música, basta mudar o tambor.

# SISTEMA OPERACIONAL

- Controle: controla dispositivos e processos; previne o uso incorreto destes recursos.
- Usável: cria um ambiente para as aplicações usarem os recursos de hardware, tais como alocação e controle dos recursos.
- Gerência: Processos, memória, arquivos e dispositivos.

# PRIMEIROS SISTEMAS

- Configuração direta no hardware; não existia software.
- Se um erro ocorria, o hardware era inspecionado para verificar a origem.
- Os registradores tinham lâmpadas para o acompanhamento visual da memória pelo operador.

# MATERIAL INTERESSANTE

- Introdução aos Sistemas Operacionais -  
Prof. Simão Toscani
- Revisão da Arquitetura de Computadores -  
Prof. Simão Toscani
- Serviços dos Sistemas Operacionais -  
Prof. Simão Toscani

# RECURSOS GERENCIADOS PELO SISTEMA OPERACIONAL (S.O.)

- Processo
- Memória
- Arquivos
- Dispositivos

# O QUE FAZ O S.O.

- Controla os recursos utilizados pela aplicação do usuário.
- Controla a aplicação do usuário.
- Previne o uso incorreto que pode acarretar erros no computador.
- Cria um sistema computacional estável.
- Atende as requisições das aplicações por recursos.



# UMA DEFINIÇÃO PARA S.O.

- Aproxima os recursos da máquina.
- Torna o uso do hardware mais estável e eficiente.
- O ambiente operacional (hardware + softwares) fica mais estável.

# HISTÓRICO DO S.O.

1. Primeiros sistemas - somente hardware
2. Surgimento das linguagens de segunda geração (alto nível) e a função do operador.
  - FORTRAN - matemática, física, engenharia.
  - COBOL - administrativo, financeiro.
  - O operador fazia todo o processo para executar as aplicações com estas linguagens.
3. Substituição do operador pelo monitor residente - os programas eram classificados em lotes (batch) para executarem no computador.

# MONITOR RESIDENTE

- Tabelas de Interrupção
- Controladores de dispositivos
- Interpretador de cartões e sequenciador de jobs
- Área para programas dos usuários

# MONITOR RESIDENTE

- Possui um interpretador de comando de acesso via tty (teletipo) para programação das rotinas que irão compilar e executar o lote de programas.
- Através deste interpretador que os recursos de hardware podem ser acessados diretamente pelo operador.

# PARA TIRAR O ÓCIO 1

- Mesmo com o monitor residente, o computador ainda ficava ocioso esperando pelos dispositivos de E/S, muito mais lentos.
- Para reduzir o problema, foram empregados computadores satélites para controlar os dispositivos lentos utilizando a fita magnética como buffer.
- Impressora imprimia o que o computador gravava na fita (spooler de impressão).
- Leitora de cartões armazenava a leitura na fita.
- O computador principal (mainframe) acessava as fitas.

# PARA TIRAR O ÓCIO 2

- Utilizando a tecnologia de interrupção pode-se fazer com que o mainframe atenda dispositivos e aplicação simultaneamente.
- O problema é quando os dispositivos interrompem muito o computador. A aplicação atrasa muito o seu processamento.
- Quando ocorre a interrupção por algum dispositivo, o computador pára o que está executando para executar o driver do dispositivo que interrompeu.

# PARA TIRAR O ÓCIO 3

- O spooling é uma evolução do processo de buffer.
- Com a invenção do HDD, o buffer dos dispositivos ficou muito mais rápido.
- O HDD também servia para guardar o processo que não estava usando a CPU no momento.

# PARA TIRAR O ÓCIO 4

- A multiprogramação é a evolução natural do spooling.
- O processo aplicação e os processos dos dispositivos trocam seus tempos de execução com a CPU através dos mecanismos de interrupção.



# CLASSIFICAÇÃO DO S.O. QUANTO AO TEMPO DE EXECUÇÃO DOS PROCESSOS (JOBS)

- Batch: O processo é criado, executa e termina sem interagir com nada a não ser com os dispositivos de E/E para pegar os dados e retornar o resultado do processamento. Se o processo for interrompido, o processamento é perdido, havendo a necessidade de iniciar novamente.
- Time Sharing: Cada processo de cada um dos usuários possui uma janela de tempo para executar. Após esse tempo, o processo em execução perde a CPU para dar lugar ao próximo processo.
- Real Time: Cada processo em execução configurado para real time deve responder a uma requisição no menor tempo possível - em escala humana a resposta deve ser instantânea. Para que isso seja possível, os processos em real time possuem prioridade de execução maiores que os demais processos (execução preemptiva).

# MELHOR USO DO COMPUTADOR

- Nos sistemas monousuário os recursos do computador não eram compartilhados, portanto não havia problemas de segurança. O ambiente operacional não operava com toda a eficiência e poderia ficar ocioso.
- Nos sistemas multiusuário, existe uma maior eficiência porque os recursos são compartilhados. A má notícia é que surgiram os problemas de segurança.

# SEGURANÇA DO S.O.

- Proteção gerada pelo hardware - qualquer problema de segurança detectado pelo hardware gera uma trap (interrupção para um serviço que corrija o problema).
- O hardware só pode ser acessado diretamente por serviços específicos. Para estes serviços o S.O. concede uma permissão absoluta de acesso (modo monitor).
- Os processos do usuário, e o próprio usuário interagindo com sistema, via terminal, tem permissão restrita para o hardware (modo usuário).
- O usuário e suas aplicações só acessam o hardware através dos serviços do S.O. Se houver uma infecção de acesso direto, uma trap é gerada.

# PROTEGENDO A MEMÓRIA

- Sistema monoprogramado: Existe um registrador limite que separa a memória onde estão os serviços do S.O. daquela que pode ser usada pela aplicação do usuário. No caso de tentativa de acesso indevido, uma trap é gerada.
- Sistema multiprogramado: Existem 2 registradores que definem os limites da área ocupada pelo processo que está executando. Quando um processo assume a CPU, os valores dos registradores são atualizados para este processo, pois cada processo ocupa (e também usa) uma posição diferente da memória. No caso de tentativa de acesso indevido, uma trap é gerada.

# PROTEGENDO A EXECUÇÃO

- O S.O. possui alguns sistemas que impedem um processo de monopolizar a CPU.
- Janela de tempo de alguns milisegundos para cada processo. Independente do que o processo esteja fazendo, ele nunca fica mais do que sua janela de tempo.

# SERVIÇOS DO S.O.

- Vamos combinar:
  - Programa fica do meio persistente (HDD, SSD, etc.).
  - Quando o programa é carregado na memória transiente (RAM) para executar, ele é chamado de processo. Podemos fazer isso várias vezes com o programa e termos dele vários processos executando.
  - Quando o processo é do S.O. ele é chamado de serviço.

# PARA QUE SERVEM OS SERVIÇOS

- Execução dos programas (carregar, executar, controle da execução, término).
- Operações de E/S (controle dos dispositivos).
- Controle do sistema de arquivos.
- Monitora a saúde do hardware e dos processos em execução, detectando e corrigindo (quando possível) os erros.
- Alocação de recursos para os processos requerentes.
- Contabilizações (para analisar, bonificar e penalizar processos).
- Segurança (impede que processos façam o que não podem no sistema.)

# SYSTEM CALL

- Conhecida como chamada de sistema.
- É utilizada pelo processo do usuário (e as vezes pelo próprio usuário) para chamar um serviço do S.O.
- O serviço chamado geralmente ativa algum dispositivo.
- Desta forma o usuário pode acessar o hardware sem precisar ter o conhecimento absoluto sobre ele e sem precisar acessá-lo diretamente.
- O serviço chamado passa para o modo monitor e pode fazer tudo com o hardware (que perigo!!!).
- Quando o serviço termina, o processo passa para o modo usuário, que possui várias restrições com relação ao hardware (modo chinelão).



# ATRAVÉS DE UM SYSTEM CALL PODEMOS...

- Iniciar, cancelar e terminar programas.
- interromper, reiniciar, abortar e terminar um processo.
- Alterar os atributos de um processo.
- Esperar gerar e sinalizar a ocorrência de um evento.
- Criar, remover, ter acesso completo e alterar os atributos de um arquivo.
- Requisitar, liberar, ter acesso e alterar os atributos de um dispositivo.
- Ter acesso e alterar aos atributos do sistema (data, hora e parâmetros de funcionamento).

# USANDO O SYSTEM CALL NAS LINGUAGENS DE ALTO NÍVEL

- Tomemos como exemplo a linguagem C.
- Comandos de E/S tais como (printf, scanf, etc.).
- Esses comandos geram system call para acionar o teclado e o vídeo.
- Na realidade cada um dos comandos de E/S é uma rotina de biblioteca (stdio.h) que faz diretamente um system call.

# USANDO SYSTEM CALL EM LINGUAGEM DE BAIXO NÍVEL

- Tomemos como exemplo uma linguagem assembly inventada na hora.
  1. `mov dl,26h` move para o registrador dl o valor, em hexadecimal, do caractere “&”.
  2. `mov ah,02h` move para o registrador ah o valor, em hexadecimal, do serviço de vídeo para exibir um caractere na tela na posição corrente do cursor.
  3. `int 10h` chama a interrupção correspondente ao dispositivo vídeo.
- Ao chamar o vídeo com o serviço selecionado, o caractere “&” será exibido na posição do cursor na tela. Um comando de alto nível que faz a mesma coisa é bem mais simples, mas internamente ele pode ser um complexo programa de baixo nível.

# INTERPRETADOR DE COMANDOS

- É a interface do monitor residente (primeiro S.O.).
- Existe em todos os sistemas operacionais (embora em alguns, ele esteja desabilitado).
- Windows (prompt do MSDos); Unix (terminal).
- É uma forma do usuário interagir diretamente com o S.O.

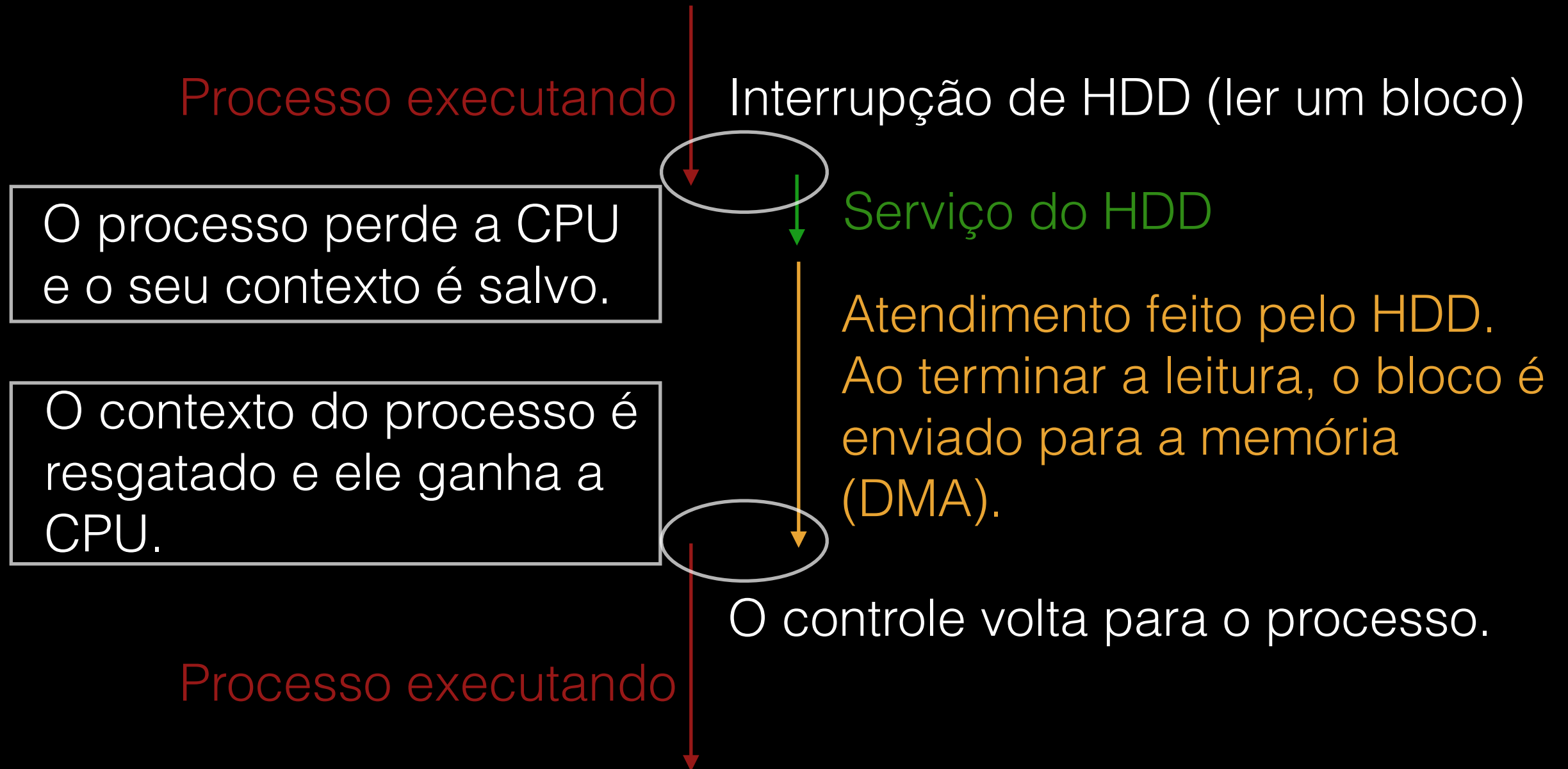
# O S.O. É UM SACO...

- ... de serviços orientados à interrupção.
- ... de interrupções que ativa os serviços dos dispositivos.

# INTERRUPÇÃO

- Podem ser atendidas na hora ou ficarem em um buffer até estarem disponíveis.
- Interrupção do relógio é atendida imediatamente (dispositivo mais rápido).
- Interrupção do HDD pode ficar no buffer até que o mesmo fique disponível (dispositivo mais lento).

# INTERRUPÇÃO



# ESCALONAMENTO

- O dispositivo deve ser escalonado para atender diversas solicitações.
- O S.O. mantém uma tabela de escalonamento e status do dispositivo.
- O serviço que manipula do dispositivo diretamente é chamado de driver.
- Existem drivers genéricos (mouse, teclado, vídeo) que ativam apenas as funções padrão.
- E tem o driver específico para aquele dispositivo, que ativa todas as suas funcionalidades.



# INTERRUPÇÃO POR ERRO

- Divisão por 0, tentativa sem direito de acesso à memória, tentativa sem direito de acesso à dispositivo.
- O hardware detecta e ativa uma interrupção para um serviço de alerta e correção do erro.
- Esse system call é conhecido como trap.

# INTERRUPÇÃO POR HARDWARE E SOFTWARE

- Interrupção de hardware é imprevisível, pois o computador não pode adivinhar, por exemplo, quando uma tecla vai ser pressionada. Todos os dispositivos podem interromper.
- Interrupção por software é previsível porque ela é inserida no programa (system call). Todo system call ativa um dispositivo ou outro serviço.
- Interrupção do timer, para calculo do time slide é a única interrupção de hardware previsível, pois o timer sempre interrompe a intervalos rigorosamente iguais.

# ATIVIDADE

- Pesquisar informações sobre um microprocessador empregado em aparelhos mobile.
  - Fabricante e modelo
  - Velocidade (Hz, Flops)
  - Memória cache
  - Mínimo de 3 modelos de equipamentos onde esse microprocessador é empregado.

Enviar no assunto do email seguido do nome completo dos integrantes da equipe.