

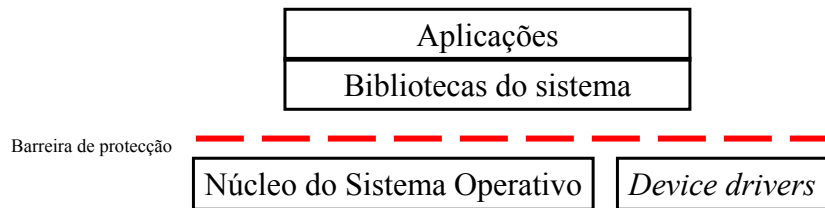
Gestor de Processos

Núcleo do Sistema Operativo

Gestor de Processos

- Entidade do núcleo responsável por suportar a execução dos processos
 - Gestão das Interrupções
 - Multiplexagem do Processador
 - Despacho – efectua a transferência de controlo entre dois processos
 - Escalonamento – optimiza a gestão dos recursos
 - Sincronização no núcleo
 - Implementação das funções sistema relacionadas com os processos e sincronização

Estrutura Monolítica

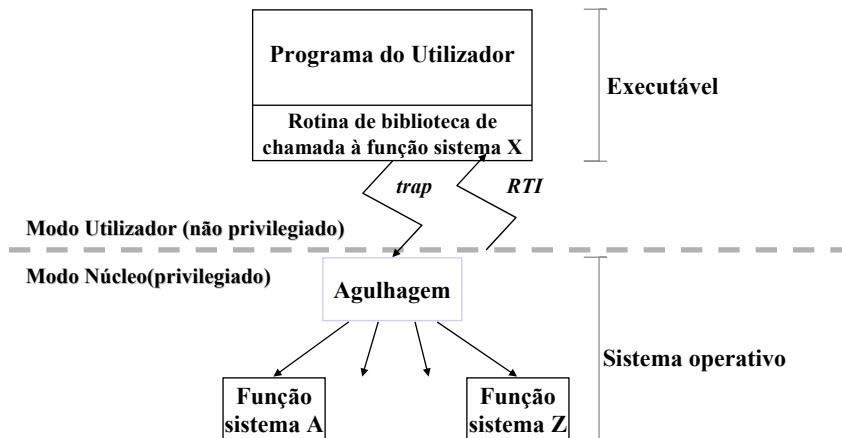


- O núcleo executa-se num modo de protecção diferente dos processos utilizadores
- A mudança automática entre os dois modos é efectuada pelas interrupções (e excepções) e pela instrução de *Return from Interrupt*

8/28/2003

José Alves Marques

Chamada a Funções Sistema



Para relembrar – Intel 80x86

INT – Interrupt Usage:

INT num

Modifies flags: TF IF

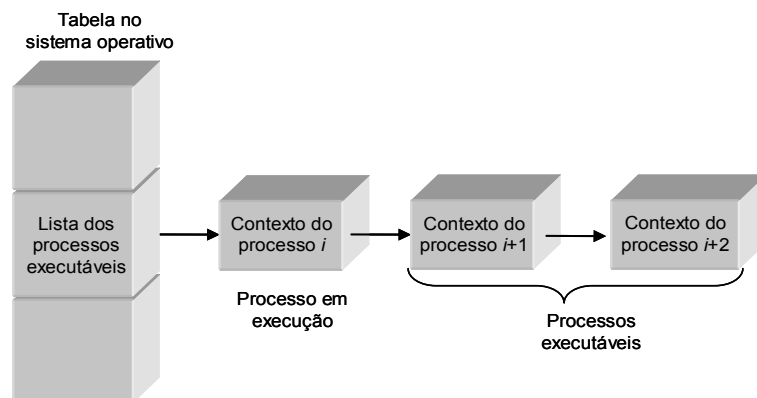
Initiates a software interrupt by pushing the flags, clearing the Trap and Interrupt Flags, pushing CS followed by IP and loading CS:IP with the value found in the interrupt vector table. Execution then begins at the location addressed by the new CS:IP

Representação dos Processos e Tarefas

Contexto: Representação de um processo no núcleo

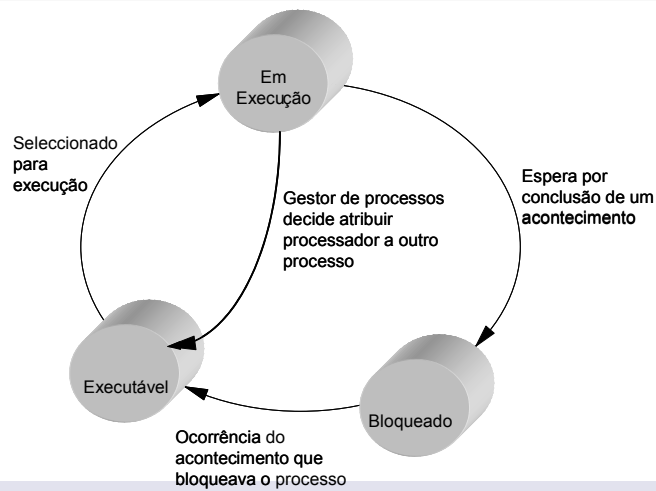
- Contexto de hardware
 - Registos do processador (acumulador, uso geral, contador de programa, stack pointer, flags de estado do CPU)
 - Registos da unidade de gestão de memória
- Contexto de software
 - Identificação (processo, utilizador, grupo)
 - Prioridade
 - Estado do processo
 - Outras informações (periféricos em uso, ficheiros abertos, directório por omissão, programa em execução, contabilização de recursos, signals pendentes, etc.)

Lista dos Processos Executáveis



Comutação de Processos

Diagrama de Estado dos Processos

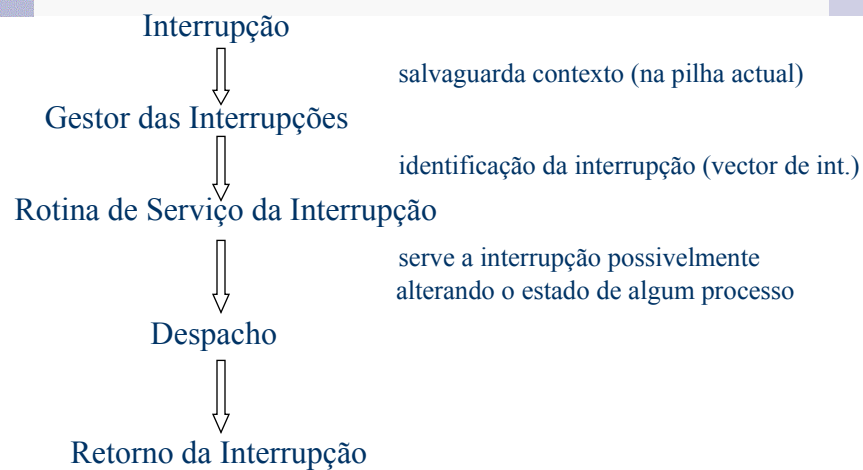


Invocação do Sistema Operativo

- Todas as actividades do sistema operativo podem ser consideradas como desencadeadas por interrupções
- As interrupções podem ser provocadas por:
 - **Hardware**, em particular pelo relógio, mas também pelos diferentes periféricos.
 - **Interrupções de software (traps, software interrupts)** usadas na chamada às funções do sistema pelos programas utilizador.
 - **Excepções** provocadas pelo programa em execução como divisão por zero ou acesso a memória indevido.

Sistema Operativo como o gestor global de todas as interrupções

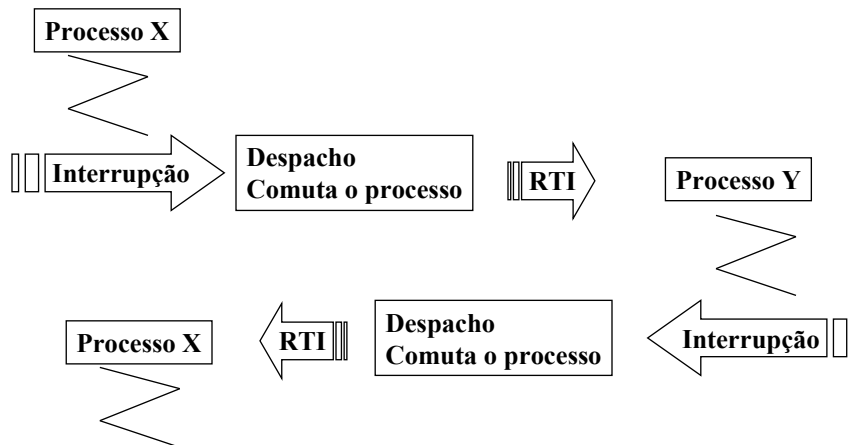
Invocação do Sistema Operativo



Rotina de Serviço da Interrupção

- Copia registos da pilha actual para o contexto do processo (tabela dos processos)
- Corre o código específico à interrupção, possivelmente alterando o estado dos processos
- Invoca o despacho para eventualmente escolher outro processo

Despacho



Despacho

- A função do despacho é comutar o processador sempre que lhe seja indicado para o fazer.
- Funcionalidade
 - copia o contexto *hardware* do processo em execução para o respectivo descritor (entrada na tabela de processos)
 - escolhe o processo mais prioritário entre os executáveis
 - carrega o seu contexto *hardware* no processador
 - transfere o controlo para o novo processo
 - coloca *program counter* guardado no contexto do novo processo na pilha núcleo
 - *return from interrupt* (RTI) é “enganado”

Escalonamento (Scheduling)

Escalonamento – Objectivos

- Optimizar a utilização do processador (e dos restantes componentes do sistema)
- Políticas de escalonamento definem objectivos mais específicos:
- Batch
 - Produtividade - Throughput – maximizar o número máximo de jobs por hora.
 - Turn around time – tempo entre a submissão do trabalho e a obtenção do resultado.
 - Utilização do processador – manter o processador com elevada ocupação
- Tempo partilhado
 - Tempo de resposta – responder rapidamente aos eventos desencadeados pelos utilizadores
- Tempo real
 - Cumprir metas temporais (deadlines) para tratamento dos acontecimentos
 - Funcionamento com desempenho previsível (e.g., multimedia)

Função de Escalonamento

- Define quem deve ser o próximo processo a executar-se, de acordo com a política de escalonamento
- Quão frequentemente deve ser invocada?
- Em teoria, útil sempre que um recurso do sistema (e.g., memória) é atribuído ou libertado.
Problema?

Escalonamento em Tempo Partilhado

- Duas classes de processos:
 - CPU bound – uso intensivo do processador
 - I/O bound – uso intensivo das E/S - interactivos
- Quais devem ter maior prioridade no acesso ao CPU?

Políticas de Escalonamento em Sistemas de Tempo Partilhado

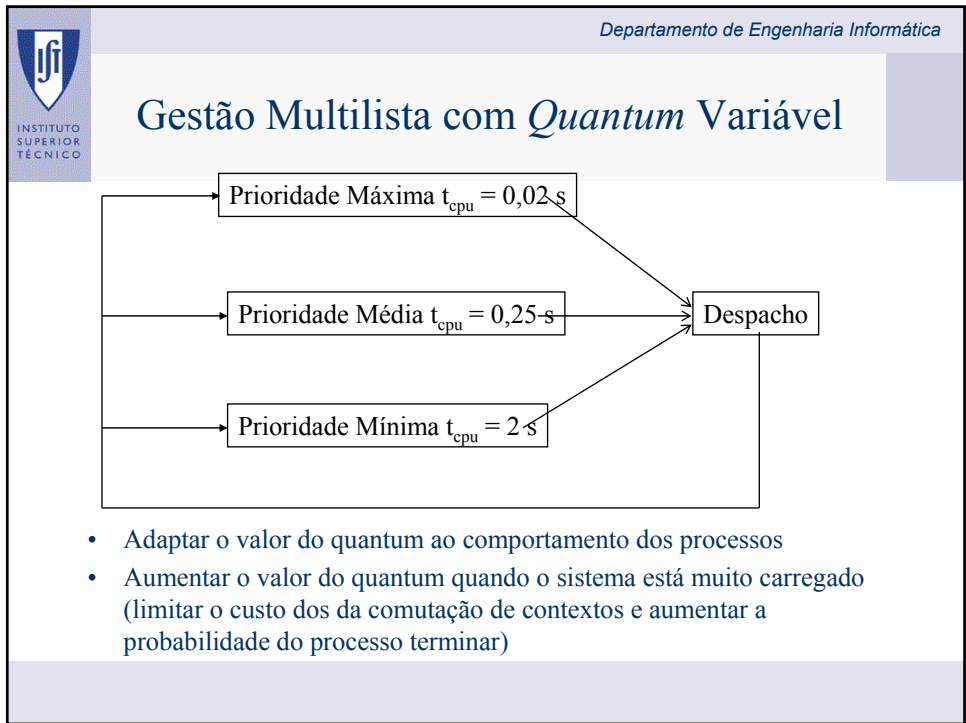
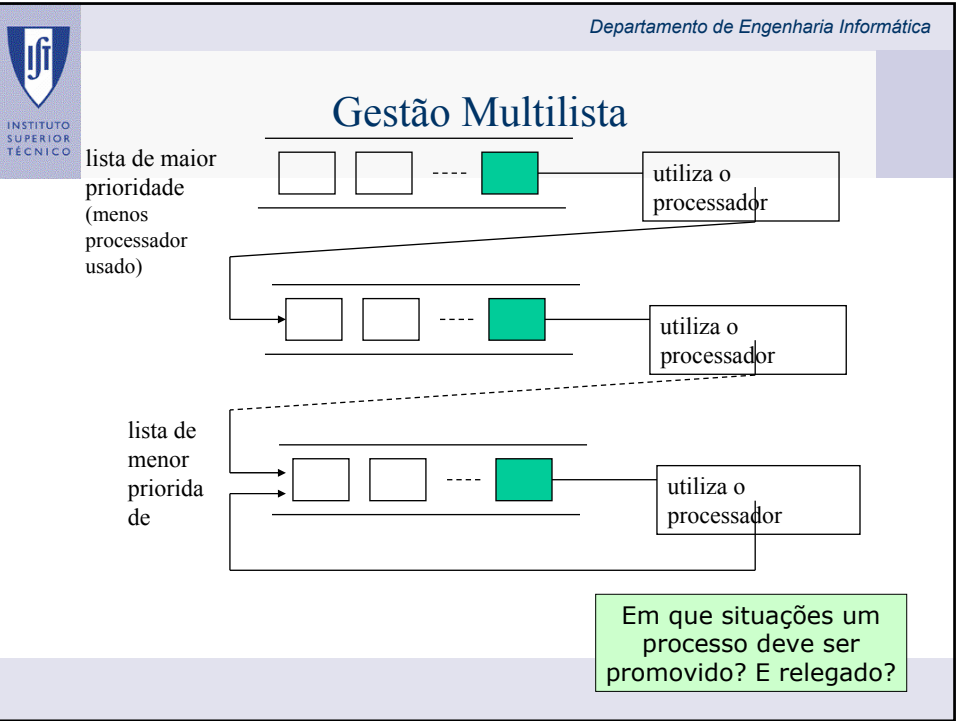
- Tempo de Execução Partilhado (time-slices)
- Prioridades
- Preempção
- Modificação dinâmica das prioridades

Tempo de Execução Partilhado: Time-Slices (ou Round-Robin)

- O que é
 - Tempo de execução de um processo é limitado a um quantum de tempo (time-slice)
- Objectivo:
 - Permitir que todos os processos executáveis tenham oportunidade de dispor do processador ciclicamente
- Implementação
 - Lista de processos executáveis é gerida em round-robin
- Desvantagens
 - Pode conduzir a tempos de resposta elevados em situações de muita carga

Prioridades

- As prioridades permitem definir a importância de um processo no processo de escalonamento
- Um processo mais prioritários tem maior probabilidade de dispor do processador
- A prioridade pode ser
 - **Fixa**, usual em processos de tempo real
 - **Dinâmica** consoante o comportamento do processo – usual nos sistemas de tempo virtual e normalmente privilegiando os processo interactivos (*I/O intensivos*).



Preempção

- **O que é:**
 - acção de retirar o processador a um processo em execução devido à existência de outro mais prioritário
- **Objectivo:**
 - permite que os processos mais prioritários reajam rapidamente a um dado acontecimento (reactividade aos acontecimentos externos).
- **Implementação:**
 - O despacho deve ser chamado na sequência de todas as acções susceptíveis de modificarem os estado dos processos
- **Desvantagens:**
 - Mudança frequente de contexto (ex.: um processo só deve ser retirado de execução depois de ter usado processador durante um tempo mínimo).

Chamadas Sistema

Chamadas Sistema relacionadas com os Processos

- `IdProcesso = CriarProc` (Código, Prioridade)
- `Sair` ([Estado])
- `EliminarProc` (`IdProcesso`)
- `Estado = EsperarTerminacao` (`Idprocesso`)
- `IdTarefa = CriarTarefa` (`procedimento`);
- `EliminarTarefa` (`IdTarefa`)
- `EsperaTarefa` (`IdTarefa`)
- `ComutarTarefa`
- `RotinaAssincrona` (`Evento`, `Procedimento`)
- `s = CriarSemaforo` (`num_unidades`)
- `Esperar` (`s`)
- `Assinalar` (`s`)
- `Suspender` (`IdProcesso`)
- `Acordar` (`IdProcesso`)
- `Adormecer` (`Periodo`)

Identificar que acções devem ser executadas por estas chamadas

Semáforos

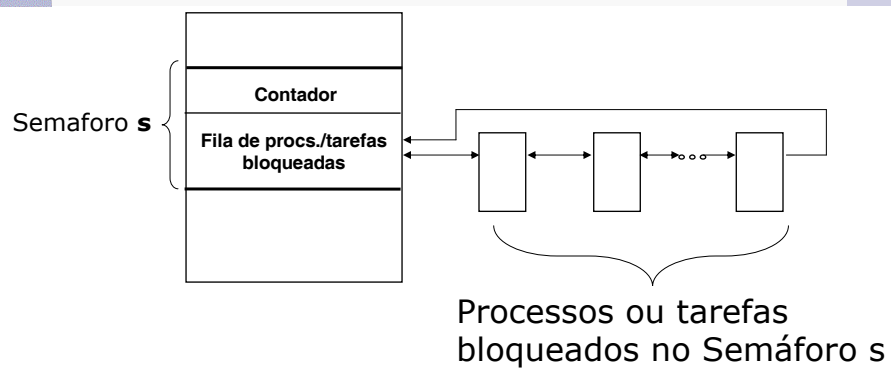


Diagrama de Estado dos Processos

