

Resumo

- Arquitetura de um computador:
 - periféricos de I/O e CPU executam concorrentemente
 - Cada controlador de periférico encarrega-se de um tipo particular de periférico e tem um buffer local.
 - O CPU desloca dados de/para a memória principal para/de buffers locais.
 - I/O acontece do periférico para o buffer local do controlador.
 - O controlador do periférico informa o CPU que terminou a sua operação, causando uma interrupção (*interrupt*).

Resumo

- Para tornar o processamento mais eficiente:
 - interrupções
 - DMA (direct memory access)

Resumo

- Protecção: hardware deve prover mecanismos
 - Dois modos de operação (alguns hw não suportavam: 8088)
 - Protecção de I/O (operações de I/O são realizadas em modo kernel)
 - Protecção de acesso a memória (usa dois registradores base e limite)
 - Protecção do CPU (timer)

Então como usar o sistema?

- Dado que as instruções de I/O são protegidas (são executadas em modo kernel), como é que um programa do utilizador faz I/O?
- Chamadas de sistema – o método usado por um processo para pedir a intervenção do SO:
 - toma habitualmente a forma de uma armadilha (trap) para um endereço no vector de interrupções.
 - o controlo passa para o interrupt-handler e o mode-bit é colocado em modo kernel.
 - o kernel verifica se os parâmetros estão correctos, executa o pedido e devolve o controlo para a instrução seguinte à chamada ao sistema.

