

## Capítulo 5 – Entrada e Saída

- O sistema operacional também controla todos os dispositivos de Entrada e Saída.
- Na medida do possível a interface entre usuário e dispositivos deve ser a mesma para todos os dispositivos

### 5.1 – Princípios do hardware de E/S

#### 5.1.1 – Dispositivos de E/S

- Podem ser de dois tipos: de bloco ou de caracteres.
- Os de bloco armazenam informações em blocos de tamanho fixo cada um com seu endereço. Cada bloco pode ser lido ou escrito independentemente do outro. Ex: CDS, discos rígidos e pen drives.
- Os de caracteres envia ou recebe um fluxo de caracteres sem se deter na noção de bloco. Não é endereçável e não dispõe de estruturas de posicionamento. Ex: Impressoras, mouses e teclados.
- O software deve ser capaz de trabalhar com diferentes taxas de transmissão, já que cada dispositivo tem uma taxa diferenciada.

#### 5.1.2 – Controladores de dispositivos

- Os dispositivos de E/S tem um componente mecânico e um componente eletrônico.
- O componente eletrônico se chama controladora.
- Num disco por exemplo a função da controladora é converter o fluxo de bits em bytes coerentes e executar a verificação de erro.
- O SO passa informações para a controladora e estas são livres para executar a tarefa desejada, retirando o trabalho pesado das mãos do SO.

#### 5.1.3 – E/S mapeada na memória

- Cada controlador de dispositivo tem registradores nos quais o SO pode escrever os comandos para execução de tarefas.
- Existem duas maneiras do SO acessar os dispositivos: através de portas. O segundo método mapeia todos os registradores dos dispositivos no espaço de endereçamento.

- Com E/S mapeada na memória não há a necessidade de funções especiais de leitura e escrita, basta usar aquelas já existentes para a memória.
- Não há necessidade de proteção especial para E/S, pois o SO pode proteger a área de endereçamento de um dispositivo facilmente.
- Toda instrução capaz de referenciar a memória pode referenciar um registrador de E/S.
- Sem E/S mapeada a CPU se ocupa mais.
- E/S mapeada complica um pouco a utilização de cache em dispositivos.
- Precisa-se ter cuidados especiais em sistemas com muitos barramentos.

#### 5.1.4 – Acesso direto à memória

- Um sistema só pode usar DMA se o hardware de controle estiver presente.
- O controlador de DMA pode ser acessado pela CPU através de escritas em registradores.
- Processo de uso da DMA em um disco:
  - 1) CPU programa a DMA
  - 2) CPU solicita dados da controladora de disco
  - 3) DMA começa a transferir dados entre o disco e a memória
  - 4) DMA avisa a CPU do fim da transferência
- Modo palavra X Modo de surto
- Modo direto.
- Por que os discos tem buffers? Para realizar a verificação de erros; para sincronizar a leitura do disco com o barramento
- Não há DMA em alguns embarcados, pois é mais barato deixar que a CPU administre a transferência de dados para a memória.

### 5.2 – Princípios do software de E/s

#### 5.2.1 – Objetivos do software de E/S

- Independência do dispositivo.
- Nomeação uniforme.
- Tratamento de erros o mais próximo possível do hardware.
- Tipo de transmissão.
- Utilização de buffers.
- Compartilhados X dedicados.

### 5.2.2 – E/S programada

- A CPU faz todo o trabalho.
- Espera ocupada, polling
- Simples, mas ocupa a CPU.
- Conveniente em sistemas embarcados.

### 5.2.3 – E/S usando interrupção

- Permite que a CPU faça outras coisas enquanto a E/S ocorre.
- Ainda dá muita responsabilidade a CPU.

### 5.2.4 – E/S usando DMA

- Interrompe menos a CPU.
- Requer um hardware especial, controladora de DMA.
- DMA é mais lenta que a CPU.

## 5.3 – Camadas do software de E/S

### 5.3.2 – Drivers dos dispositivos

- Os drivers de dispositivo são bem diferenciados.
- Geralmente escrito pelo fabricante e vem com o dispositivo.
- Cada driver trata de um dispositivo diferente.
- Drivers executam no modo núcleo.
- Driver vem abaixo do SO.
- Drivers são dinamicamente carregados durante a execução.
- Aceita e executa operações abstratas de leitura e gravação.
- O drive deve inicializar o dispositivo.
- Drivers reentrantes.

## 5.4 – Discos

### 5.4 – Hardware do disco

- Discos magnéticos são caracterizados por tanto a leitura quanto a escrita serem rápidos. Ideais para memórias secundárias.

#### Discos magnéticos

- Organizados em cilindros.
- Cada cilindro contém tantas trilhas quanto o número de cabeçotes.
- Cada trilha tem vários setores e o número de setores é fixo em todas as trilhas.

- Discos atuais contêm uma controladora capaz de remapear dados defeituosos e controlar a cache.
- Nem todos os parâmetros de um disco evoluíram linearmente durante os anos.
- Alguns discos têm zonas com um maior número de setores.
- Existe um disco virtual apresentado para o SO.

### RAID

- O desempenho da CPU aumenta mais que o desempenho dos discos.
- Arranjo redundante de discos baratos X disco único grande e caro
- Um RAID deve parecer com um SLED oferecendo melhor desempenho e confiabilidade.
- RAID0: Striping. Quebra o dado e separa nos discos. Desempenho excelente e implementação direta. Baixa confiabilidade.
- RAID1: Duplica-se os discos. Desempenho inferior, mas maior confiabilidade.
- RAID2: Usa o código de Hamming. Exige bastante do controlador e muitas exigências engessadas.
- RAID3: Versão simplificada do 2. Usa disco de paridade.
- O número de requisições de E/S que RAID 2 e 3 pode tratar não é melhor que num disco único.

### 5.4.2 – Formatação de disco

- Cada setor tem um preâmbulo, dados e o ECC.
- O preâmbulo contém informações sobre o setor.
- ECC contém informações para recuperação de erros.
- Cylindric skew.
- Head skew.
- A capacidade é reduzida por conta da formatação.
- Entrelaçamento simples.
- Entrelaçamento duplo.
- Registro mestre de inicialização.
- Formatação de alto nível.
- Bootstrap

#### 5.4.3 – Algoritmos de escalonamento de braço de disco

- Tempo de leitura depende: tempo de posicionamento, atraso de rotação e tempo de transferência real.
- FCFS
- Mantém-se uma tabela com as requisições.
- Pode considerar a próxima requisição como a mais próxima – SSF. As requisições mais longe do meio podem não ser bem servidas.
- Movem-se numa direção até não haver mais requisições naquela região e depois move na outra direção. Algoritmo do elevador. Para qualquer número de requisições o valor máximo para a distância total é fixo: duas vezes o número dos cilindros.
- Discos modernos leem vários setores para uma cache pra guardar.
- Os algoritmos são efetuados pelo controlador

#### 5.4.4 – Tratamento de erros

- Tecnologia atual não permite a criação de discos sem defeitos.
- Tratar erros via controlador ou via SO.
- Driver pode reinicializar o controlador.

#### 5.4.5 – Armazenamento estável

- Quando uma escrita é lançada para o disco, ou escreve-se corretamente nele ou não escreve-se nada.
- Objetivo de manter o disco consistente.
- Trabalha-se com um par de discos idênticos.
- Escrita estável: escrever e reler o bloco. Em cada HD, um por vez.
- Leitura estável: Ler de uma unidade, se não der certo n vezes ler da outra.
- Após uma falha os discos são igualados.
- Manter qual bloco estava sendo escrito para melhorar a performance.

### 5.5 – Relógios

- Temporizadores.
- Mantém a hora certa, controla a troca de contexto.

#### 5.5.1 – Hardware do relógio

- Oscilador + registrador contador + registrador de apoio.
- Valor colocado no contador conta até zero e quando chega em zero gera uma interrupção.
- Modo disparo e modo onda quadrada.
- Essas interrupções são os tiques do relógio.

#### 5.5.2 – Software do relógio

- Tudo é resolvido por software, o hardware só gera tiques.
- Manter hora do dia, timeslices, uso da cpu, chamadas alarm, temporizadores watch dog, perfis de execução.
- Os timeslices são guardados em tiques.
- O driver do relógio simula vários relógios virtuais

#### 5.5.3 – Temporizadores por software

- Interrupções são muito custosas nos computadores atuais.
- Os temporizadores por software evitam interrupções.
- Melhor que polling e interrupções.