Capítulo 7

Recursos ou Dispositivos

Trataremos aqui de algo muito importante, o gerenciamento de dispositivos de entrada e saída (E/S). O usuário final ou programador de alto nível não quer saber de detalhes técnicos do hardware, como é feito acesso a determinados dispositivo, sua velocidade, os parâmetros passados e/ou recebidos, operações envolvidas e coisas relacionadas com o acesso ao mesmo recurso ou dispositivo.

Existem camadas de software e hardware utilizadas na implementação de acesso a esses recursos, responsáveis pelo "mascaramento" de acesso em baixo nível do dispositivo. Elas se dividem em dois grupos, aonde em um os dispositivos são vistos como algo único, e no segundo aonde existe um para cada dispositivo.

7.1 Entendendo E/S

Quando um processo realiza uma operação do tipo E/S, o acesso ao mesmo deve parecer para a aplicação da forma mais amigável possível. Sendo assim quem desenvolve um sistema operacional deve se preocupar com todo o hardware subjacente conectado ao hardware principal, monitorando e desenvolvendo rotinas para tratamento do mesmo.

A maioria das rotinas das linguagens de programação de alto nível possui um meio de acessar recursos de uma forma simplista através do próprio encapsulamento ou mascaramento do sistema operacional.

Por exemplo, quando usamos uma instrução de cópia de um arquivo ou tabela de um banco de dados, não nos preocupamos quando vamos copiar para o disco rígido, disquete ou CD-ROM, apenas desejamos copiar e temos um comando pronto em nossas bibliotecas disponível para tal uso.

Acontece que o sistema operacional sabe que dependendo do destino do arquivo ou tabela, ele deverá atuar de uma forma, ou seja, usa uma interrupção, registradores e parâmetros diferentes para o acesso.

Tendo em vista isto, deve-se ter consciência que o para esse tipo de comunicação, fazemos uso das *system calls*, ou chamadas de sistema, que são rotinas presentes nas camadas superiores de um sistema operacional.

Resumidamente, temos aqui uma interação entre as rotinas de linguagens de programação de alto nível e as *system calls*, que trabalham através de passagem de mensagens com seus respectivos parâmetros e assim realizam a tarefa desejada, sem mostrar ao usuário final ou programador de alto nível todo o trabalho "bruto" realmente envolvido.

Posteriormente, temos algo chamado de sub-rotina de E/S, responsável por procedimentos padrões de dispositivos e recursos, onde os detalhes específicos de cada um fica a cargo dos *drivers* de dispositivos. Essa sub-rotina ou subsistema é a responsável real pelo acesso ao recurso desejado.

Vamos tomar como exemplo o Linux e seus acessos a cd-rom. Quando a unidade de cd-rom é montada para acesso a mesma, localiza-se geralmente no caminho "/mnt/cdrom". A função do sistema operacional é mapear o nome do dispositivo ao seu respectivo driver. Quando uma camada superior tentar fazer acesso ao cd-rom utiliza apenas o nome do dispositivo, e não o mecanismo todo de acesso ao mesmo.

Alguns recursos trabalham com unidades de medidas diferentes umas das outras, como bits, bytes, caracteres, etc. Esse subsistema ou sub-rotina cria uma unidade em comum para comunicação entre recursos ou dispositivos.

Para entender melhor o papel do subsistema basta ver o seu desempenho quanto ao disco rígido do computador, aonde o compartilhamento deve ocorrer de forma segura e simultânea entre todos os aplicativos e processos que desejam fazer acesso ao disco rígido. Ele cuida de todo o processo que possa interferir no perfeito funcionamento de um dispositivo ou recurso.

No caso de acesso a uma impressora aonde os trabalhos enviados de forma seqüencial, evita-se enviar os trabalho para ela a todo instante, mas primeiramente armazenar todos os trabalhos a serem impressos em uma área temporariamente (buffer) e depois sim enviar todos esses trabalhos para ela, sem que ocorra múltiplos acessos desnecessários a impressora ou qualquer outro recurso que permita o uso de buffer.

7.2 Dispositivos de E/S

Existem vários dispositivos de entrada (teclado, mouse, scanner etc.), saída (impressora, monitor etc.) e outros que realizam as duas funções (modem, disquetes etc.), que são responsáveis por desenvolver funções específicas.

A comunicação entre esses dispositivos e o sistema operacional ocorre por troca de blocos informações, monitorados pela UCP. Pela forma de como os dados são armazenados nesses dispositivos, temos classificações diferentes.

Na primeira, chamada de dispositivos de caractere ou não estruturados, os dados são enviados em seqüência de caracteres. Ou seja, o acesso ao dado não pode ser feito após a transmissão, pois a seqüência de caracteres não é endereçável, como podemos notar, por exemplo, nas impressoras.

No segundo tipo, chamados de dispositivos de blocos ou estruturados, os blocos de informações tem um tamanho fixo em bytes (32, 64, 128, e demais potências de 2) aonde o acesso aos dados pode ser feito independentemente através do endereço desejado, como podemos ver na estrutura de disquetes magnéticos, cujo acesso é direto por meio de endereçamento. Nas pouco utilizadas atualmente fitas magnéticas, o acesso estruturado é feito seqüencialmente, ao contrário dos disquetes, aonde o acesso é direto e endereçado.

7.3 Controladores de Dispositivos

Controladores de dispositivos são hardwares responsáveis por gerenciar outros hardwares. Para ilustrar melhor a situação, os drivers de dispositivos se comunicam não diretamente com um hardware (recurso ou dispositivo), mas comunica-se com eles através de um controlador.

Normalmente, um controlador possui memória própria e outros recursos disponíveis exclusivamente para ele para gerenciar o dispositivo que lhe cabe. Um exemplo clássico do uso de controlador é para discos rígidos SCSI, conhecidos por sua alta velocidade entre outras características que os destacam.

A comunicação neste caso é por blocos e utiliza-se uma técnica de transferência chamada DMA, que faz o trabalho entre o controlador do dispositivo SCSI e a memória principal. As operações de E/S realizadas pelo driver de dispositivo grava os comandos na memória do controlador, que processa as tarefas a UCP fica livre para outros processos. Ao final do processamento pelo controlador, então os resultados são retornados para memória principal e uma interrupção é gerada indicando o final do trabalho.

Resumindo, a controladora de discos SCSI avisa ao sistema operacional quando finalizar a operação de escrita dos dados do controlador para o disco rígido, ou ainda avisar o término da gravação quando os dados ainda estiverem na memória do controlador, de onde deduzimos uma das grandes características que destacam os dispositivos SCSI em termos de velocidade.

7.4 Drivers de Dispositivos

Como a maioria dos técnicos em informática sabe, esse é um termo muito utilizado na informática “prática” dos mesmos, e corresponde a um programa desenvolvido para estabelecer a comunicação com um hardware externo, dispositivo ou recurso. O subsistema trata de funções globais de dispositivos, entretanto os drivers geralmente são específicos para cada dispositivo em questão, variando de acordo com a marca, modelo, versão do sistema operacional e outras peculiaridades.

No sistema operacional Linux, que possui seu código fonte aberto, geralmente os drivers copiados dos sites dos seus respectivos fabricantes, vem em código fonte, ou seja, você precisa compilar ele e em seguida associá-lo ao dispositivo desejado. Em outros sistemas esse processo ocorre, pelo menos em teoria, de maneira mais simplista, aonde um driver já compilado está disponível no site do fabricante e você apenas copia e o instala, sem ter que gerar o executável dele propriamente dito.

O driver contém praticamente todas as rotinas utilizadas para se comunicar com o hardware. Estamos escrevendo um programa, e mandamos imprimir um relatório de clientes no mesmo, ele envia um comando padrão de impressão que chama uma system call de E/S, que faz acesso ao driver da impressora, que então interpreta os dados enviados, transforma esses dados em parâmetros compreensíveis pelo hardware em questão e então os envia e aguarda por uma resposta do mesmo, tratando essa resposta e traduzindo-a retornando, por exemplo, como “Impressão OK” ou “Não impressão”.

Todos esses drivers se aglomeram ao núcleo do sistema operacional, e quando são instalados passam a fazer parte do mesmo, disponibilizando todo o acesso disponível ao hardware que ele corresponda.

Exercícios do Capítulo 7

1. Qual a responsabilidade das rotinas do Sistema Operacional de tratamento de E/S?

2. O que é um driver de dispositivo?

3. Defina controladores de dispositivos e cite um exemplo.

4. Como podemos classificar os diferentes tipos de dispositivos segundo sua função? Cite dois exemplos de cada um.

5. Quem é o responsável pela troca de blocos de informações entre o sistema operacional e os dispositivos?