

João Pedro Miguel de Moura

Pedro Toupitzen Specian

LABORATÓRIO DE SISTEMAS OPERACIONAIS

ENTRADA E SAÍDA

TÓPICOS ABORDADOS

- ▶ Conceito de E/S
- ▶ Funcionamento Geral de E/S
- ▶ Pedido de Interrupção
- ▶ E/S no Linux
- ▶ Device Drivers
- ▶ Operações de E/S
- ▶ Dispositivos de Bloco
- ▶ Dispositivos de Caracteres
- ▶ Redirecionamento de E/S

CONCEITO DE E/S

- ▶ Segundo Tanenbaum*: "Um dos principais encargos de um sistema operacional é fazer o controle dos dispositivos de Entrada e Saída(Input/Output)
- ▶ O sistema deve enviar comandos aos dispositivos, analisar interrupções, tratar possíveis erros e prover uma interface que permita fazer chamadas ao sistema (system calls) para a leitura e a escrita nos dispositivos.
- ▶ É basicamente a comunicação do Sistema Operacional com todos os demais dispositivos.



*: Andrew Stuart Tanenbaum é o chefe do Departamento de sistemas de computação, na Universidade Vrije

FUNCIONAMENTO GERAL DE E/S

- ▶ O tratamento de E/S segue os padrões a seguir:
 - ▶ Independência do dispositivo: O sistema operacional deve ser capaz de acessar o dispositivo sem conhecê-lo previamente. Ou seja um programa deverá ser capaz de ler/escrever um arquivo da mesma forma para qualquer dispositivo. Deve haver uniformidade nos métodos de escrita e leitura.
 - ▶ Nomeação uniforme: O nome de um arquivo ou de um dispositivo deveria ser uma cadeia de strings ou um número inteiro totalmente independente do dispositivo. Dessa forma, os arquivos e diretórios são endereçados pelo nome do caminho. Sem possibilidade de conflitos.

FUNCIONAMENTO GERAL DE E/S

- ▶ O tratamento de E/S segue os padrões a seguir:
 - ▶ Tratamento de erros: De maneira geral, espera-se que os erros, como de leitura por exemplo, sejam tratados em níveis mais baixos, o mais próximo do hardware.
 - ▶ Tipos de conexão: Os dispositivos I/O podem se conectar de forma serial ou paralela. Na interface serial existe apenas uma linha por onde os dados trafegam. Na interface paralela os dados são transmitidos simultaneamente através das várias linhas para dados.
 - ▶ A transferência pode ser síncrona (bloqueante) - após um read, o programa é suspenso até que os dados estejam disponíveis no buffer
 - ▶ Ou assíncrona (orientada à interrupção) - a CPU inicia uma transferência e segue realizando outra atividade até ser sinalizada por uma interrupção (o que acontece na maioria das E/S físicas)

FUNCIONAMENTO GERAL DE E/S

- ▶ Existem requisições de entrada e saída que necessitam ser processadas pelo Sistema Operacional após serem emitidas ou recebidas.
- ▶ Esta requisição demanda tempo de CPU, e muitas vezes ela deve ser realizada de imediato (Como inputs no teclado que não podem esperar que a CPU fique livre para serem processados).
- ▶ Esta requisição é denominada de pedido de interrupção (IRQ).

PEDIDO DE INTERRUPÇÃO

- ▶ Um pedido de interrupção (abreviação IRQ (em inglês)) é a forma pela qual componentes de hardware requisitam tempo computacional da CPU. Um IRQ é a sinalização de um pedido de interrupção de hardware.
- ▶ Dessa forma IRQ's são canais de comunicação com o processador. Ao receber um pedido através de algum destes canais, o processador percebe a solicitação de interrompimento de um dispositivo.
- ▶ Quando um programa de usuário emite uma chamada ao sistema, esta é encaminhada ao driver apropriado. Para evitar que a CPU fique ocupada interrogando se dispositivo terminou a operação de E/S (espera ociosa), o driver pede ao dispositivo que lhe sinalize quando isso ocorrer. Dessa forma, o Sistema Operacional poderá executar outras tarefas enquanto o programa que o chamou pedindo o serviço se encontra bloqueado. Ao final da operação, o controlador do dispositivo gera uma interrupção, ao chip controlador de interrupção, para sinalizar à CPU.

PEDIDO DE INTERRUPÇÃO

- ▶ Caso nenhuma outra interrupção esteja pendente ou em tratamento e nenhum outro dispositivo fez uma requisição de maior prioridade no mesmo momento, a interrupção é tratada imediatamente. Caso contrário, ela é ignorada, e o dispositivo continuará emitindo sinal de interrupção.
- ▶ Após executar uma instrução, a CPU verifica as linhas de interrupções para saber se alguma interrupção foi sinalizada.
- ▶ Caso tenha sido sinalizada uma interrupção, o hardware grava os registradores da CPU na pilha do programa que estava em execução e carrega o contador de programa com o endereço da rotina referente à interrupção sinalizada

PEDIDO DE INTERRUPÇÃO

- ▶ Interrupções podem ser geradas por hardware ou por software. No caso do hardware, pelo dispositivos periféricos ou pelo relógio (timer).
- ▶ As interrupções geradas por software são conhecidas como system calls (chamadas ao sistema) ou trap (armadilha).
- ▶ É possível prever quando as interrupções de software irão acontecer, o que não é possível no caso das interrupções de hardware.

E/S NO LINUX

- ▶ O sistema de E/S do Linux fornece uma visão semelhante a de qualquer sistema UNIX, onde os drivers de dispositivos (device drivers) são apresentados para o usuário como arquivos normais, a fim de abstrair o resto do sistema ou o usuário das particularidades do hardware. Assim, o acesso a um dispositivo ocorre da mesma maneira da abertura de um arquivo.
- ▶ Estes arquivos que fornecem acesso a um dispositivo são denominados arquivos especiais e estão associados a um diretório que se encontra dentro de /dev.
- ▶ Os arquivos especiais podem ser acessados da mesma maneira que qualquer outro arquivo por meio de system calls de leitura e escrita.

E/S NO LINUX

- ▶ Estes arquivos especiais são arquivos com referências a um driver de dispositivo. Assim, permissões de leitura e escrita nos dispositivos são concedidas aos usuários do sistema operacional da mesma forma com que é feita em arquivos.
- ▶ Tipos de permissão dependerão do dispositivo
 - ▶ Monitores, impressoras, headphones são dispositivos de saída apenas.
 - ▶ Mouse, teclado, joystick, scanners são dispositivos de entrada apenas.
 - ▶ HDs, SDs, dispositivos USB(pendrives, câmeras, celulares) podem conceder tanto permissão de escrita quanto de leitura.
- ▶ (aprofundamento mais adiante)

DEVICE DRIVERS

- ▶ Os drivers de dispositivos são conectados ao núcleo (kernel) através de módulos carregáveis, que podem ser ligados ao núcleo, quando este estiver em atividade. Basicamente o sistema operacional utiliza o ponteiro para a função de ativação do dispositivo e o inclui junto ao Kernel.
- ▶ Logo, ao se acrescentar ou remover dispositivos, o correspondente driver deve ser ativado ou desativado, respectivamente, junto ao kernel. Isto evita uma ação manual no núcleo para inclusão ou remoção de drivers.

DEVICE DRIVERS

- ▶ O linux permite apenas um driver por dispositivo, mas pode-se utilizar o mesmo driver para dispositivos diferentes, como no caso de pendrives que utilizam o mesmo driver feito para tratar discos SCSI*.
- ▶ Assim, há para cada driver um número de dispositivo principal que o identifica e também um número de dispositivo secundário que identifica o dispositivo em si.

*: SCSI é padrão de tecnologia que permite ao usuário conectar uma larga gama de periféricos. Para que um dispositivo SCSI funcione em seu computador é necessário ter uma equipamento que realize a interface entre a máquina e o hardware SCSI. Essa interface é chamada de "Host Adapter".

O máximo de conexões permitidas no padrão SCSI é de 15 dispositivos que são identificados por um código binário, chamado ID SCSI. Só é permitida a transmissão entre dois dispositivos de cada vez.

OPERAÇÕES DE E/S

- ▶ Nos sistemas UNIX, os arquivos especiais são separados em duas classes: blocos e caracteres (no Linux também existem os arquivos especiais de redes). Cada classe é representada por uma tabela.
- ▶ Seguindo o número principal do driver você pode encontrar a tabela e bdevsw, tabela dos arquivos especiais de blocos e a cdevsw, tabela dos arquivos especiais de caracteres.
- ▶ Para cada linha dessas tabelas associasse um driver para um dispositivo, e em cada coluna de tais linhas há um ponteiro para uma função respectiva daquele dispositivo.

OPERAÇÕES DE E/S

- ▶ As funções mais comuns são: `open`, `close`, `read`, `write`, `ioctl`. Estas funções são referenciadas na tabela de arquivos especiais em suas respectivas colunas. A coluna será nula, caso não exista aquela determinada função, como no caso da memória, que não apresenta `open` nem `close`, apenas `read` e `write`.
- ▶ As operações de I/O não são realizadas usando registros ou métodos de acesso. Em vez disso, elas são realizadas como uma sequência de bytes.
- ▶ Com as operações sendo meras sequências de bytes, as `system calls` de I/O podem manipular qualquer tipo de dispositivo de forma uniforme.
- ▶ Estas operações são tratadas de modo diferente pelos dispositivos de bloco e os de caracteres.

DISPOSITIVOS DE BLOCO

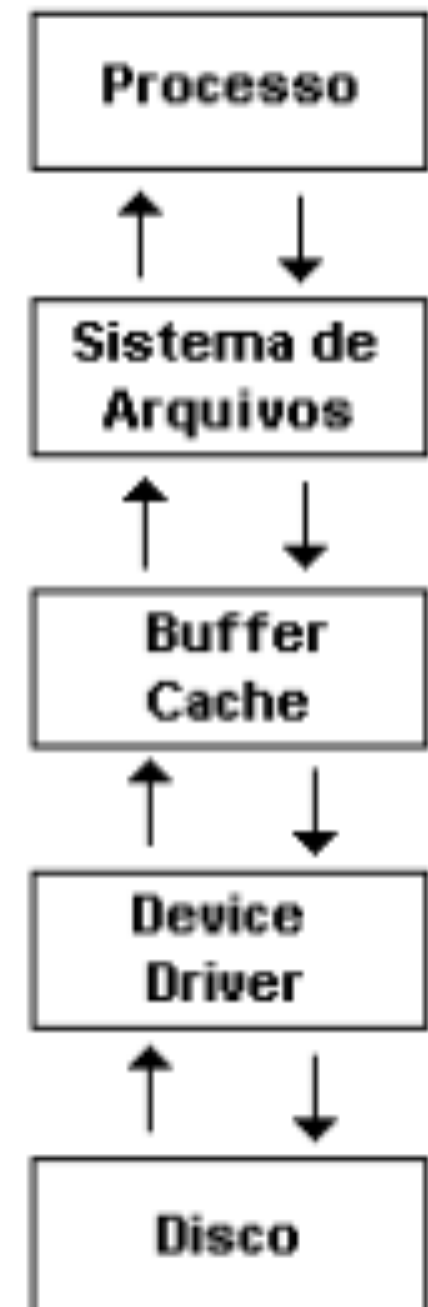
- ▶ Os blocos são unidades enumeradas com tamanho fixo em uma sequência. Assim, cada bloco tem seu endereço próprio e possui acesso individual, ou seja, apresenta leitura e escrita independente dos outros blocos.
- ▶ Geralmente estão associadas a dispositivos com altas taxas de transferência (discos, por exemplo)
- ▶ Para o sistema operacional é de fundamental importância o desempenho dos discos. Então, os dispositivos de blocos devem ser preparados para fornecer funcionalidades que mantenham uma qualidade do acesso a este meio, isto é, que seja feito da melhor forma possível

DISPOSITIVOS DE BLOCO

- ▶ Tem o objetivo de minimizar o n° de transferências entre o dispositivo e a memória, usando buffer e caches.
- ▶ Quando uma operação de leitura a disco é realizada, um bloco é transferido para a memória e depois processado.
- ▶ Quando uma operação de escrita em disco é realizada, o bloco é transferido para a memória e seus dados são posteriormente escritos no disco.

DISPOSITIVOS DE BLOCO

- ▶ Utilizado para minimizar a quantidade de transferências entre o dispositivo e a memória. É uma tabela situada no kernel que abriga os blocos mais recentemente utilizados. Então, quando é necessário acessar um bloco, deve verificar antes se ele está nesse cache, evitando assim, o acesso ao disco.
- ▶ No Linux, o cache de buffers de bloco engloba duas características principais, ser um pool de buffers para E/S ativas e um cache para E/S concluídas. Pois o cache manipula tanto escritas quanto leituras.



DISPOSITIVOS DE BLOCO

- ▶ Além dos buffers, o cache de buffers é formado também pelos descritores destes buffers, chamados de buffer heads, um para cada buffer.
- ▶ Os buffer heads contém as informações dos buffers que o kernel necessita para controlá-los. Como qual dispositivo de bloco o buffer está servindo, a parte (o deslocamento mais o tamanho) dos dados que este buffer refere no dispositivo, que juntos formam o identificador do buffer.

DISPOSITIVOS DE BLOCO

- ▶ Para o controle, o kernel utiliza quatro listas de buffers: intacta ou limpa, modificada ou suja, bloqueada e livre. Nessas listas os últimos blocos acessados são colocados no início da lista. Assim, quando é necessário retirar blocos, remove os menos recentemente usados, ou seja, os que estão no final da lista.
 - ▶ Lista intacta é aquela contém buffers que não sofreram alterações.
 - ▶ Lista modificada é aquela que contém os que sofreram e necessitam ser gravados para não se perderem.
 - ▶ Lista bloqueada é a que contém buffers que não podem ser modificados.
 - ▶ Lista livre é a que contém buffers não designados.

DISPOSITIVOS DE BLOCO

- ▶ A lista modificada é descarregada à força no disco quando está cheia. Mas para evitar uma espera grande, os blocos desta lista são escritos a cada 30 segundos no disco. Já na lista livre, pode-se inserir blocos por eventos de sistema de arquivos, como no caso de um arquivo ser deletado. Mas dela é removida pelo kernel, quando este tem necessidade de mais buffers. O kernel também pode ampliar a lista caso haja memória livre disponível.

DISPOSITIVOS DE BLOCO

- ▶ Há um controlador responsável pela gerência de requisições de leitura e escrita feita pelos drivers dos dispositivos: O gerente de pedidos. Ele utiliza o buffer head mais a determinada ação (leitura ou escrita) e encaminha um pedido, mas aguarda sua conclusão. Assim, ele guarda o ponteiro da lista de buffer head emitida. O pedido é removido da lista apenas quando concluído o evento de E/S.

DISPOSITIVOS DE BLOCO

- ▶ Ele reserva uma lista de pedidos para cada driver de dispositivo de bloco. Mas, ao surgir novos pedidos, ele procura agrupar pedidos antes de encaminhar e bloqueia o grupo.
- ▶ Desta maneira, pedidos são enviados agrupados para diminuir o número de requisições individuais, aumentando a eficiência.
- ▶ Porém, quando os buffer heads vão sendo atendidos, eles já vão sendo desbloqueados individualmente, a fim de não causar espera até terminar de atender todo o grupo.

DISPOSITIVOS DE CARACTERES

- ▶ Empregados em dispositivos que não apresentam acesso aleatório para blocos de dados. Geralmente utilizados para dispositivos que realizam E/S por meio de fluxo de caracteres com baixa taxa de transferência, tais como, impressoras, mouses, teclados.
- ▶ Antigamente, dispositivos de rede eram considerados dispositivos de caracteres. Mas hoje não são mais dispositivos demorados, assim como as redes antigas. E para se acessar dados é necessário abrir uma conexão junto ao kernel.

DISPOSITIVOS DE CARACTERES

- ▶ Os dispositivos de caracteres devem prover ao kernel funções para suas operações de E/S, pois o kernel não faz nenhum pré-processamento, apenas encaminha pedidos de leitura ou escrita para estes dispositivos, que ficam responsáveis pelo tratamento dos pedidos.
- ▶ Dispositivos de terminais são uma exceção: O kernel mantém um buffer para o fluxo de dados e oferece uma interface padrão em um interpretador denominado de tratador de linhas (line discipline). Este tratador substitui tabulação por espaços, retorno de carro por avanços de linha, operando como uma espécie de filtro. A tty é o tratador mais comum e faz a comunicação do fluxo básico de E/S entre processos e terminais.

REDIRECIONAMENTO DE E/S

- ▶ No Linux, é possível redirecionar os fluxos de I/O padrão
- ▶ O usuário pode redefinir de onde um comando/programa receberá sua entrada e para onde sua saída irá.
- ▶ A entrada de um comando são os dados sobre os quais o comando irá operar.
- ▶ Tais dados podem vir de um arquivo especificado pelo usuário, de um arquivo de sistema, do terminal ou mesmo da saída de um outro comando.

REDIRECIONAMENTO DE E/S

- ▶ A saída de um comando é o resultado da operação que ele realiza sobre a entrada.
- ▶ A saída de um comando pode ser impressa na tela do terminal, enviada a um arquivo ou servir como entrada para outro comando.

CONCLUSÃO

- ▶ SILBERSCHATZ, A., Galvin, P., Gagne, G. Sistemas Operacionais: Conceitos e Aplicações ;tradução de Adriana Richie. 1a Edição. Elsevier. Rio de Janeiro. 2000.
- ▶ TANENBAUM, Andrew S. Sistemas Operacionais Modernos. 2 Edição. Prentice Hall. 2003.
- ▶ wikibooks, Sistemas operacionais/Gerenciamento de dispositivos de entrada e saída: https://pt.wikibooks.org/wiki/Sistemas_operacionais/Ger%C3%Aancia_de_dispositivos_de_entrada_e_sa%C3%ADda