TCHELINUX 2007

Dispositivos de Bloco e Sistemas de Arquivos no Linux

Fábio Olivé
<olive@tchelinux.org>

Douglas Landgraf
<dougsland@tchelinux.org>

Tópicos Abordados

- O que são dispositivos de bloco
- Drivers de dispositivos de bloco
- Mecanismo versus Policy
- Sistemas de Arquivos
- Chamadas de sistema do VFS

O que são Disp. de Bloco?

- Informação acessada em blocos de tamanho fixo
- Qualquer bloco pode ser acessado a qualquer momento
  - Embora nem sempre a velocidade seja igual
- Geralmente é armazenamento estável e de massa
  - Um monte de espaço, e continua lá quando desliga
- O tamanho do bloco (setor) e sua quantidade variam
  - Discos: 512 bytes
  - CD/DVD: 2048 bytes
  - Bloco físico (setor) != bloco lógico (“bloco”:)

O que são Disp. de Bloco?

- Exemplos típicos:
  - Discos rígidos
  - CD/DVD
  - Flash drive (memória flash apresentada como disco)
- Exemplos não tão típicos:
  - Ramdisk (driver que simula blocos em memória)
  - Loop device (driver que simula blocos em arquivos)

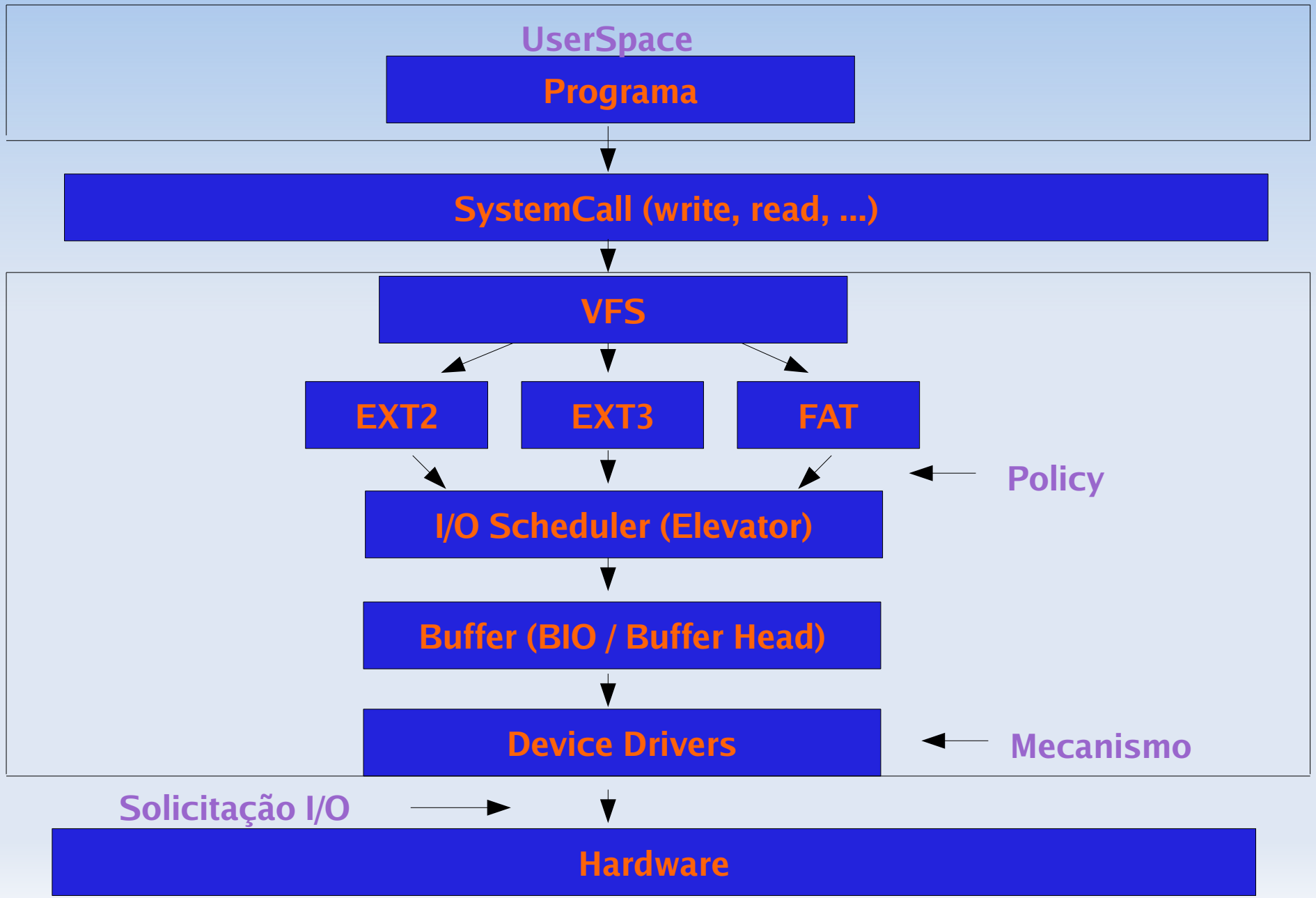
O que são Disp. de Bloco?

- Discos tem uma “geometria”
  - Cilindros, cabeças, setores por cilindro
  - Totalmente inútil em qualquer disco atual
  - É um legado com que temos que viver
- CD/DVD são uma seqüência de blocos
  - Uma grande espiral, como um disco de vinil
  - Lento!
- E o que são partições?
  - Mera convenção social (particionar não estraga!)

Drivers de Disp. de Bloco

- Adaptam os detalhes de um controlador específico para a visão genérica do kernel
- Implementam o MECANISMO de acesso
  - COMO ler e gravar blocos?

Drivers de Disp. de Bloco



Mecanismo versus Policy

- Saber COMO escrever é uma coisa
  - **Saber O QUÊ, QUANDO e o PORQUÊ é outra**
- Geralmente os drivers implementam mecanismos, ou seja, como é que se faz para um dispositivo funcionar e estar disponível
- Outras camadas (superiores) do kernel implementam o que se conhece como Policy, que é o que dá **significado** às operações do driver

Mecanismo versus Policy

- Mecanismo está mais perto do hardware
- Policy está mais perto do usuário
- Usuários gravam arquivos, não blocos
  - Esperem pra pensar nisso mais tarde
- Existem várias camadas de Policy sobre um driver de dispositivo de bloco
  - Desempenho, consistência, semântica, ...

Mecanismo versus Policy

- Desempenho
  - Elevator, ou “I/O scheduler”
  - Diferentes tecnologias de armazenamento possuem diferentes características de desempenho dependendo do padrão de acesso
  - Com a falácia da “geometria”, fica cada vez mais difícil saber onde o disco vai gravar os dados
  - Operações típicas:
    - Agrupar requisições contíguas
    - Ordenar requisições na fila

Mecanismo versus Policy

- Consistência
  - O Sistema de Arquivos pode ter certas restrições quanto à ordem com que os blocos são escritos
    - Journal DEVE ser atualizado antes dos meta-dados
    - Nós da árvore DEVEM ser atualizados na ordem emitida
    - Inode DEVE ser marcado como utilizado
    - Etc etc etc
  - Estes detalhes podem ser implementados por uma camada independente de dispositivo ou sistema de arquivos

I/O Scheduler Linux (DEMO)

- make menuconfig

Mecanismo versus Policy

- Semântica
  - Cada Sistema de Arquivos organiza os blocos de um jeito diferente, portanto cada bloco tem uma função ou significado diferente
    - Superbloco, bitmaps, inodes, indirect blocks
  - Que tipo de informação cada bloco contém?
    - É possível dizer de forma estática/determinística?
  - O que significa escrever ou ler cada bloco?
  - Alguns blocos são mais importantes que outros

Sistemas de Arquivos

- Definem como um dispositivo de blocos vai ser utilizado pelo sistema
- Gasta alguns blocos para organizar o resto
- Principais abstrações/objetos:
  - Arquivo: contém dados
  - Diretório: contém listas de arquivos ou diretórios
    - Geralmente um diretório é um arquivo “especial”

Sistemas de Arquivos

- Diferentes sistemas de arquivos armazenam diferentes informações para cada arquivo
  - Ext3:
    - Tipo, dono, grupo, permissões, número de links, timestamps detalhados de alta precisão, atributos simples e extendidos, ...
    - Feito para Sistemas Operacionais multi-usuário
  - FAT:
    - Praticamente só atributos simples e timestamp ralado
    - Feito para Sistemas Operacionais mono-usuário

Syscalls do VFS

- Demonstração do código de sys_open, sys_read, etc..

Dúvidas ?
Sugestões ?

Fabio Olivé
<olive@tchelinux.org>

Douglas Landgraf
<dougsland@tchelinux.org>