

Universidade Estadual de Campinas
Instituto de Computação

MO806/MC914
Tópicos em Sistemas Operacionais

Seminário: MINIX

Daniel Bruno Alves dos Santos
danielbrunoalves@gmail.com

Roteiro

- A história do MINIX
- MINIX2 versus MINIX3
- Processos
- Entrada e saída
- Gerenciamento de memória
- Sistemas de arquivos
- Referências

A história do MINIX

- Criado por Tanenbaum com propósitos acadêmicos, para exemplificar os conceitos de seu livro: “Sistemas Operacionais: projeto e implementação” (1987)
- Deriva da palavra (mini-UNIX)
- Microkernel
- Projetado para ser compatível com a versão 7 do UNIX
- Escrito a partir da linguagem C, como o UNIX
- Padrão POSIX para as chamadas de sistemas
- Inspirou a criação do Kernel do Linux
- Atualmente encontra-se na versão 3 (2005)

MINIX

- O MINIX destina-se a computadores pessoais e não a sistemas de compartilhamento de tempo de grande porte
- Foi projetado para funcionar em todos os IBM PC

Versões do MINIX

- Versão 1 (1987):
 - Compatível com a versão 7 do UNIX
 - Cerca de 12.000 linhas de código C, contemplando Kernel, gerenciamento de memória, sistemas de arquivos
 - Liberou o código fonte e binários através de diskets com um manual de referência
 - Grupo de discussão USENET (cerca de 40.000 assinantes, em 3 meses)

Versões do MINIX

- Versão 2 (1997):
 - Baseado na segunda versão do livro de Tanenbaum [1]
 - Disponível apenas para x86, Solaris em arquitetura SPARC
 - Adicionou o padrão POSIX
 - Distribuído a partir de CD-ROM

Versões do MINIX

- Versão 3 (2005):
 - Foi reprojeta para ser “usada como um sistema sério sobre recursos limitados e computadores embarcados e para aplicações que requerem alta confiabilidade” [4]
 - Suporta apenas arquitetura IA-32
 - Disponível a partir de LiveCD
 - Licença FreeBSD

MINIX2 versus MINIX3

- Melhorias em Relação à versão 2:
 - Novas características:
 - Instalação baseada em LiveCD
 - Sistema de janelas X Window
 - Suporte até 4GB de memória
 - Blocos de disco com tamanhos: 1, 2, 4 ou 8KB
 - Adição das chamadas de sistema “Select”
 - Inclusão de novos programas: gcc, g++, emacs, python, perl, etc
 - Servidor de informação para depuração
 - *Reincarnation Server* que “mata” e substitui drivers defeituosos em tempo de execução

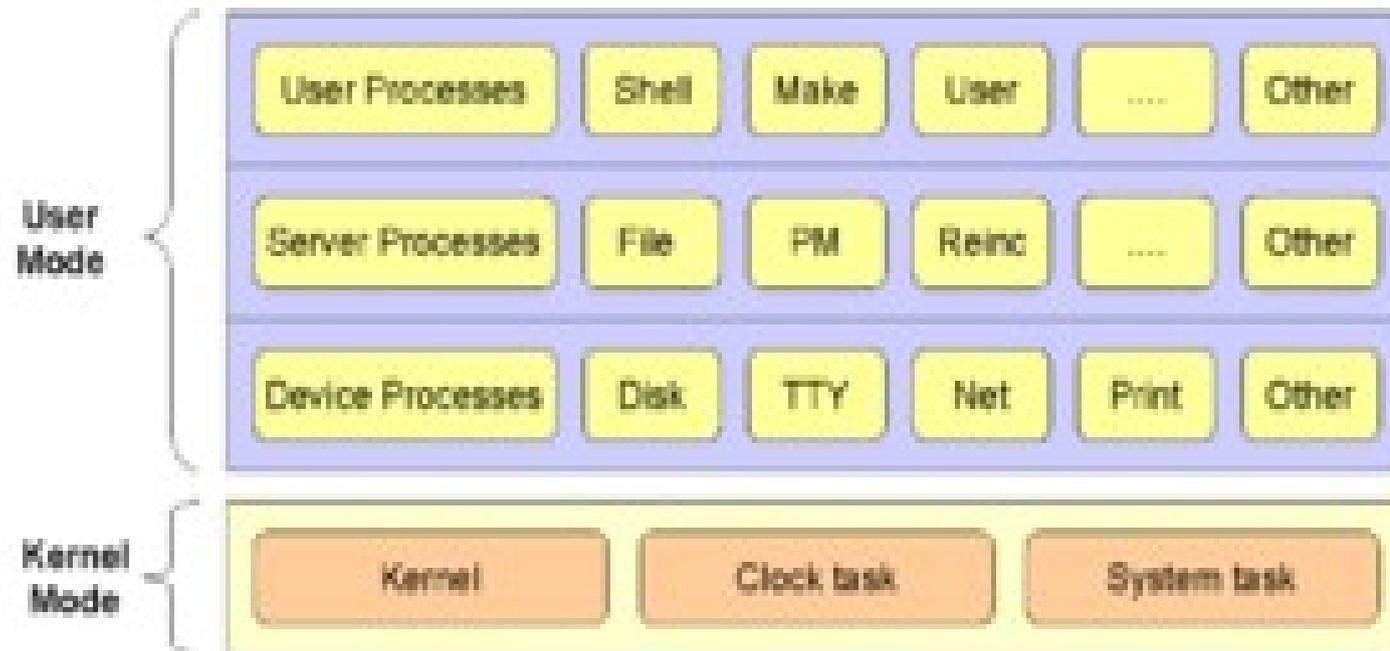
MINIX2 versus MINIX3

- Estrutura de Sistema:
 - Reescrita do Kernel e diminuição de cerca de 4000 linhas de código
 - Cada *driver* de dispositivo (exceto o relógio) é um processo do usuário separado
 - Novas características de confiabilidade
 - Mecanismo de Comunicação entre Processos não bloqueante
 - Melhoria no gerenciamento do Timer
 - Escalonador se tornou mais geral
 - etc...

Processos no MINIX

- O MINIX é uma coleção de processos que se comunicam entre si e com processos de usuários através de mensagens
- É estruturado em camadas, que executam funções específicas
- A comunicação é feita na mesma camada e/ou da camada superior com a inferior

Processos – Estrutura Interna



The MINIX 3 Microkernel Architecture

Fonte: Wikimedia

Processos – Estrutura Interna

- A camada inferior:
 - Captura todas as interrupções e traps (interrupções de software)
 - Faz o escalonamento e fornece modelo de processos sequenciais independentes que se comunicam por mensagens
- A camada 2:
 - Contém os processos de E/S, chamados de **tarefas e/ou drivers de dispositivos**
- Todas as tarefas na camada 2 e todo código na camada 1 formam o Kernel

Processos – Estrutura Interna

- Camada 3:
 - Contém processos que fornecem serviços úteis para os processos de usuário
 - Realiza a interpretação das chamadas de sistemas
 - Os processos de servidor executam em um nível menos privilegiado que o kernel e as tarefas que não podem acessar E/S diretamente
 - Exemplos:
 - Gerenciador de memória e Sistemas de arquivos

Processos – Estrutura Interna

- Camada 4:
 - Contém todos os processos de usuário
 - Exemplos:
 - *Shells*, editores, compiladores e programas escritos pelos usuários

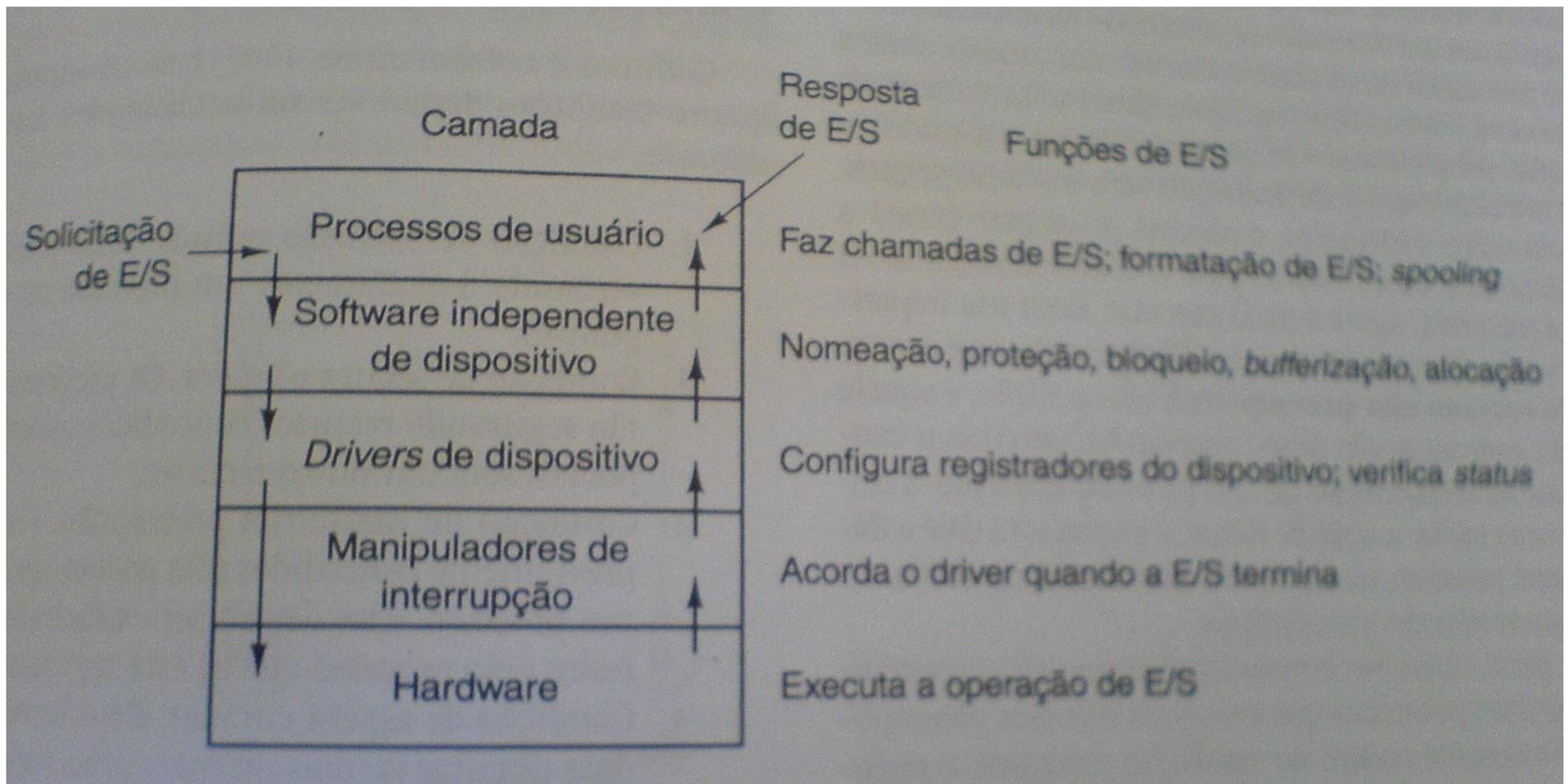
Processos no MINIX

- Gerenciamento de processos:
 - Chamadas de sistema FORK e EXEC
- Comunicação interprocessos:
 - através de mensagens
- Agendamento de processos:
 - Filas em 3 níveis, correspondendo às camadas 2, 3 e 4
 - Tarefas e servidores executam até bloquearem
 - Processos de usuário são escalonados por *round robin*
 - Tarefas tem a prioridade mais alta, seguido do gerenciador de memória e servidor de arquivos e por fim, processos do usuário

Entrada e Saída (E/S)

- Funções:
 - Controlar todos os dispositivos de E/S
 - Enviar comandos para todos os dispositivos
 - Capturar interrupções
 - Tratar erros
 - Oferecer uma interface entre os dispositivos e o restante do sistema que seja simples e fácil de usar

Entrada e Saída no MINIX



Fonte: Tanenbaum (2000)

Gerenciamento de memória

- Conceitos:
 - Hierarquia de memória
 - Gerenciador de memória (parte do SO que gerencia a hierarquia)
- Objetivo:
 - Controlar que partes da memória estão em uso e que partes não estão
 - Alocar memória para processos quando estes precisam e desalocar quando terminam
 - Gerenciar a troca entre a memória principal e o disco

Gerenciamento de memória no MINIX

- Gerenciamento simples: não utiliza paginação nem troca (*swap*) com o disco
- Compatibilidade com computadores antigos
- Não faz parte do Kernel
- É tratado pelo gerenciador de memória que executa no espaço do usuário e comunica-se com o Kernel por mensagens
- Memória é alocada quando um processo executa chamadas de sistemas como FORK ou EXEC

Gerenciamento de memória no MINIX

- O gerenciador de memória monitora a memória livre através de uma lista de lacunas e do algoritmo do primeiro ajuste
- O seu maior trabalho é executar as chamadas de sistemas relacionadas com o gerenciamento de memória

Sistemas de arquivos

- O sistema de arquivos deve:
 - Alocar e desalocar espaço para arquivos
 - Monitorar blocos de disco e liberar espaço
 - Proteger os arquivos contra uso não autorizado

Sistema de arquivos no MINIX

- Encontra-se fora do Kernel, no espaço de usuário
- Por esta razão pode ser usado como servidor de arquivos de rede independente
- Implementado todo em C
- Copiou a estrutura básica do Sistema de arquivos do UNIX
- Evitou características complexas

Sistema de arquivos no MINIX

- As vantagens dessa independência do sistema de arquivos são:
 - Pode sofrer modificações quase que de forma independente do restante do MINIX
 - Pode ser inteiramente removido, recompilado e usado como servidor remoto

Sistema de arquivos no MINIX

- Projeto e implementação:
 - Bloco de Boot (Boot Block)
 - SuperBloco (Superblock)
 - Inode bitmap
 - Zone bitmap
 - Inode
 - Área de dados (data area)

Sistema de arquivos no MINIX

- O MINIX utiliza um cache de blocos para melhorar o desempenho do sistema de arquivos
- O cache é implementado como uma matriz de buffers
- Todos os buffers que não estão sendo utilizados são encadeados em uma lista duplamente encadeada do mais recente utilizado (MRU) para o menos recentemente utilizado (LRU)

Referências

- [1] Tanenbaum, Andrew S; Woodhull, Albert S. **Sistemas operacionais: projeto e implementação**. Tradução de Edson Furmankiewics. 2. ed. Porto Alegre: Bookman, 2000.
- [2] Tanenbaum, Andrew S. **Sistemas operacionais modernos**. 2.ed. São Paulo: Prentice Hall, 2003.
- [3] **MINIX3**. Disponível em: <<http://www.minix3.org/>>, Acesso em: 26 Nov. 2007.
- [4] **Wikipedia**. Disponível em: <<http://en.wikipedia.org/wiki/Minix>>, Acesso em: 26 Nov. 2007.
- [5] **Wikipedia**. Disponível em: <http://en.wikipedia.org/wiki/MINIX_file_system>, Acesso em: 26 Nov. 2007.
- [6] **Some Notes on the "Who wrote Linux" Kerfuffle**. Disponível em: <<http://www.cs.vu.nl/~ast/brown/>>, Acesso em: 26 Nov. 2007.

Dúvidas?