

Sistema Operacional BeOS

Isaac Newton Beltrão

Universidade Federal Rural de Pernambuco, Curso de Licenciatura
em Computação

Caixa Postal: 52 171-900 – Recife - Pernambuco - Brasil
newtonbeltrao@yahoo.com.br

Abstract

The article below describes the characteristics of the operational system BeOS, how much its structural aspects, implemented architecture and technologies. Citing between them, the preemptive multitask, the support the symmetrical multiprocessing, has supported multi-threading, resources of I/O, system of archives of 64 bits with journaling, memory management, and the fact of the operational system to be guided objects, and to be come back toward applications multimedia.

Resumo

O artigo abaixo descreve as características do sistema operacional BeOS, quanto a seus aspectos estruturais, arquitetura e tecnologias implementadas. Citando entre elas, a multitarefa preemptiva, o suporte a multiprocessamento simétrico, suporte a multi-threading, recursos de I/O, sistema de arquivos de 64 bits com journaling, gerência de memória, e o fato do sistema operacional ser orientado a objetos, e ser voltado para aplicações multimídia.

1. Introdução

O BeOS é um Sistema Operacional moderno em sua essência. Ao contrário dos SO's mais antigos, o sistema foi desenvolvido desde o princípio com o objetivo de maximizar o desempenho e com tecnologias realmente inovadoras.

O BeOS consegue extrair excelentes performances de computadores comuns como os PCs e PowerPCs (Mac), permitindo uma grande economia no investimento em equipamentos. Essa diferença começa a ser notada desde o boot do sistema, já que todo o processo demora algo em torno de 12 segundos.

Surgiu em 1990 nos EUA, no estado da Califórnia, nas proximidades da cidade de San Francisco. Desenvolvido pela Be Inc. O seu idealizador e proprietário chama-se Jean-Louis Gassée – ex-vice-presidente da Apple. Quando o BeOS surgiu, a finalidade era

trazer o multi-processamento para o Computador Pessoal (PC). A idéia era juntar 5 processadores simples e muito baratos que fariam a mesma função de um 486. Acontece que, para fazer tudo isto, era necessário ter um Sistema Operacional que suportasse mais de um processador. Foi criado então o BeOS.

O BeOS usa símbolos e idéias familiares de outros sistemas operacionais (arquivos, pastas, desktops, etc.). O BeOS GUI é conhecido como Tracker, e a chave ao Tracker é o TrackBar, que pode ser movido e configurado segundo as preferências do usuário.

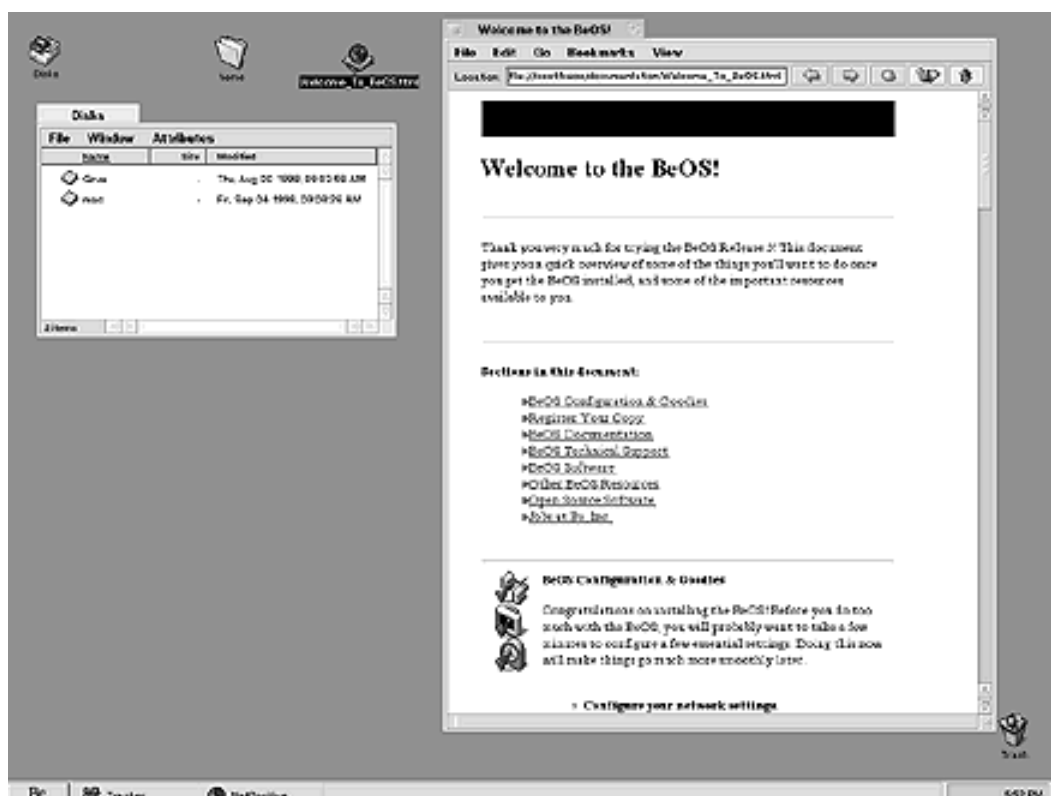


Figura 1. Interface do BeOS

2. Estrutura do Sistema Operacional

A arquitetura do BeOS é composta basicamente de três camadas:

- Microkernel e Drivers de dispositivo;
- Servidores Do BeOS;
- Kits De software do BeOS;

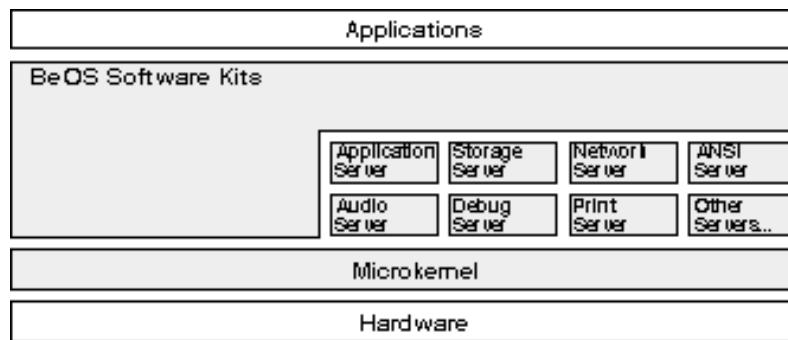


Figura 2. Arquitetura em camadas no BeOS

O MicroKernel trabalha diretamente com o hardware. Ele toma conta das tarefas de baixo nível e de controle crítico além de controlar o acesso à memória. O kernel também fornece outras coisas que os programas utilizam como ferramentas para o sistema de arquivos, escalonamento de threads e primitivas de chaveamento da memória.

Os Servidores estão na camada acima do kernel. Esta camada é composta por um número de processos de servidores que são executados em background e cuidam das tarefas para as aplicações que estão executando. A arquitetura cliente-servidor, no BeOS faz uso de protocolos de comunicação simples do tipo requisição/resposta. A fim de obter um serviço, um processo cliente envia uma requisição ao servidor. Este, por sua vez, executa as operações associadas ao serviço e envia uma resposta ao cliente, contendo dados ou um código de erro caso o serviço não possa ser executado.

A camada acima dos servidores é composta pelos kits de software. Um Kit consiste em um número de classes de objetos orientados que o programador utiliza para desenvolver aplicações para o BeOS. Coletivamente, estas classes juntas compõem a API do BeOS.

2.1. Sistema Operacional Orientado a Objetos

O BeOS foi desenvolvido com base nesse paradigma da Engenharia de Software. Até mesmo o kernel do BeOS é Orientado a Objeto.

Para um usuário, a coisa mais visível é o fato de ser possível customizar o SO a níveis inimagináveis sem muito esforço, o que faz com que surjam muitos programas destinados a customização (principalmente no que diz respeito à Interface). Para o programador, isso significa que é fácil reaproveitar e melhorar o que o SO fornece, sem ter que reescrever partes inteiras do mesmo (conceito de reusabilidade).

2.2. Gerência de Processos: SMP

SMP significa *Simetrical Multi Processor* - Multiprocessamento Simétrico. Neste conceito, um grupo de processadores trabalha conjuntamente. As aplicações são divididas em correntes que podem rodar concorrentemente em qualquer processador que esteja disponível.

O SMP melhora o desempenho da própria aplicação, como também o processamento total do sistema. Os sistemas SMP requerem alguma forma de memória compartilhada e caches de instrução locais, e o mais importante: sistemas de SMP requerem aplicações que podem tirar proveito do paralelismo.

Como o BeOS, desde o princípio, foi criado para ser multiprocessado. Assim, quando surgiu o SMP bastou a Be Inc incorporá-lo ao seu sistema operacional, sendo que todo o seu funcionamento já era voltado para mais de um processador.

2.2.1. Multitarefa Preemptiva

O BeOS é um sistema operacional que foi projetado para dividir a aplicação em vários threads mesmo que o programador não tenha projetado assim. Isso aumenta a eficiência e a performance das aplicações. Na multitarefa preemptiva, o compartilhamento da CPU é gerenciado por um componente do sistema operacional chamado escalonador (*scheduler*), que atribui tempo de CPU para as várias tarefas baseadas em um conjunto de heurísticas. O escalonamento é feito por Prioridade.

Desse modo, nem o programador nem o usuário necessitam conhecer os detalhes. No BeOS o recurso de multitarefa é levada a um patamar acima, através do recurso de Multithreading. Uma Thread é um pedaço (geralmente uma função) de um programa que roda simultaneamente com outras threads no espaço alocado pelo SO para a aplicação. No caso do BeOS, o Multithreading é pervasivo, ou seja, existem muitas threads mesmo ao nível mais baixo do SO (O kernel, por exemplo, é composto por 4 threads).

2.2.2. Multi Threading

No BeOS o processador não se torna mais rápido que em outros sistemas operacionais, mas as aplicações ficam mais sensíveis, uma vez que elas recebem mais atenção do processador todo o tempo. O conjunto de instruções não é executado seqüencialmente como outros sistemas fazem, e sim em paralelo.

Ser multi-thread não tem nada a ver com multi-processamento, mas para que um sistema tenha suporte a multiprocessamento ele tem que ter tarefas bem divididas, para poder aloca-las a cada processador. É mais fácil alocar tarefas bem pequenas.

Sistemas como Windows NT não são "fortemente multi-thread" pois eles trabalham com threads, mas ainda assim são pesadas e poucas.

No BeOS, cada tarefa de uma aplicação é dividida em muito mais pedaços, sendo que muitos deles são do próprio sistema operacional.

2.3. Gerência de Memória

O MicroKernel do SO controla diretamente o acesso a memória. Devido ao multiprocessamento, o barramento de memória é o eixo de simetria entre os processadores. Assim, nenhum deles poderá acessar diretamente a uma memória ou a um dispositivo de entrada/saída. Além disso, a memória é partilhada nativamente, ao contrário dos clusters, que o emulam ao nível aplicativo.

Os drivers no BeOS são carregados em tempo real, indo para a memória apenas quando o dispositivo for utilizado, assim é economizado tempo e espaço em memória.

O Sistema Operacional também conta com um eficaz sistema de memória protegida. Um dos aspectos mais críticos para a estabilidade de um sistema operacional é a proteção de memória. Esse recurso controla o espaço de memória evitando as colisões. Como resultado, se alguma aplicação travar, o sistema e as outras aplicações continuam rodando perfeitamente.

2.4. Sistemas de I/O

Os drivers de I/O no BeOS são multithreaded, levando em conta a distribuição de tarefas de Entrada e saída através de processadores múltiplos, e resposta mais rápida em sistemas de processador únicos.

Os drivers são carregados dinamicamente como necessitado, sem requerer a reinicialização de sistema. Vamos supor que o usuário tinha uma placa de rede, mas que não funcionava no BeOS. Conseguiu o driver, ou o fabricante disponibilizou-o. O driver de rede será copiado para um diretório específico, é reiniciando este módulo do sistema operacional, é o suficiente para que este passe a funcionar.

O BeOS, envia informações de vídeo diretamente para a placa gráfica e então para o monitor. Assim deixa de existir sobrecarga no sistema e a taxa de quadros aumenta usando o mesmo hardware. O processo de enviar dados diretamente para o coração da placa de vídeo é chamado DMA, *Direct Memory Access*.

Os Kits de software fazem a comunicação entre os dispositivos de I/O e os servidores (Camada Intermediária do SO). Por exemplo:

Para uma entrada de teclado, o Input Kit fará a interface entre o programa e o servidor Input Server.

2.5. Sistema de Arquivos

O File System do Sistema Operacional foi desenvolvido para suportar altas taxas de transferência de dados. Isto inclui os drivers de acesso aos discos e um sistema de arquivos de 64 bits que suporta volumes muito grandes. O tamanho máximo do sistema de arquivos do BeOS é de 18.000 Petabytes. Isto é o mesmo que gerenciar 3 bilhões de HD de 6 GigaBytes.

O BeOS ainda traz um Gerenciador de Boot, e o seu carregador além de ser muito rápido é pequeno o suficiente para caber em um disco. Vamos dizer que o Windows foi instalado depois de instalar o BeOS; você não poderia mais iniciar o computador no BeOS porquê o Windows remove o gerenciador de Boot do Sistema. Mas o disquete de boot do Sistema é capaz de encontrar seu volume BeFS e carrega o Sistema Operacional, direto de seu HD.

O sistema de arquivos no BeOS funciona de maneira semelhante a um Banco de Dados, permitindo que os arquivos possuam um número infinito de atributos e que os mesmos sejam de qualquer tamanho, de forma que fique fácil fazer pesquisas complexas através do localizador (find) do sistema. Também é similar o fato de existirem índices no sistema de arquivos, o que aumenta ainda mais seu desempenho na busca de arquivos. Para evitar sobrecarga no sistema. A maioria dos Sistemas Operacionais, não atualiza pequenas transações para o disco imediatamente. Ao invés disso, mudanças são armazenadas em memória por um tempo, sendo enviadas para o HD quando o sistema está livre.

BeOS usa um sistema de jornal (Journaling) para manter todas as transações registradas. Uma vez que o jornal é atualizado em tempo real, se o sistema for desligado incorretamente ou travar, todos os dados são recuperados do jornal, resultando em um reboot tão rápido quanto um boot normal.

O seu Sistema de Arquivos reconhece e dá acesso a outras Partições de outros sistemas como Fat16/32 (Windows), ISO9660, HFS (Mac OS), NFS entre outros.

3. Finalidade do BeOS

Quando foi projetado, o BeOS teve sua utilização voltada para a multimídia. Assim, muitos cuidados foram tomados para que o sistema realmente suportasse aplicativos que trabalhasse com som, imagem e vídeo. Um sistema para trabalhar com Multimídia deveria ser um sistema operacional de tempo real. Para garantir um processamento em tempo real, o BeOs trabalha com multithreads em

um sistema microkernel. Permitindo que os drivers sejam carregados na memória apenas quando for necessário.

Diversos outros fatores foram analisados para o desenvolvimento deste SO. Porém, a característica que mais vale ressaltar é o fato dele ser novo e criado a partir do zero. Há 30 anos atrás, nunca se imaginou que um computador iria precisar de Gigabytes de dispositivos de armazenamento secundário e de memória. O problema atual é que a maioria dos sistemas existentes no mercado atual, foram desenvolvidos a mais de 10 anos ou funcionam em cima de sistemas antigos. Assim, a proposta do BeOS era ser um sistema novo para a realidade atual.

Apesar disso, a Be teve sua parte intelectual comprada pela Palm em agosto/2001, patrocinada pela Microsoft e pela Apple, e finalizou suas atividades em dez/2001. Devido a questões de mercado, a Palm decidiu não continuar seu desenvolvimento.

Atualmente, existem dois novos esforços para evitar a morte do BeOS: OpenBeOS e BlueEyedOS, sendo que o primeiro tenta reproduzir o BeOS com código aberto e o segundo quer usar o kernel do Linux e implementar compatibilidade com os programas do BeOS.

4. Referências

Dados Técnicos sobre o BeOS:

www.beatjapan.org/mirror/www.be.com/products/beos/beos_data_sheet.html

BeOS Specifications:

www.beatjapan.org/mirror/www.be.com/products/beos/beos_specifications.html

BeDepot.com – Your on line superstore for BeOS products:

www.bedepot.com

BeBits – Welcome: www.bebits.com

Grupo Brasileiro de Usuários de BeOS: www.bug-br.org.br

Sistemas operacionais Alternativos: www.thebox.org.br

Site sobre Hardware no BeOS: www.frizbe.net

Tanenbaum, Andrew S. (1992) “Sistemas Operacionais modernos”, LTC, Livros Técnicos e Científicos Editora S.A., 1992

Tanenbaum, Andrew S. (1988) Organização Estruturada de Computadores, LTC, Livros Técnicos e Científicos Editora S.A.