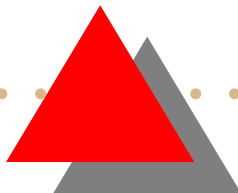


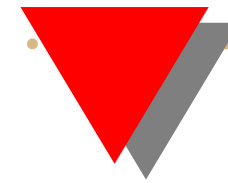
GoboLinux

Revolucionando a estrutura de diretórios do Linux

Hisham H. Muhammad - `hisham@gobolinux.org`

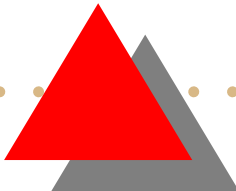
Andre Detsch - `detsch@gobolinux.org`

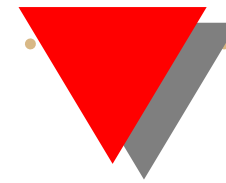




Introdução

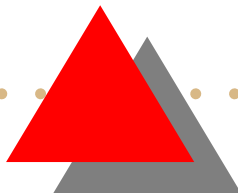
- Quando falamos de GoboLinux, estamos falando de duas coisas:
 - de uma distribuição Linux alternativa
 - de um novo modelo de hierarquia de diretórios para sistemas UNIX

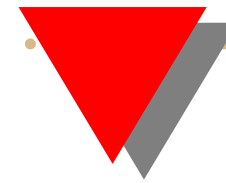




Sumário

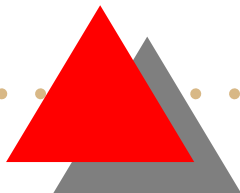
- Motivação
- Princípios do GoboLinux
 - Outras abordagens
 - Idéia geral da Estrutura
- Um *tour* pelo GoboLinux
- Distribuição GoboLinux
- Futuro do GoboLinux





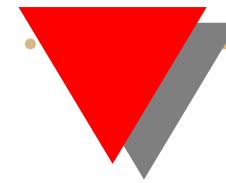
A *Árvore Unix*

- Contexto histórico:
 - Décadas de 60/70
 - Computação centralizada
- Diretórios servem para diferenciar:
 - Categorias de arquivos
 - Localização na rede



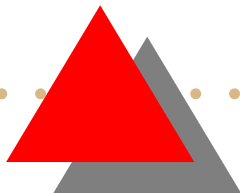
A Árvore Unix

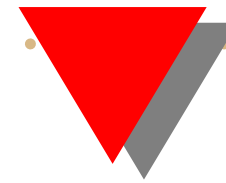
	Executáveis	Bibliotecas	Headers
Remoto	<code>/usr/bin</code> <code>/usr/sbin</code>	<code>/usr/lib</code>	<code>/usr/include</code>
Local	<code>/usr/local/bin</code> <code>/usr/local/sbin</code> <code>/bin</code> <code>/sbin</code>	<code>/usr/local/lib</code> <code>/lib</code>	<code>/usr/local/include</code>



A realidade atual

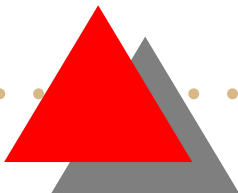
- Software livre, instalação freqüente de novas versões
- Gerenciadores de pacotes
 - Instalação de aplicativos a partir da compilação do código fonte leva a inconsistência na base de dados
- "Contorno":
 - `/usr` para programas instalados via pacotes
 - `/usr/local` para programas compilados





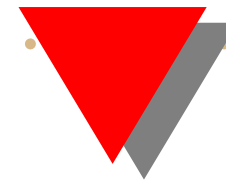
A realidade atual

- Continua o problema: como remover do `/usr/local`?
- `/opt`
 - Uma "brecha" no padrão Unix FHS
 - É o reconhecimento da existência do problema
- Será esta a filosofia ideal para o mundo desktop?



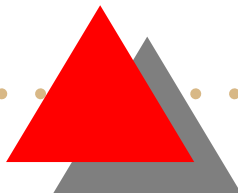


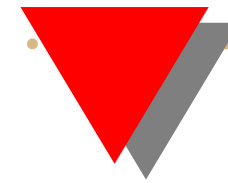
Princípios do GoboLinux



Objetivos

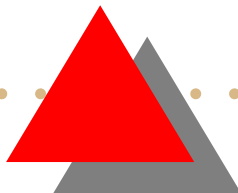
- Organizar a estrutura de diretórios
 - Compilação de programas
 - Troca de versões compiladas
 - Remoção de programas
- Manter compatibilidade total com a árvore UNIX

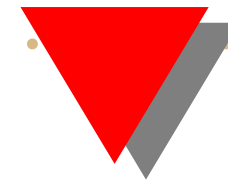




Mac OS X

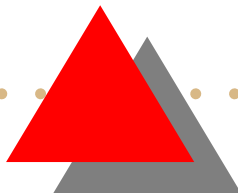
- Sucessor do Mac OS, da Apple
- Kernel Mach 3.0, ferramentas baseadas no FreeBSD
- Desafio: combinar a hierarquia UNIX à interface do Mac OS
- Solução encontrada: exibir na interface gráfica um subconjunto da árvore real

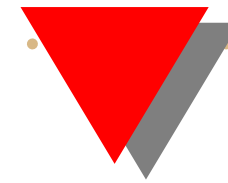




AtheOS

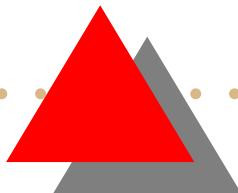
- SO livre voltado para PCs desktop
- Alto grau de compatibilidade UNIX
- `/usr` é utilizado como `/opt`
- Nas aplicações nativas, os caminhos são definidos em função do diretório `^` (diretório do executável)
 - Permite mover aplicações
 - Técnica incompatível com o UNIX

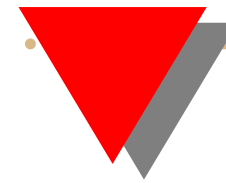




Depot, GNU Stow e Encap

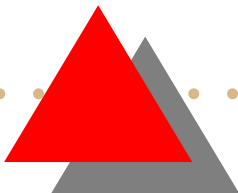
- Duas árvores de diretórios:
 - Árvore real, com um diretório por aplicação
 - Árvore de links simbólicos
- Combinação das árvores de cada aplicação
- Depot: mantém uma base de dados
- GNU Stow: make e make install com caminhos diferentes
- Encap: similar, mas com suporte a versões

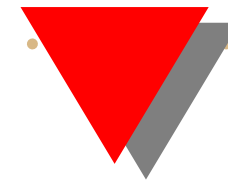




LinuxStep

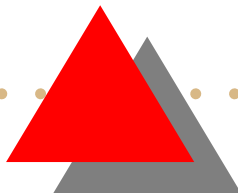
- Construção de um sistema operacional a partir do:
 - Kernel do Linux
 - GNUStep, implementação livre da API do sistema NeXTSTEP
- Alguns diretórios similares aos do GoboLinux
- Não há comprometimento com compatibilidade com hierarquia atual
- Projeto em estágios iniciais





Estrutura do GoboLinux

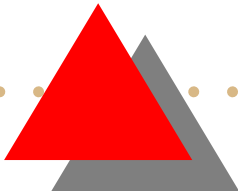
- Combina algumas das idéias anteriores em um modelo de fácil utilização
- Três níveis de hierarquias:
 - Real - /Programs
 - Links - /System/Links
 - Legacy - /usr, /bin, /lib, /sbin, /etc

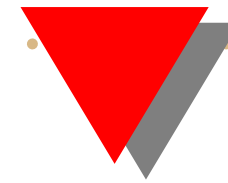




Hierarquia Real - /Programs

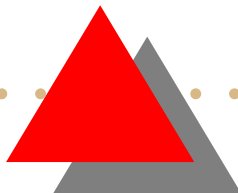
- Diretório que contém a grande maioria dos arquivos propriamente ditos
- Cada aplicação possui seu subdiretório
- Desta forma, não precisamos de gerenciadores de pacotes como RPM e apt-get

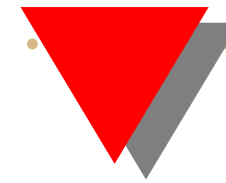




Hierarquia Links - /System/Links

- Centraliza, através de links, os arquivos da hierarquia real por categoria
- Os arquivos são divididos em:
 - Executáveis
 - Bibliotecas
 - Headers
 - Documentação





Hierarquia Legacy

- Série de links criados para satisfazer caminhos hardcoded existentes em aplicações.
 - `/bin`
 - `/sbin`
 - `/etc`
 - `/usr/lib`
 - `/usr/share, etc.`
- Pode ser “escondida” através de programa específico





Um *tour* pelo Gobolinux



/Programs

- Cada programa possui um diretório
 - `/Programs/XFree86`
- Cada versão possui um diretório
 - `/Programs/XFree86/4.3`
 - `/Programs/XFree86/4.3/bin`
 - `/Programs/XFree86/4.3/lib`
 - ...



/Programs

- A versão atualmente usada é indicada por um link
 - `/Programs/XFree86/Current` → 4.3
- As configurações são independentes de versão
 - `/Programs/XFree86/Settings`



/System

- `/System/Links`
 - `/Executables` → **executáveis** (`bin` e `sbin`)
 - `/Libraries` → **bibliotecas** (`lib`)
 - `/Headers` → **headers** (`include`)
 - `/Shared` → **arquivos compartilhados entre aplicações** (`share`)
 - `/Manuals` → **documentação** (`man` e `info`)



/System

- `/System/Settings`:
 - Contém arquivos globais de configuração e links para os arquivos de configuração das aplicações
 - Na hierarquia legacy, este diretório é conhecido como `/etc`
- `/System/Boot`:
 - Contém arquivos para o boot do sistema: kernel, `System.map` e configuração do bootloader.
 - Equivale ao `/boot`



/System

- `/System/Devices`
 - Contém os dispositivos do sistema
 - `/dev`
- `/System/Status`
 - Contém as informações sobre processos e estado do sistema
 - `/proc`



/Users, /Mount

- `/Users`

- Contém os diretórios dos usuários

- `/Users/gobo`

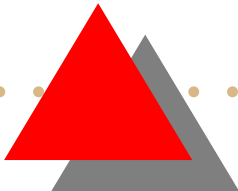
- Não há `/home`


- `/Mount`

- Contém os pontos de montagem de dispositivos

- `/Mount/Floppy`

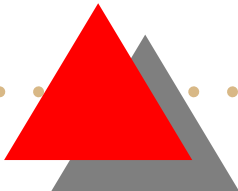
- Não há `/mnt`





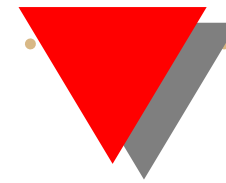
/Files

- Armazena arquivos compartilhados por diversas aplicações
- Obtidos, em geral, de forma “avulsa”, sem relação direta com alguma determinada aplicação
- Alguns exemplos:
 - `/Files/Fonts`
 - `/Files/Codecs`
 - `/Files/Plugins`



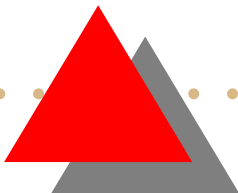


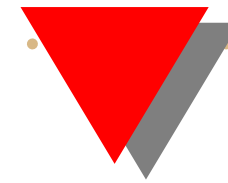
Distribuição GoboLinux



Histórico

- A distribuição existe há um ano
- Versão atual: 006Delta
- Estável
 - Utilizada como sistema único pelos desenvolvedores
- `gobolinux.org` desde outubro de 2002





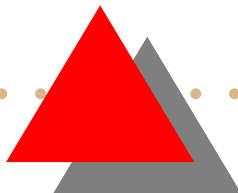
Estado do GoboLinux

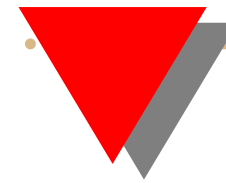
● Documentação

- Artigo apresentado no Fórum Software Livre 2002
- Man-pages dos scripts GoboLinux
- Livro do GoboLinux em desenvolvimento

● Página

- Vários textos online no site: FAQ, instalação...
- Mais de 400 pacotes disponíveis para download
- Lista de discussão: `gobo-1@gobolinux.org`

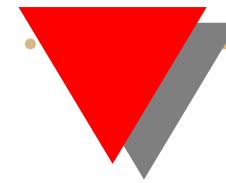




Compilação de Programas

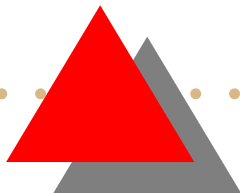
- Script `ins` - três passos básicos internos:
 - Preparação da compilação: roda `configure` se disponível, modifica `Makefile` se necessário
 - Compilação: execução do `make`
 - Instalação:
 - Execução do `make install` em ambiente restrito (sandbox)
 - Geração dos links

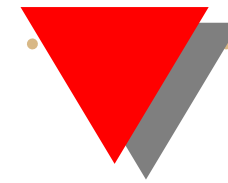




Sistema de Pacotes

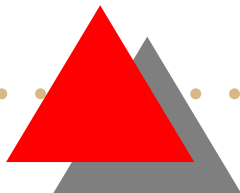
- Um pacote é simplesmente o diretório da aplicação, compactado.
- Por exemplo:
`KDE--3.1.1--i686.tar.bz2`
- Contém: `KDE/3.1.1/{bin,lib,...}`
- O script `package` gera um pacote

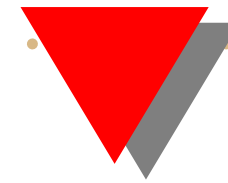




Instalação de pacotes

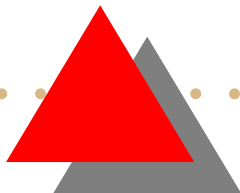
- `unpackage` instala um pacote
- O mesmo procedimento de geração de links usado na compilação de programas é usado pelo `unpackage` nos pacotes instalados.
- Diferentes versões de bibliotecas instaladas podem co-existir
- A migração do GoboLinux para o GCC 3 foi trivial

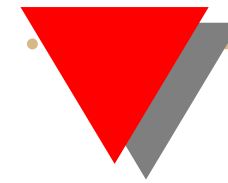




Instalando o GoboLinux

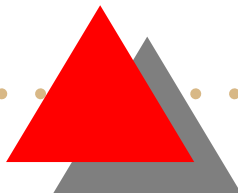
- CD-ROM bootável
 1. Criar uma partição
 2. Montar a partição nova
 3. Instalar o sistema base
 4. Configurações básicas
 5. Instalar o boot loader
 6. Reiniciar :-)

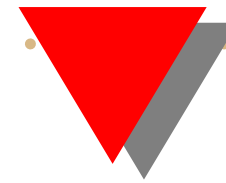




Sistema de boot

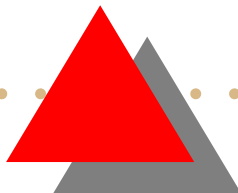
- Existem dois modelos tradicionais do UNIX:
 - System V : RedHat, Slackware
 - BSD: Debian, FreeBSD
- GoboLinux introduz um modelo próprio
- Scripts de boot localizados em
`/System/Settings/BootScripts`

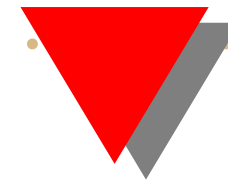




Sistema de boot

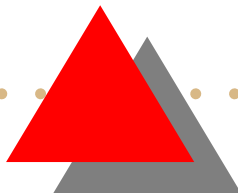
- Arquivos executados na inicialização e finalização: `Init` e `Done`
- Arquivos específicos para operação realizada: `Reboot`, `Shutdown`, `Single` e `Multi`
- Arquivo para customizações: `Options`
- Suporte a temas
 - Emular a aparência de outras distribuições
 - Tema próprio GoboLinux, versões texto e gráfica

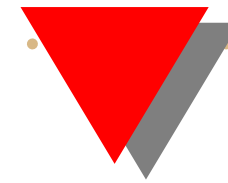




Metodologia de Scripts

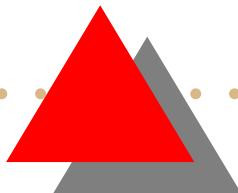
- Uma série de tarefas podem ser automatizadas através de scripts
- Pacote `/Programs/Scripts`
 - Provê um kit de funções básicas para geração de novos scripts
 - Geração de documentação dos scripts

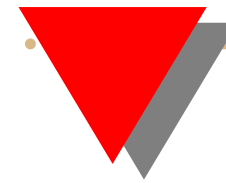




Exemplos de scripts

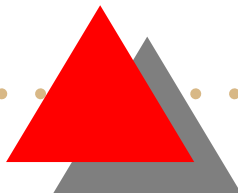
- Remoção de links quebrados
- Correção de atributos de arquivos
- Automação da compilação do kernel
- Front-ends para comandos UNIX
 - g: grep
 - f: find
 - l: ls
 - colormake: make...

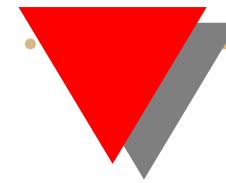




GoboHide

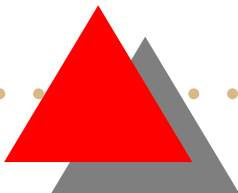
- Permite ocultar determinados arquivos da árvore de diretórios
- Arquivos permanecem acessíveis ao sistema
- Usado para ocultar a hierarquia legacy
- Implementado em duas partes:
 - Patch do kernel
 - Ferramenta user-space para gerenciamento

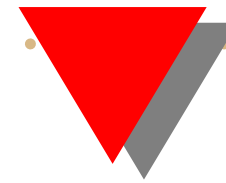




Rootless

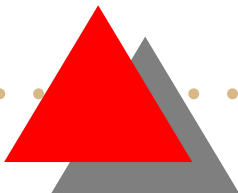
- Pacote que “emula” o GoboLinux dentro de um diretório qualquer
- Para ser utilizado por usuários não-root em outras distribuições
- Uma forma de organizar a instalação de programas dentro do `$HOME`
- Utiliza o pacote de Scripts padrão do GoboLinux





Futuro

- ChangedFiles: gerenciamento de links em kernel
- pacotes “source”: compilação durante instalação
- GoboLinux enquanto distribuição
- GoboLinux enquanto estrutura de diretórios
- Importância de distribuição não convencional para “melhorar” a flexibilidade do software livre





Demonstração

- Agora, um breve “passeio” pelos diretórios...

