



# **GoboLinux: uma nova proposta para a árvore de diretórios Unix**

**Hisham Muhammad**

**[gobolinux.org](http://gobolinux.org)**





# Resumo

- Introdução ao GoboLinux
  - Características
  - Uso / gerenciamento do sistema
- Evolução do CD
- Ferramentas de gerenciamento
  - Instalador
  - “Compile”, a ferramenta de compilação
  - Manager, Freshen
- Processo de desenvolvimento



# O que é o GoboLinux?

- Distribuição realmente alternativa
- Nova organização da estrutura de diretórios, adaptada aos padrões de uso atuais
- Alguns marcos
  - 2002, criação
  - 2003, kuro5hin, slashdot
  - 2004, Revista do Linux
  - 2005, cooperação com IBM
  - 2006, comunidade internacional de desenvolvedores



# Diretório raiz

~] `cd /`

/] `ls`

Programs

Users

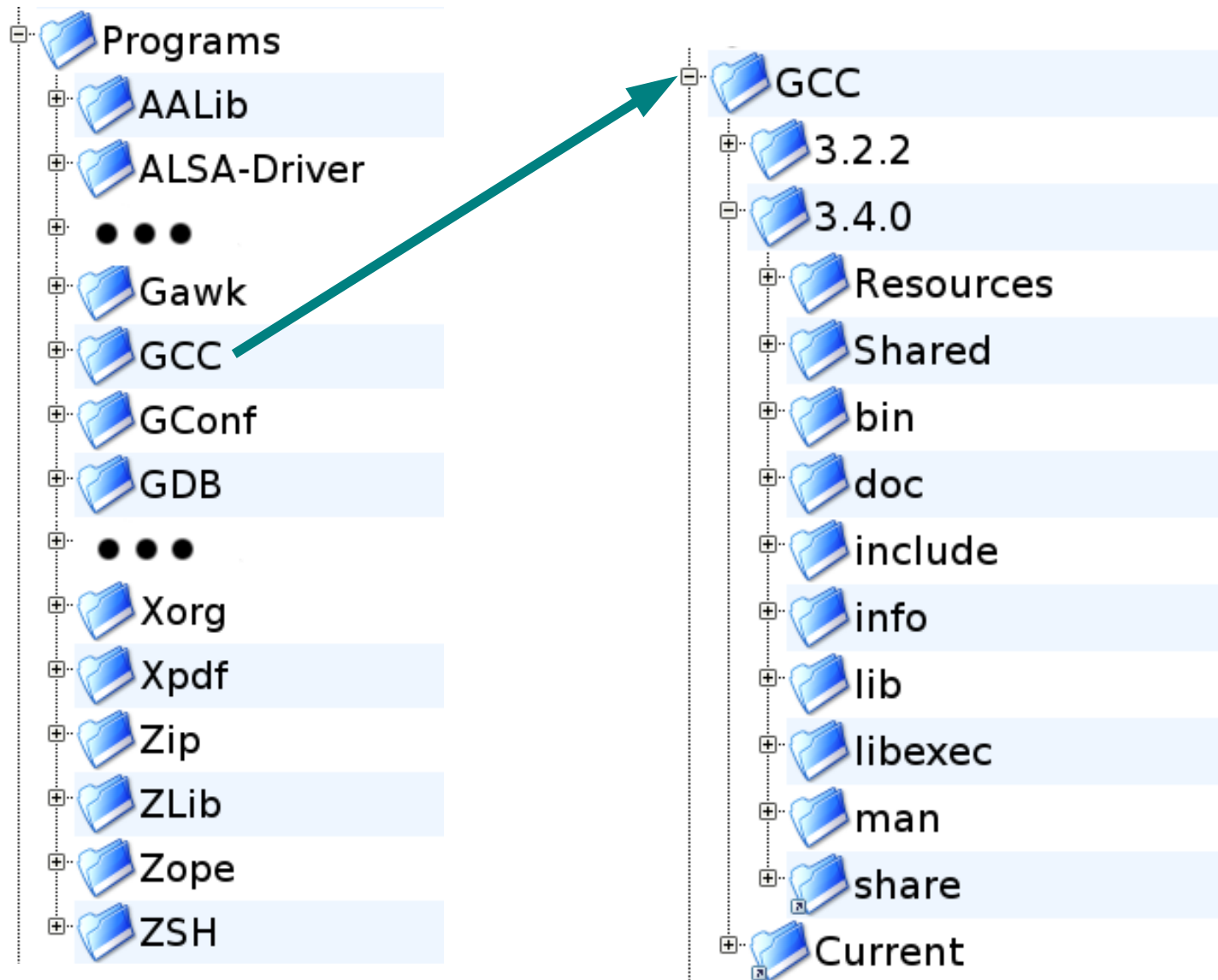
System

Files

Mount

Depot

# /Programs

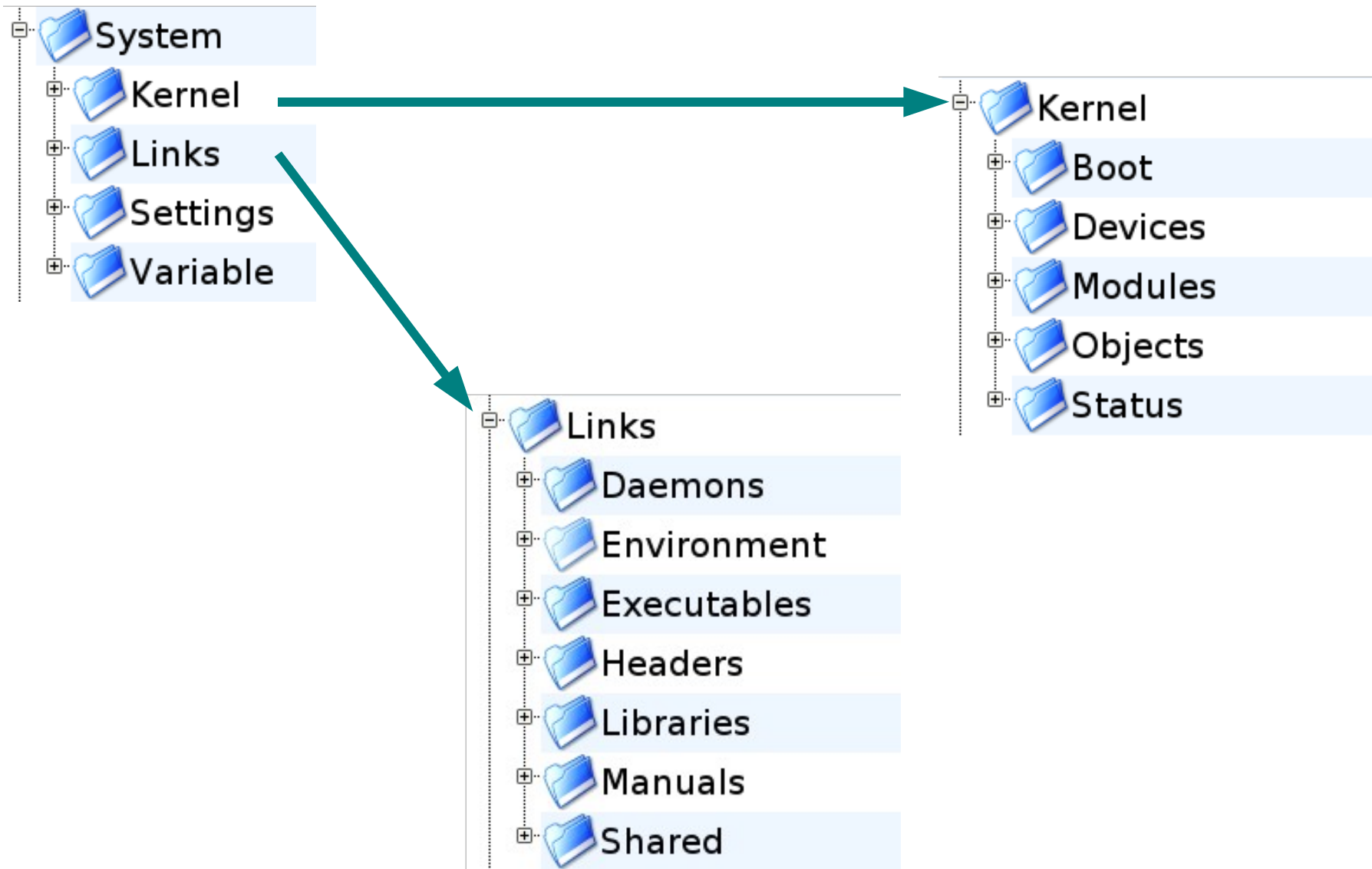


# Programas

```
/Programas] ls -l OpenOffice
total 8
drwxr-xr-x  9 root root 4096 01:07 1.1.4
drwxr-xr-x  3 root root 4096 04:36 2.0
lrwxrwxrwx  1 root root    5 04:36 Current -> 2.0
```

```
/Programas] ls -l GTK+
total 12
drwxr-xr-x 10 root root 4096 01:39 1.2.10
drwxr-xr-x  9 root root 4096 05:48 2.6.7
lrwxrwxrwx  1 root root    6 01:39 Current -> 2.6.7
drwxr-xr-x  4 root root 4096 01:39 Settings
```

# /System



# Como funciona

```
/System/Links/Libraries] ls -l | cut -b 49-  
...  
libgtk-1.2.so.0 -> /Programs/GTK+/1.2.10/lib/libgtk-  
1.2.so.0.9.1  
libgtk-1.2.so.0.9.1 -> /Programs/GTK+/1.2.10/lib/libgtk-  
1.2.so.0.9.1  
libgtk.a -> /Programs/GTK+/1.2.10/lib/libgtk.a  
libgtk.la -> /Programs/GTK+/1.2.10/lib/libgtk.la  
libgtk.so -> /Programs/GTK+/1.2.10/lib/libgtk-1.2.so.0.9.1  
libgtk-x11-2.0.la -> /Programs/GTK+/2.6.7/lib/libgtk-x11-  
2.0.la  
libgtk-x11-2.0.so -> /Programs/GTK+/2.6.7/lib/libgtk-x11-  
2.0.so.0.600.7  
libgtk-x11-2.0.so.0 -> /Programs/GTK+/2.6.7/lib/libgtk-x11-  
2.0.so.0.600.7  
libhfs.a -> /Programs/CDRTools/2.01/lib/libhfs.a  
libhistory.a -> /Programs/Readline/5.0/lib/libhistory.a  
libhistory.so -> /Programs/Readline/5.0/lib/libhistory.so.5.0  
...
```



# Estrutura de diretórios

- Links para a hierarquia “legacy”
  - /bin -> /System/Links/Executables
  - /etc -> /System/Settings
  - /usr/sbin -> /System/Links/Executables
  - ...
- Caminhos hardcoded típicos funcionam (/bin/sh, /usr/bin/python, /lib/modules...)



# Características do sistema

- Links de compatibilidade (/bin, /usr, /etc...), que podem ser ocultados -> GoboHide
- Pacote binário é um “tar.bz2”
- Remoção é um “rm -rf”
- Manutenção garantida: independência de pacotes, ou gerenciador de pacotes
- Scripts automatizam tarefas, mas são opcionais
- Sistema que pode ser totalmente gerenciado com as ferramentas Unix clássicas



# Características do sistema

- Diretório Resources do pacote armazena informações relativas a ele
  - Description
  - Dependencies, BuildDependencies
  - Environment: variáveis de ambiente
- Resources/Defaults/Settings
  - Permite restaurar configurações
- Resources/Unmanaged
  - Arquivos relevantes somente para a instalação local do pacote (/var, etc.)



# Evolução do CD

- **001-004:** Um release por novo usuário :)
  - Arquivão .tar.gz com sistema funcional
- **005-006:** 2003: Primeiras versões oficiais
  - CD bootável
  - Ainda com o arquivo .tar.gz, mas com vários pacotes individuais
  - Processo de instalação manual é documentado
- **007:** Primeiro LiveCD de fato
  - Módulo Cloop (depois substituído por SquashFS)
  - CD se torna um sistema completo



# Evolução do CD

- **010:** Revista do Linux
  - Detecção de hardware
  - Instalador gráfico / modo texto
  - Configurações customizadas para o usuário
  - Fase de testes minuciosa
- **011:** FISL 2004
  - Distribuição de CDs
  - Integração de ferramenta de compilação automatizada



# Evolução do CD

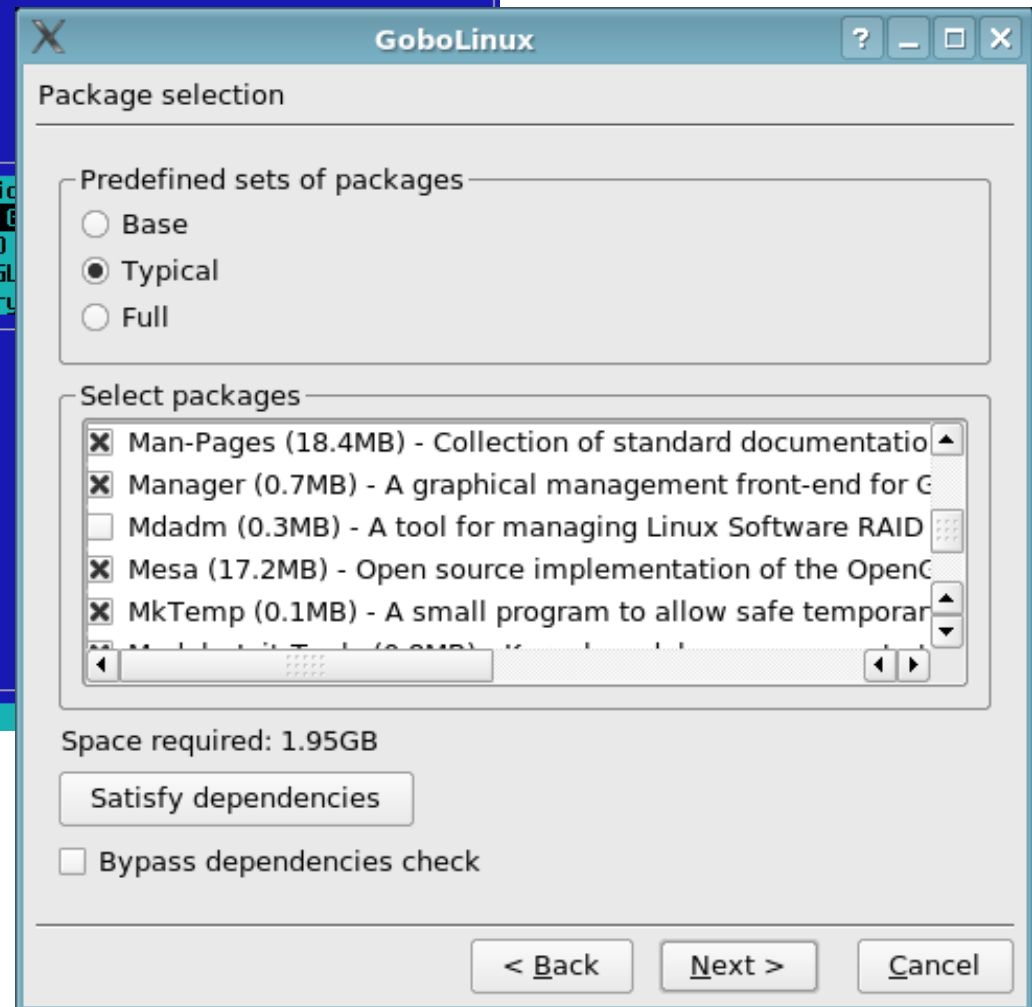
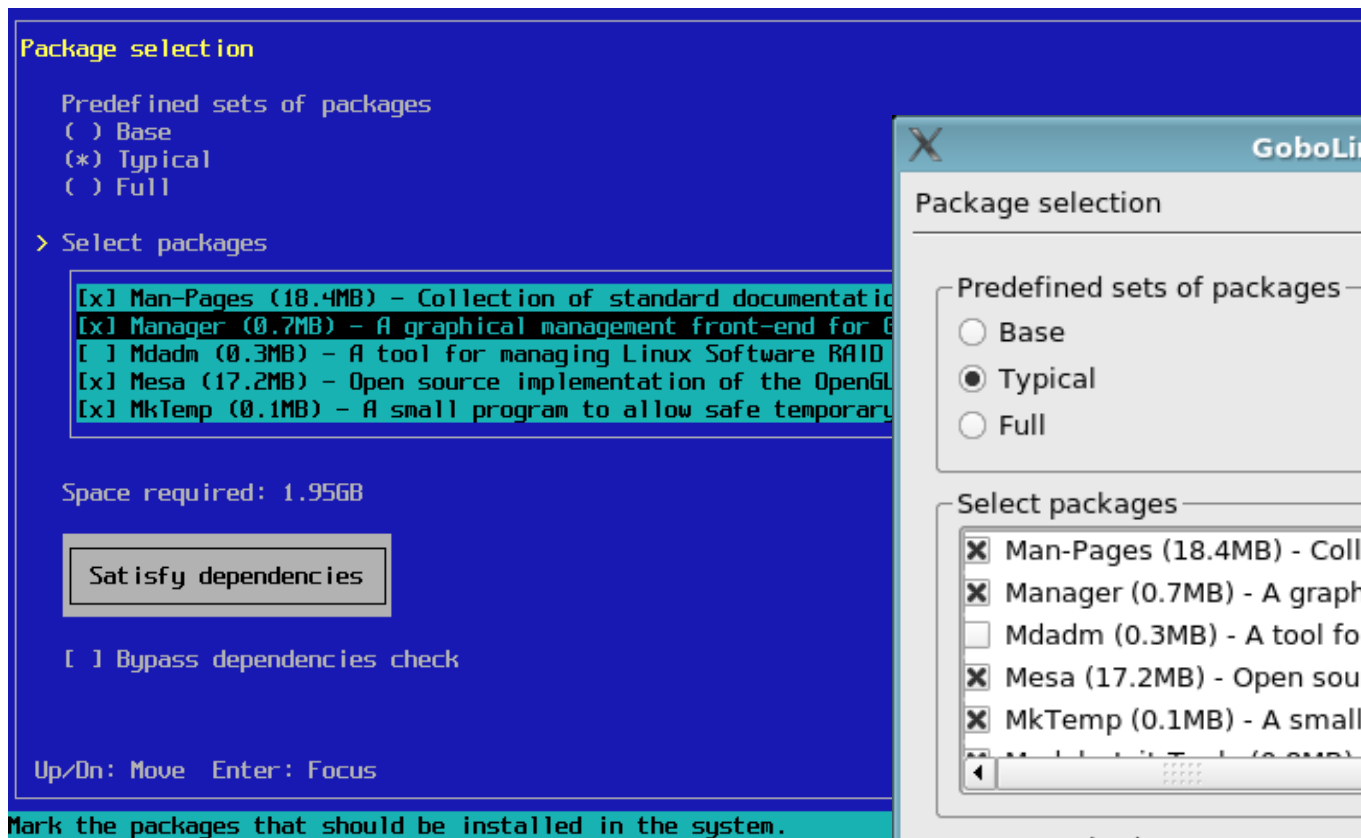
- **012:** 2005
  - Ferramenta gráfica de gerenciamento do sistema
  - Inglês, português, alemão
- **013:** 2006
  - Desenvolvimento mais aberto
  - Evoluções tecnológicas (UnionFS no CD, Listener)
- **014:** Em desenvolvimento
  - Snapshots lançados freqüentemente
  - Inglês, português, alemão, húngaro
  - Disponível até o final do ano



# Instalador

- Como desenvolver?
  - Outros instaladores são demasiadamente complexos: criar um próprio.
  - Toolkit?
    - Dialog: muito restrito
    - Bindings: muito específico
- Criamos o AbsTK
  - Codificação independente de toolkit
  - Implementações Qt e Curses
- Permite formatar partição, adicionar usuários, escolher pacotes, etc.

# Instalador







# Compile - Motivação

- Ferramenta para compilação automatizada
- Iniciativa inicial: *Source packages*
  - Descrevia passo-a-passo a compilação
  - Manutenção difícil
- Possibilidade: adaptar o portage (Gentoo )
  - Arquivos ebuild são complexos demais
- Em geral pacotes podem ser compilados com um comando... (CompileProgram)
- Adição de pequenas alterações (opções para o configure, patches, etc.) deve ser simples



# Compile - Funcionamento

- Baseado na idéia de recipes
  - Arquivos que especificam “o que” deve ser feito
  - No caso mais simples (e bastante comum), podem conter apenas 2 linhas
  - Buscadas no servidor ou armazenadas localmente
- Arquivos do diretório Resources acompanham a recipe
- Suporte a meta-recipes
- Geração semi-automática de recipes
  - MakeRecipe <http://www.foo.org/foo-1.0.tar.gz>

# Exemplos de recipes

## Gimp

```
url=ftp://ftp.gimp.org/.../gimp-2.0.1.tar.bz2  
recipe_type=configure
```

## GTK+

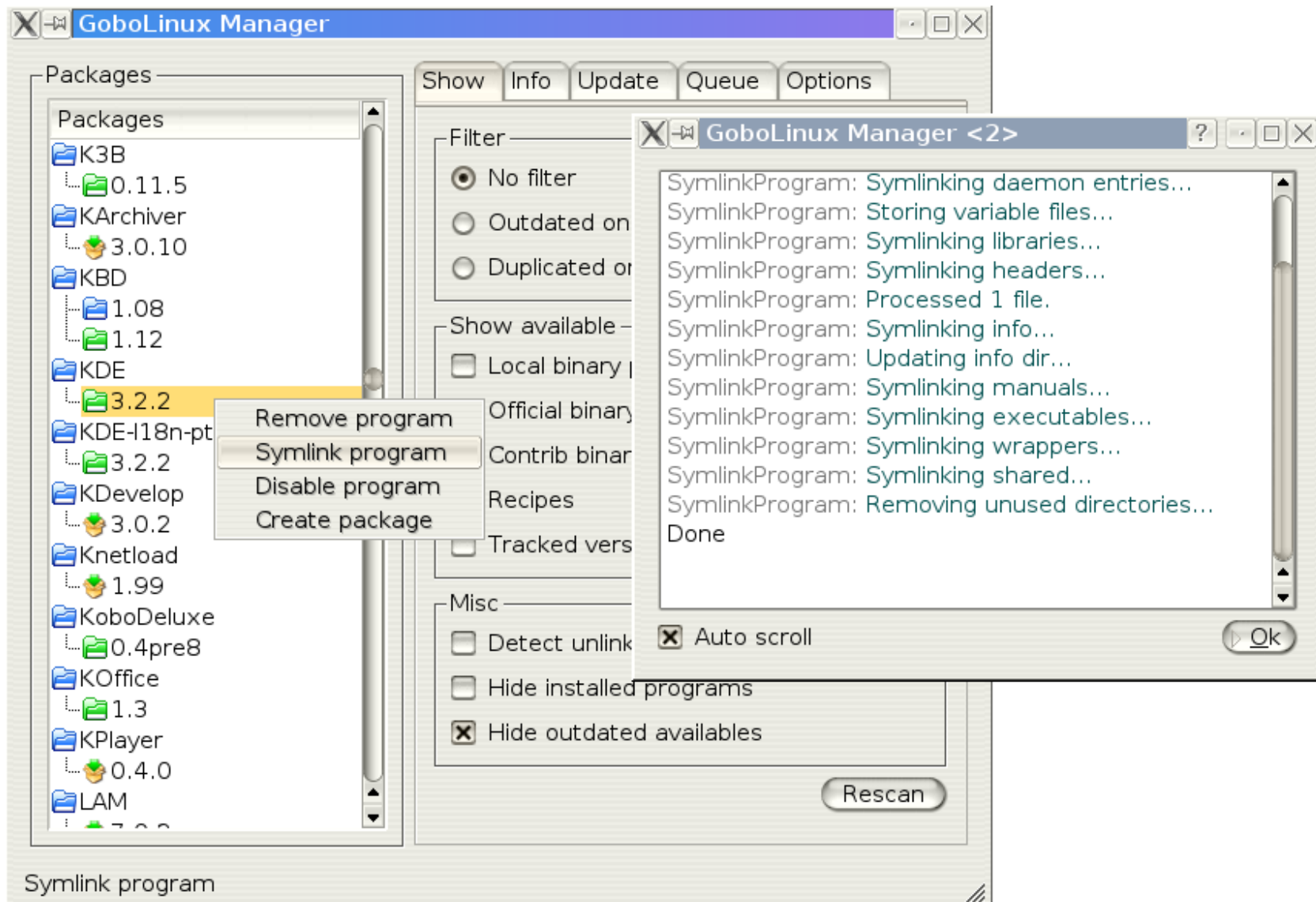
```
url=http://ftp.gnome.org/.../gtk+-2.3.0.tar.bz2  
recipe_type=configure  
configure_options=(  
    "--with-x"  
    "--with-libpng"  
    "--with-libjpeg"  
    "--with-libtiff"  
)
```



# Ferramentas de gerenciamento

- Freshen
  - Identifica programas desatualizados em relação às recipes disponíveis
  - Desenvolvido por Michael Homer (NZ)
- Manager
  - Gerenciador gráfico de pacotes
  - Front-end para scripts de gerência e Compile
  - Identifica pacotes binários, locais, recipes e versões mais atuais sem recipe ou pacote binário
  - Permite a execução em lote de tarefas

# Manager





# Processo de desenvolvimento

- Inicialmente grupo de amigos no RS
  - Comunidade se expandiu com o tempo
- Algumas tarefas fixas são distribuídas entre desenvolvedores
  - Kernel, geração do CD da distribuição...
- Desenvolvimento das ferramentas
  - Grupo original + novos devs espalhados pelo mundo
- Manutenção dos pacotes
  - Comunidade colabora, modelo bastante aberto



# Como contribuir

- Utilizando o sistema
  - Reports de instalação / uso
  - Bugs, sugestões de melhorias
- Auxiliando na documentação
- Contribuindo com recipes
- Auxiliando no teste de versões pré-release
  - Snapshots GoboLinux 014 disponíveis no site!



# Considerações Finais

- Uso da estrutura de diretórios é viável e traz benefícios significativos
- Desenvolvimento e amadurecimento de ferramentas que fazem uso destas vantagens
- Usuário não fica dependente dos desenvolvedores da distribuição
- Terreno fértil para novas idéias...





**Perguntas?**

**<http://www.gobolinux.org>**

**[hisham@gobolinux.org](mailto:hisham@gobolinux.org)**

