

Conhecendo um pouco do GNOME por dentro!

Lincoln de Sousa

January 15, 2009



Concepção

Em meados de 1996 e 1997, era notável a ascensão de ambientes desktop, porém todos proprietários. O próprio KDE, apesar de ser licenciado sob GNU/GPL, usava a biblioteca Qt, até então também proprietária.

Esse foi o momento que dois hackers mexicanos se uniram, para conceber o GNOME. (Miguel de Icaza e Federico Mena)



Bases iniciais

Depois da idéia lançada, os desenvolvedores precisavam escolher alguma tecnologia que, além de poderosa, fosse livre para servir de base para o GNOME.

Para o trabalho mais complexo, como desenhar janelas, botões e controles em geral, o time gnomico escolheu uma biblioteca desenvolvida para suportar um programa de edição de imagens, **O GIMP!**





A GTK

Depois de ter alguns problemas com a utilização do *Motif*, Peter Mattis (criador do GIMP), decidiu escrever a sua própria *Tool kit*, que foi batizada de **GIMP Tool Kit**, ou como costumamos chamar carinhosamente, a **GTK**.

Parte da fundação da GTK é constituída por uma sub-biblioteca chamada GDK, que tem como função desenhar objetos primários, como pontos, retas, quadrados e etc. Ambas foram escritas em C.





Evolução da GTK

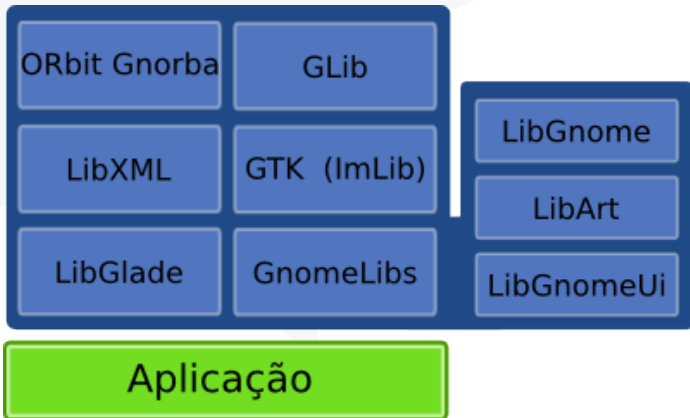
Até a versão 1.2, o sistema de orientação a objetos se manteve embutido. Depois disso, tanto a parte que continha funções utilitárias e estruturas de dados diversas quanto a parte que continha o sistema de orientação a objetos, foram separados em bibliotecas diferentes, a **GLib** e a **GObject**.

A partir da série 2.x, a GTK ganhou suporte a multihead, suporte a Pango para as fontes, atk para acessibilidade, ficou bonita =) e etc.



Conhecendo um pouco do GNOME por dentro!

Layout das bibliotecas na série 1.x





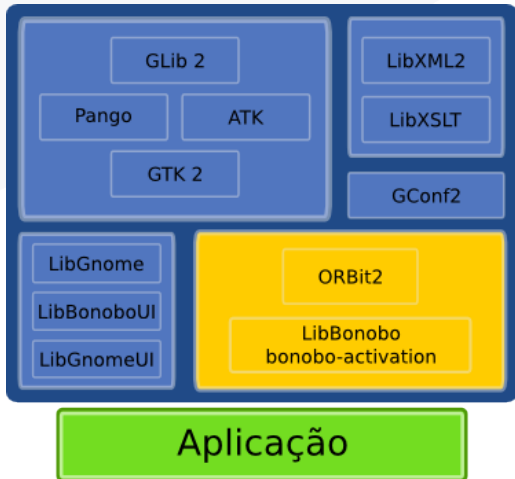
O que mudou na série 1.4





Conhecendo um pouco do GNOME por dentro!

A situação da versão 2.x (atual)





GLib

É a camada mais baixa na pilha de bibliotecas Gnomicas. Contém a implementação do sistema de tipos GObject, um conjunto de estruturas de dados, wrappers de portabilidade, implementação de *MainLoop*, uma boa quantidade de funções de manipulação de strings, de arquivos, de gerência de memória e etc.



GLib

além de todas as estruturas de dados, como listas, *hash tables*, arrays e etc, a GLib também contém várias sub-bibliotecas com as mais diversas funções:

- ▶ GObject
- ▶ GModule
- ▶ GThread
- ▶ GIO



GLib::GObject

É o sistema de objetos da GLib. E implementa features como:

- ▶ Definição dinâmica de tipos
- ▶ Herança
- ▶ Interfaces
- ▶ Introspecção
- ▶ Garbage collection de objetos
- ▶ (etc)



GLib::GIO

É a nova camada de IO da GLib. Ela foi criada para prover uma API simples e de alto nível para *VFS*, além disso, veio para ocupar o lugar certo na pilha de bibliotecas.

Um dos fortes motivos para a sua criação foi a necessidade de substituir o GnomeVFS.



GConf

Trata-se de um servidor de configurações. Tem papel fundamental de centralizar a configuração de programas gnomicos e provê uma API extremamente simples para acesso aos dados.



GStreamer

É um framework de streaming de media, baseado em plugins que é usado pelo GNOME. Escrito em C, o GStreamer também utiliza a GLib como biblioteca de base e seu sistema de plugins é baseado na **GObject**.



Bindings

Todas as bibliotecas já citadas até agora foram desenvolvidas em C, porém, um dos grandes objetivos da GLib e sua turma é prover um caminho tranquilo para que possam ser escritas bindings para outras linguagens.



Bindings oficiais

- ▶ gtkmm
- ▶ pygtk
- ▶ Gtk#
- ▶ gtkperl
- ▶ JavaGnome





Exemplinoxx XD

- ▶ GLib+GObject
- ▶ Gtk
- ▶ PyGtk
- ▶ PyGst
- ▶ Gtk-Ruby
- ▶ Algumas aplicações mais gnomicas =)



Ambiente de desenvolvimento

- ▶ Emacs, vim, Anjuta
- ▶ Devhelp
- ▶ Glade



Quick tips

- ▶ Preocupação com a interface (usabilidade)
- ▶ Acessibilidade (Orca!)
- ▶ GNOME HIG!
- ▶ Vala!



Passado, presente e futuro

- ▶ A mania do GNOME de seguir padrões
- ▶ Evolução da biblioteca
- ▶ DBus (o kov já falou =)



Kbô

Lincoln de Sousa - lincoln@minaslivre.org
<xmpp:lincoln@jabber-br.org>