Desenvolvimento de Aplicações Gráficas para o X Window System

Alfredo Kengi Kojima
Paulo César Pereira de Andrade
Carlos Augusto Moreira dos Santos

Mini-curso apresentado no Software Livre 2000
Porto Alegre, 5 de maio de 2000

Este documento foi redigido usando apenas software livre.

– Typeset by FoilTEX –

Apresentação

Objetivo deste mini-curso: Introdução à arquitetura do *X Window System* e às técnicas de programação.

Estrutura

1. Descrição dos princípios do protocolo X (ferramenta de suporte à criação de aplicações concorrentes e distribuídas)

2. Apresentação da API *Xlib* e técnicas básicas para autoria de aplicações.

3. Exemplos de diversos toolkits de interface.

4. Bibliografia recomendada e endereços na Internet com mais informações sobre o assunto.

Histórico

1984. Scheifler e Gettys criam o X Window System no MIT.

1988. Criado o MIT X Consortium para prover suporte ao desenvolvimento.

1993. X Consortium, Inc. torna-se organização independente e autônoma.

1996. (31 de dezembro) o X Consortium deixa de existir e os os direitos passam à Open Software Foundation (OSF).

1997. OSF une-se ao X/Open, criando "The Open Group" (TOG).

O quê é o X Window

- O X Window System é uma especificação formal de software. Isso o diferencia de outros projetos de software livre.

- O Open Group distribui uma *implementação de exemplo* (*sample implementation* - SI).

Em caso de discrepância entre a implementatção e a especificação, prevalece a especificação.

Arquitetura do X Window System

Possui arquitetura cliente-servidor

- O servidor localiza-se na estação de trabalho do usuário, oferecendo "serviços" de acesso a dispositivos de entrada e saída.

- Os clientes podem rodar local ou remotamente e conectam-se ao servidor através de uma rede.

Serviços oferecidos pelo servidor

- Acesso transparente ao hardware gráfico.

- Mecanismos de comunicação entre clientes

- Entrada de dados por dispositivos como mouse e teclado.

Modelo do sistema de janelas

Sistema de janelas (segundo Gettys e Scheifler):

Interface de programação. Biblioteca em uma linguagem de programação.

De baixo nível. Provê funções gráficas primitivas.

De alto nível. Provê funções para apresentação de menus e outros elementos constituintes da interface. X não define uma interface de alto nível.

Interface de aplicação. Define a interação *mecânica* entre aplicação e usuário.

Interface de gerenciamento. Define a política de controle do posicionamento, tamanho e sobreposição das janelas sobre a área de trabalho. X não provê uma interface de gerenciamento padrão.

Interface com o usuário

- Gettys e Scheifler chamam de *interface com o usuário* a soma das interfaces de aplicação e de gerenciamento. Essa classificação define uma arquitetura em camadas e sugere independência entre *macanismo* e *política*.

- X fornece mecanismos para que diferentes políticas de interface sejam implementadas.

- X não possui uma interface com usuário padrão. Fica a cargo da aplicação definir sua interface. O uso de toolkits permite que as aplicações sejam consistentes entre si.

Estrutura

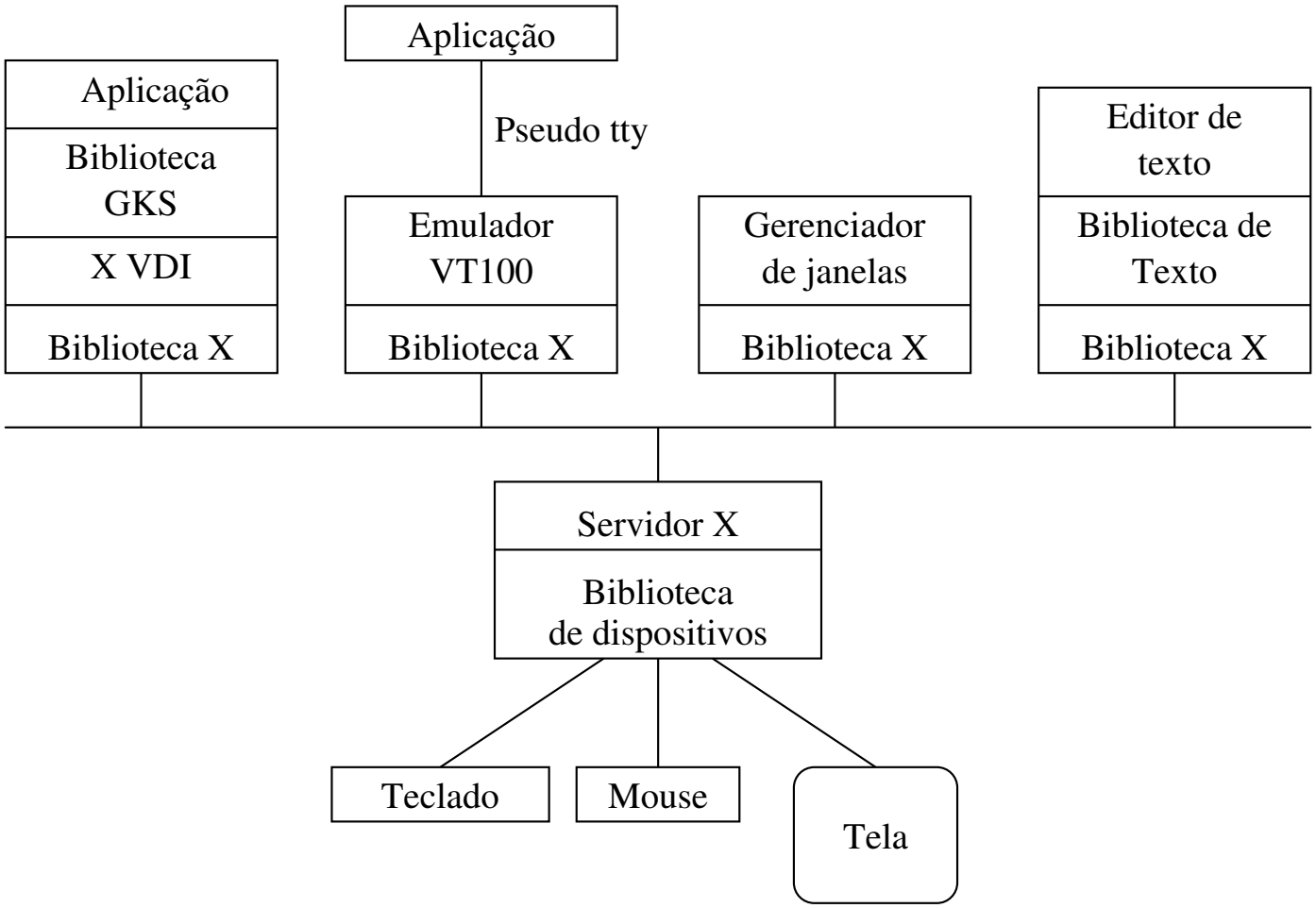

Figura 1: Estrutura de um sistema X

Conceitos importantes

Tela (screen). X possui suporte a múltiplos monitores ligados a uma mesma máquina ao mesmo tempo. Cada monitor ligado é denominado *screen*.

Display. Conjunto de hardware de vídeo, teclado e mouse mais o servidor que gerencia esses dispositivos. Múltiplas telas podem conviver, formando um display. Uma máquina dedicada que serve apenas como display, é chamade de *terminal X*.

Janela (window). Tem dois sentidos, dependendo do contexto:

Baixo nível. Área retangular em uma das telas na qual um ou mais clientes podem realizar operações de entrada e saída (desenhos, etc.). A hierarquia de windows permite ao cliente estabelecer políticas de tratamento de eventos.

Alto nível. Janela de aplicação (*toplevel window*, ou *toplevel shell*).

Área de trabalho. Janela raiz de uma tela e as janelas criadas sobre ela. O gerenciamento da área de trabalho cabe ao window manager.

Widget. Um elemento de interface gráfica (botões, *labels*, menus, etc.).

Servidor. Programa que compartilha um ou mais Displays com os clientes por meio do protocolo X.

Cliente. Aplicação que usa um ou mais displays, comunicando-se com o(s) servidor(es) por meido do protocolo X.

Gerente de Display (*display manager*). Aplicação responsável por fornecer autenticação de acesso a um ou mais displays. O gerente de display padrão é o programa XDM.

Comunicação entre clientes

- Clientes podem se comunicar uns com os outros usando o servidor como intermediário. Eles fazem isso enviando eventos às janelas dos outros clientes e alterando as *propriedades* existentes nelas.

- Eventos permitem envio de mensagens, mas não são adequados ao compartilhamento de dados entre clientes.

Atoms, properties, cut buffers e selections

Átomos (atoms). Nomes que os clientes podem usar para comunicar informação uns aos outros. Cada átomo *internado* no servidor recebe um rótulo único de 32 bits.

Propriedades (properties). Blocos de dados que um cliente "pendura" em uma janela, contendo quantidades arbitrárias de informação. Propriedades são mantidas na memória do servidor.

Áreas de recorte (cut buffers). Oito ropriedades existente na janela raiz, identificadas pelos átomos XA_CUTBUFFER0 a XA_CUTBUFFER7. Usadas para implementar um sistema simples de recortar-e-colar.

Seleções (selections). Mecanismo de compartilhamento de dados que funciona em sistema de revezamento. Apenas um cliente pode deter a seleção de cada vez. O detentor da seleção deve atender às requisições dos outros clientes convertendo os dados para o formato solicitado, armazenando o resultado em uma propriedade e notificando o solicitande da disponibilidade.

Arrastar e soltar (drag-and-drop)

- Técnica interativa baseada na metáfora de selecionar um objeto em uma janela e transportá-lo para outra.
- X não provêm DnD explicitamente.
- Há muitos protocolos de drag-and-drop incompatíveis entre si.

A falta de um padrão real é, provavelmente, o motivo principal da relativa escassez de aplicações para X Window que suportem drag-and-drop.

Protocolos de DnD

Motif. DnD baseado em mensagens entre clientes, properties e selections. Usados pelo Netscape Navigator e programas integrantes do CDE.

Offix. Pacote desenvolvido por César Crusius, incluindo um protocolo DnD. Distribuído livremente e usado por um grande número de aplicações (SIAG, XEmacs, etc.).

Xdnd. Um protocolo recente que está se tornando padrão de fato entre os pacotes de software livre. Incluído nos *toolkits* GTK, Qt, JX e LessTif.

Gerenciamento de janelas

- Implementa as políticas de atribuição do foco, redimensionamento e sobreposição de janelas.
- As convenções foram estabelecidas pelo ICCCM (Inter-Client Conventions Manual).
- O uso de um WM não é obrigatório, mas é muito útil.

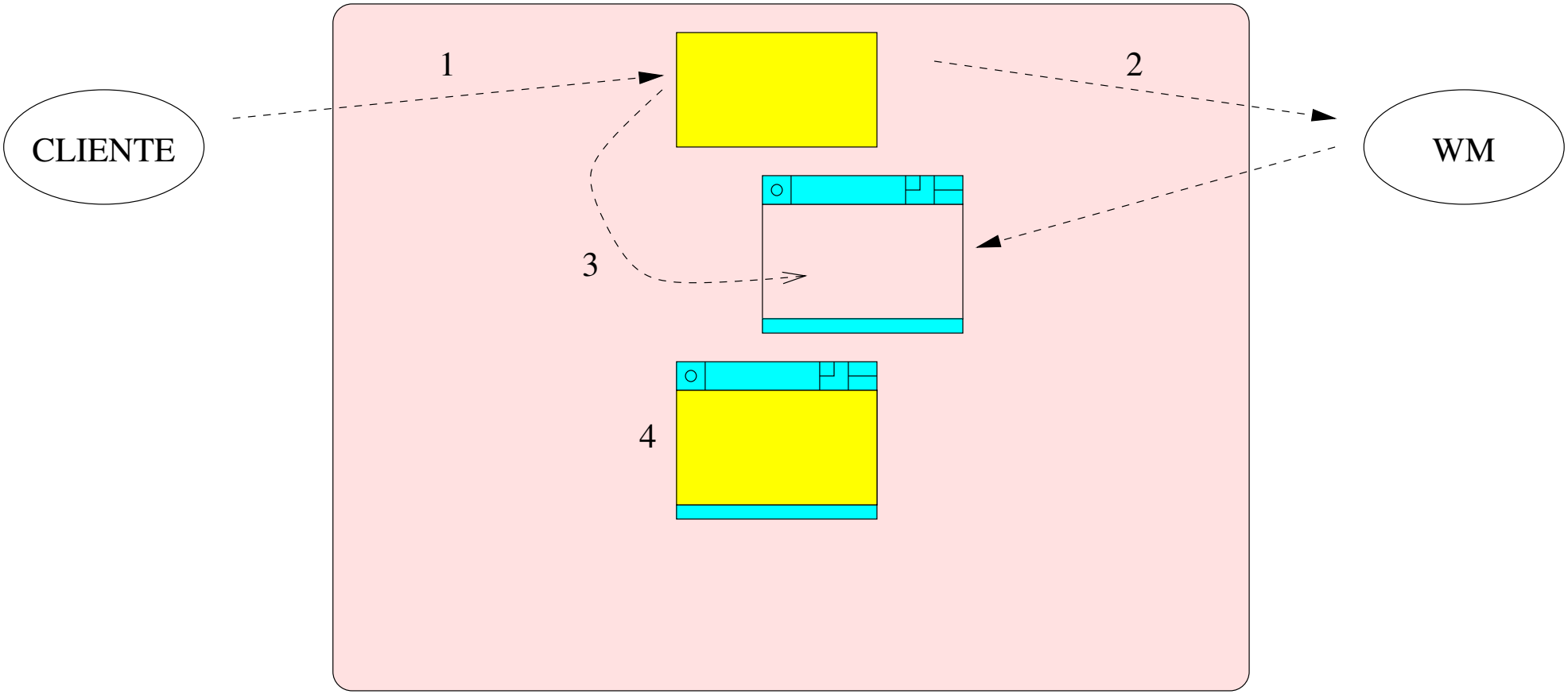

Figura 2: Janela de um cliente dentro do contêiner criado pelo window manager.

Desenvolvimento de aplicações

Toolkits/Bibliotecas

- Implementam interfaces de programação para X.
- O X Consortium nunca impôs um padrão para interfaces com o usuário, o que resultou na grande proliferação de toolkits diferentes.

Toolkits — Xlib

- Biblioteca em linguagem C, que faz a comunicação com o servidor X e oferece funções com correspondência direta, ou quase, com o protocolo X.
- É a API de mais baixo nível disponível para se programar aplicações para X.
- Definida como padrão pelo X Consortium.
- Possui funções para se abrir uma conexão com um display, criação, destruição e manipulação de janelas, desenho, recebimento de eventos etc.
- Todos toolkits de mais alto nível usam Xlib.
- Pode ser utilizada para escrever aplicações simples mas não é aconselhável para aplicações complexas.
- Programas que realizam operações "de baixo nível" têm que usar as funções da Xlib.

Figura 3: Estrutura de uma aplicação baseada em Xlib.

Uma abordagem bastante comum em aplicações complexas é usar um toolkit para construir a interface e funções da Xlib para operações de baixo nível. Neste caso deve-se tomar cuidado para que as chamadas feitas à Xlib não interfiram com a política implementada pelo toolkit.

Toolkits — X Toolkit (Xt)

- Provê recursos básicos para criação de toolkits gráficos. Não oferece widgets completos.
- Implementa um *modelo de objetos* por meio de convenções e técnicas de programação. Exige que o programador domine a linguagem de programação e disciplina de trabalho rigorosa.
- Pode ser usado com outras linguagens além de C e C++.
- Xt e os toolkits derivados são flexíveis e possuem recursos únicos dentre os demais.
- Funçoes para
  - inicialização do toolkit e da aplicação;
  - criação, manipulação e destruição de widgets;
  - adição e remoção de *callbacks* e *actions* (resposta a eventos) a widgets;
  - gerenciamento de Xt Resources;
  - tratamento de eventos, *timeouts* e sinais.

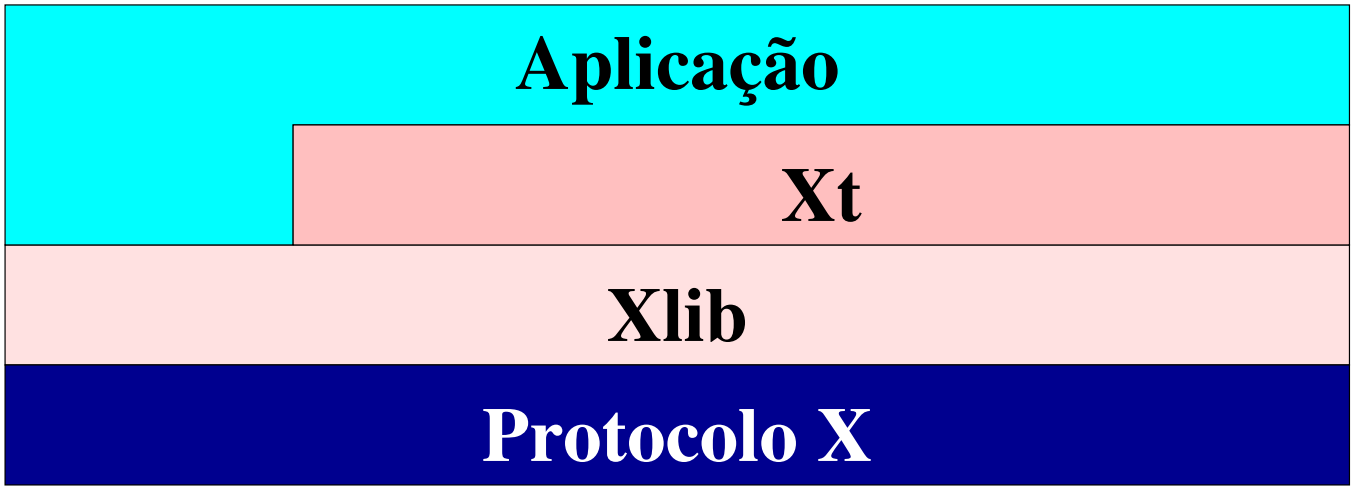

Figura 4: Estrutura de uma aplicação baseada em Xt.

Não é comum criar aplicações que usem apenas Xt, porque é muito trabalhoso. Normalmente usa-se Xt associado a um toolkit de alto nível.

Toolkits — X Athena Widget Set (Xaw)

- Criado como um exemplo de uso do Xt (nunca foi um toolkit completo, para uso profissional). Nunca foi adotado como padrão pelo X Consortium.
- Tornou-se muito usado porque era distribuído junto com a implementação do X.
- Carece de muitos recursos encontrados em outros toolkits.
- As principais defiências são
  - não implementar uma política de *foco de entrada*;
  - não possui *keyboard traversal*;
  - os widgets não se diferenciam muito visualmente.
- Surgiram diversas variantes de Xaw: Xaw3D, Xaw95, neXtaw, XawM, etc.. Algumas aplicações também tentam resolver os problemas por meio de truques de programação.

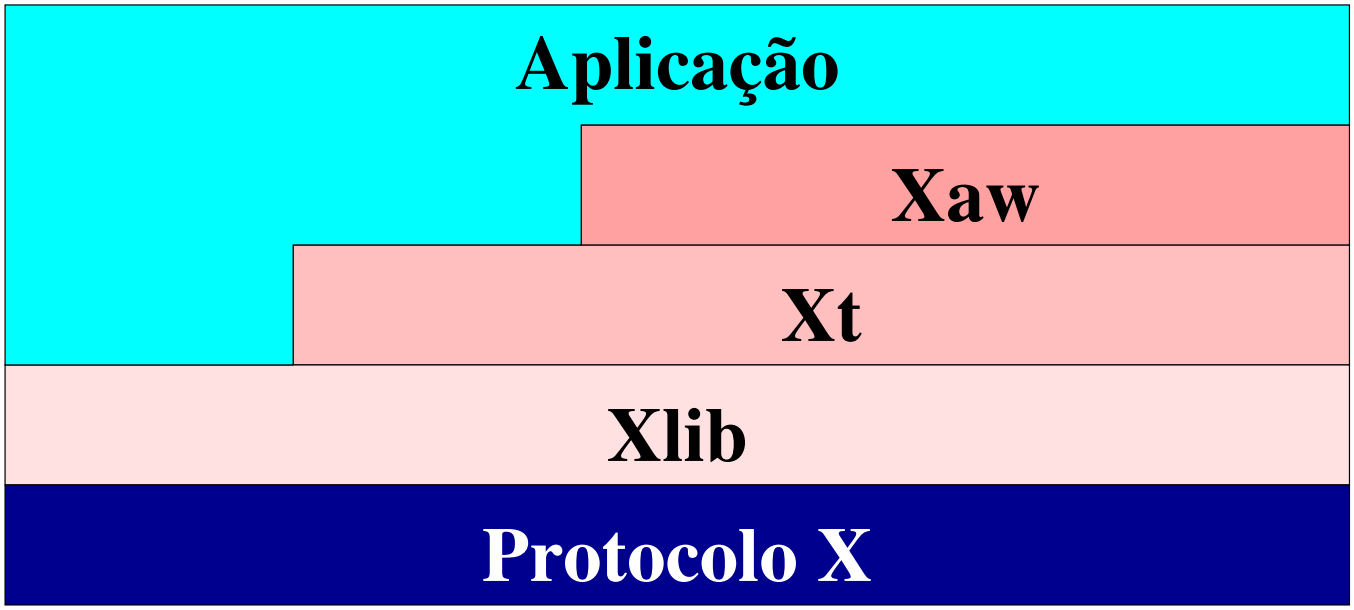

Figura 5: Estrutura de uma aplicação baseada em Xt e Xaw.

Como regra geral podemos dizer que Xaw não se presta para o desenvolvimento de aplicações grandes. Para isso existe Motif.

Toolkits — Motif

- Criado pela OSF como uma ferramenta de desenvolvimento profissional (ao contrário do Xaw).
- Interface com o usuário inspirada no DEC Windows, da Digital, no padrão CUA, da IBM, e no Windows (pré-95), da Microsoft.
- Adotado como interface gráfica padrão pelos fabricantes de Unix.
- Não é distribuído livremente como X ou Xaw.
- O fato de ser comercializado restringiu bastante o número de aplicações Motif que rodam em sistemas operacionais livres.
- Vendedores independentes de software comercializam implementações de Motif que rodam em Linux e BSD, a preços bastante acessíveis.

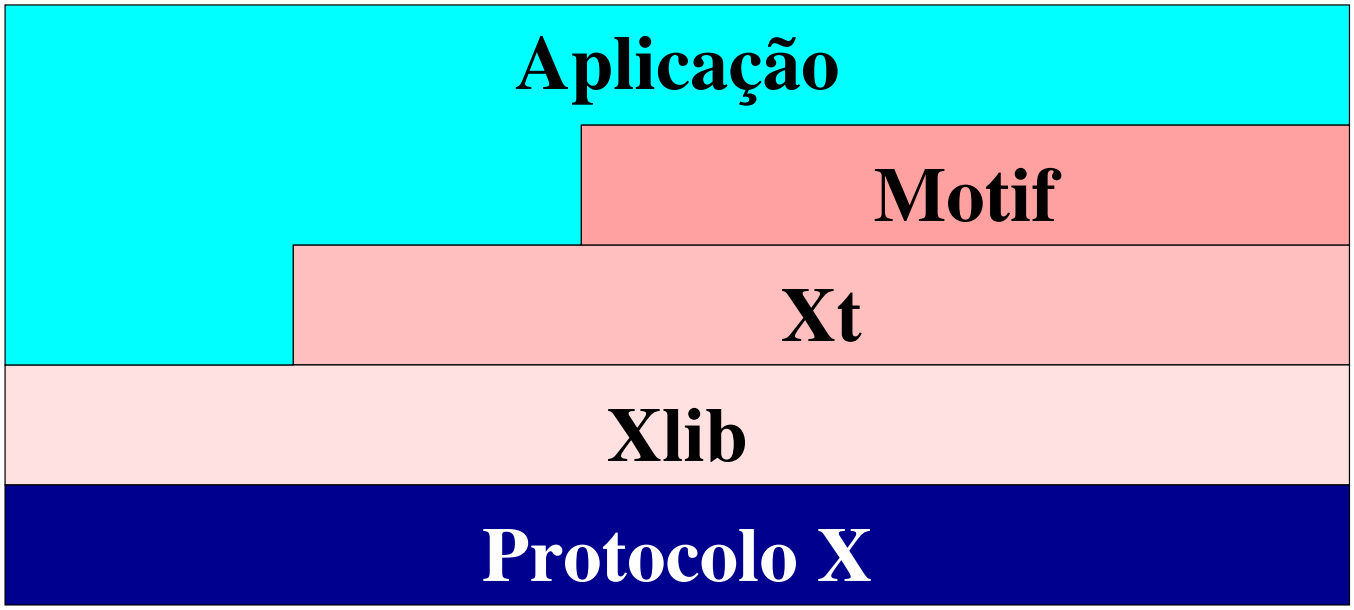

Figura 6: Estrutura de uma aplicação baseada em Xt e Motif.

Apesar de Motif ser um software comercial, sua especificação é aberta. Isso permitiu que um grupo desenvolvesse um clone sob licença GPL chamado LessTif, que já imita com bastante sucesso as características do Motif versão 1.2.

Toolkits — Gtk

- Biblioteca de widgets escrita em C, com inspiração em orientação a objetos, a exemplo do X Toolkit, criada para substituir Motif como toolkit do GIMP. Como consequência existe muita influência de Xt e Motif sobre Gtk.

- A programação pode se tornar tediosa porque é necessário usar uma disciplina de programação.

- Existem interfaces diversas linguagens, além do C. O suporte nativo à orientação a objetos da linguagem é aproveitado, quando existe.

- Esconde toda a interface de baixo nível. O programador não precisa saber nada de X Window para usá-lo, se ficar restrito aos recursos do toolkit.

- Foi criada uma versão para Microsoft Windows, também.

- Existe um construtor de interfaces gráficas para Gtk, chamado Glade.

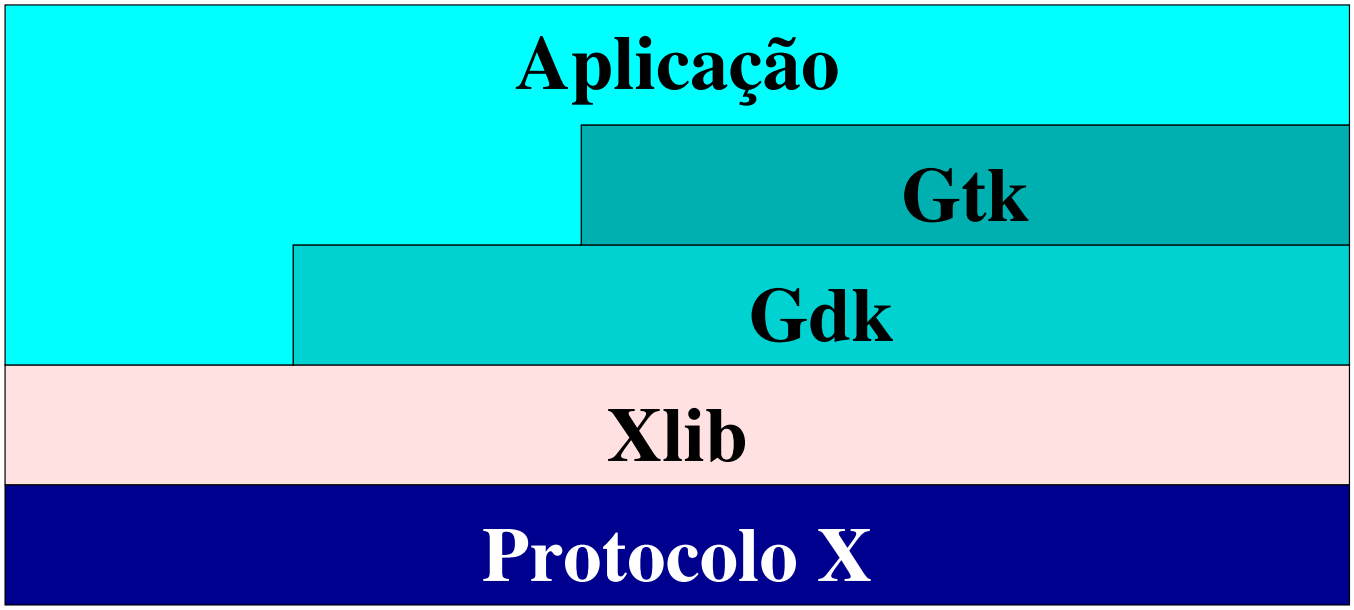

Figura 7: Estrutura de uma aplicação baseada em Gtk.

As aplicações mais conhecidas que usam Gtk são o próprio GIMP e o ambiente GNOME (GNU Network Object Model Environment). Apesar de ter muitos recusos, não é estável e há incompatibilidades entre as diversas versões.

Toolkits — Qt

- Toolkit em C++, orientado a objetos e multiplataforma, desenvolvido pela Troll Tech.
- A versão para X Window é distribuída livremente, mas as versões para outras plataformas são comerciais.
- Interface do Qt influenciada pelo Windows 95 (MFC).
- Modelo de comunicação entre widgets "signals and slots". Sinais gerados por um widget podem ser conectados a um slot existente noutro.
- Existem ambientes de desenvolvimento que incluem ferramentas de construção de interface.
- Difícil de adaptar a outras linguagens devido às diferenças com o modelo de objetos do C++.

Figura 8: Estrutura de uma aplicação baseada em Qt.

As aplicações mais conhecidas que usam Qt são o ambiente KDE (K Desktop Environment) e o navegador Opera. Devido ao fato de ser um produto comercial, suportado pela Troll Tech, o toolkit é bem documentado.

Toolkits — Tk

- Toolkit gráfico criado por John Ousterhout para a linguagem de scripts Tcl.
- A interface peovida pelo Tk é idêntica, ou pelo menos muito semelhante, à do Motif, mas a implementação é completamente diferente.
- Tcl e Tk são multiplataforma (X Window, Windows e Macintosh).
- Tcl não é apropriada para a escrita de aplicações grandes, mas é boa para de pequenos programas e prototipação.
- Tk pode ser usado também com outras linhguagens de programação, além de Tcl.
- Há um modelador de interfaces para Tcl/Tk livre chamado Visual Tcl (VTCL).

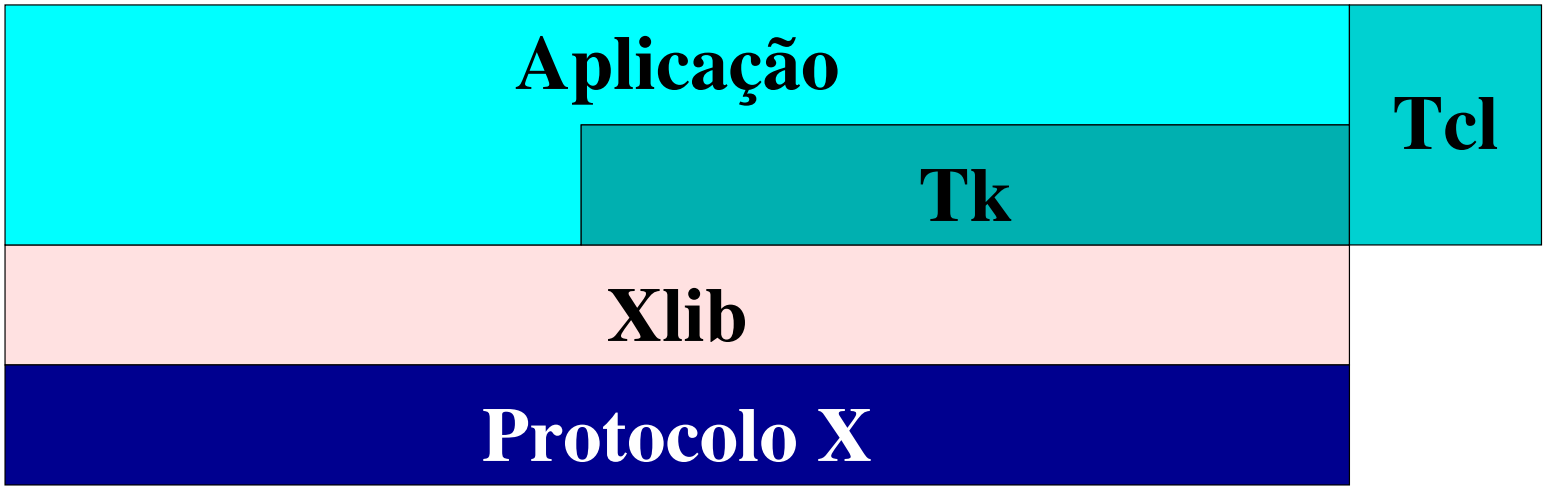

Figura 9: Estrutura de uma aplicação baseada em Tcl/Tk.

Programação

Um programa para X, realiza os seguintes passos, em linhas gerais:

1. Inicialização (conexão com servidor X)

   Tratamento de opções de linha de comando, conexão com o servidor X e alocação de estruturas internas. Isto é feito internamente pelo toolkit, caso seja utilizado um.

2. Montagem da interface gráfica

   Montagem da interface (janelas, menus, botões etc.) utilizando os widgets oferecidos pelo toolkit, ou manualmente, usando-se as primitivas do Xlib.

3. Loop de tratamento de eventos.

   Fica esperando por eventos vindos do servidor. Em toolkits, os eventos são primeiro tratados internamente e dependendo do caso são repassados ao programa através de *callbacks* ou mecanismos similares.

Exemplos de programas

Ver o material impresso e os arquivos distribuídos.

Documentação/URLs

Xlib, Xt e Xaw

- `ftp://ftp.x.org/pub/R6.4/xc/docs/hardcopy/X11/`
- `ftp://ftp.x.org/pub/R6.4/xc/docs/hardcopy/Xt/`
- `ftp://ftp.x.org/pub/R6.4/xc/docs/hardcopy/Xaw/`
- `http://www.xfree86.org/`

Motif:

- Home-page do projeto LessTif: `http://www.lesstif.org/`
- The Motif Developer: `http://www.motifzone.com/tmd`
- Motif FAQ: `http://www.rahul.net/kenton/mfaq.html`

Gtk:

- `http://www.gtk.org/`

Qt:

- `http://www.trolltech.com/`

Tcl/Tk

- Livro de John Ousterhout (veja na bibliografia).

- Scriptics: `http://www.scriptics.com/`

Seção sobre X Window Sistem do Yahoo:

- `http://dir.yahoo.com/Computers_and_Internet/Software/Operating_Systems/X_Window_System/`

Referências

[1] Scheifler, Robert; Gettys, Jim. The X Window System. ACM Transactions on Graphics, New York, ACM, 5 (2): 79-109, april, 1986.

[2] Balkovich, E.; Lerman, S.; Parmelee, R. P. Computing in higher education: The MIT Athena experience. Communications of the ACM, New York, ACM, 28 (11): 1214-1224, novembro, 1985.

[3] Liskov, B.; Scheifler, Robert. Guardians And Actions: Linguistic support for robust, distributed programs. ACM Transactions on Progamming Languages Systems, New York, ACM, 5 (3): 381-404, julho, 1983.

[4] Gettys, James & Scheifler, Robert. Xlib — C language X interface. Maynard, Digital Equipment Corporation/X Consortium, 1994.

[5] McCormack, Joel; Asente, Paul; Swick, Ralph; Converse, Donna. X Toolkit Intrinsics — C Language Interface. Digital Equipment Corporation/X Consortium, 1994.

[6] Peterson, Chris D. Athena Widget Set — C Language Interface. Maynard, Digital Equipment Corporation/X Consortium, 1994.

[7] Open Group Desktop Technologies — Motif. Disponível via WWW em <http://www.camb.opengroup.org/tech/desktop/motif/>.

[8] Ousterhout, John. Tcl and the Tk Toolkit Reading, Addison-Wesley, 1994.

[9] ZHAO, T. C.; Overmars, Mark. Forms Library: A Graphical User Interface for X. Disponível via WWW em <http://bragg.phys.uwm.edu/ xforms>.

[10] Mattis, Peter; Kimball, Spencer; MacDonald, Josh. The GIMP toolkit. Disponível via WWW em <http://www.gtk.org>.

[11] Scheifler, Robert. X Window System Protocol. Cambridge, X Consortium, 1994.

[12] The Open Group. X Window System, Version 11, Release 6.4: Release Notes. Open Group, 1998.

[13] Tanembaum, Andrew S. Computer Networks. Upper Saddle River, Prentice Hall, 1996.

[14] Jones, Oliver. Introduction to the X Window System Englewood Cliffs, Prentice Hall, 1990.

[15] Rosenthal, David; Marks, Stuart W. Inter-Client Communication Conventions Manual. Sun Microsystems/X Consortium, 1994.

[16] The F? Virtual Window Manager home page. Disponível via WWW em <http://www.fvwm.org>

[17] Ghostview and GSview.Disponível via WWW em <http://www.cs.wisc.edu/~ghost>

[18] Lesstif. Disponível via WWW em <http://www.lesstif.org>.

[19] Taylor, Lloyd. Client/Server Frequently Asked Questions. Disponível via WWW em <http://www.abs.net/~lloyd/csfaq.txt>.

[20] Tanembaum, Andrew S. Sistemas Operacionais Modernos. Rio de Janeiro, Prentice Hall do Brasil, 1995.