Tvorba uživatelských rozhraní
Systém X Window

Jozef Mlích

Ústav počítačové grafiky a multimedií
Fakulta informačních technologií, Vysoké učení technické v Brně
Božetěchova 2, 612 66 Brno, Czech Republic
http://www.fit.vutbr.cz/~imlich/
imlich@fit.vutbr.cz

30.10.2014

Agenda

- ~~Historie X – od 1984~~
- Principy a funkce
  - XServer
  - XProtocol
  - Rozšíření
  - Užitečné nástroje
- Programování XAplikací
  - Xlib
  - toolkity (xtoolkit, motif, gtk, qt, wxWindows)
  - Makefile/imake/qmake/autotools/cmake
  - printf(“Hello XWindow\n”);

Principy a funkce - XServer

Co to je?

- nádstavba operačního systému poskytující grafické uživatelské rozhraní (pro linux/qnx/windows/atd.)

Co to umí:

- architektura klient-server
- síťově transparentní
- nezávislé na hardware
- ani centralizované/ani distribuované zpracování
- rozšiřitelné

Principy a funkce - XServer 2

Uživatelské prostředí je dáno:

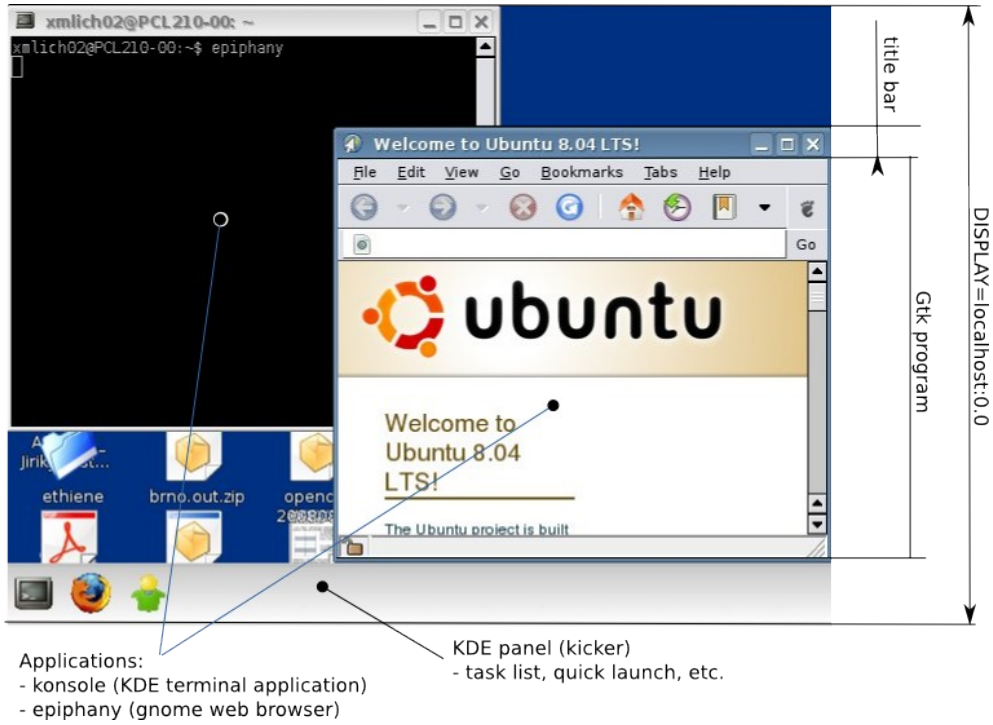
- prostředky XServeru
- správce oken (window manager)
  - vzhled a umístění oken (KDE, gnome, blackbox, ...)
  - ovladací prvky oken (minimalizovat/ukončit/...)
- správce sezení (session manager)
  - stará se o okna v rámci jednoho přihlášení (xsm, ksmserver)
  - X Session Management Protocol (XSMP)
- knihovny nástrojů (toolkity - Qt, Gtk, Xtoolkit, Motif)
- aplikace

Principy a funkce - XServer 3

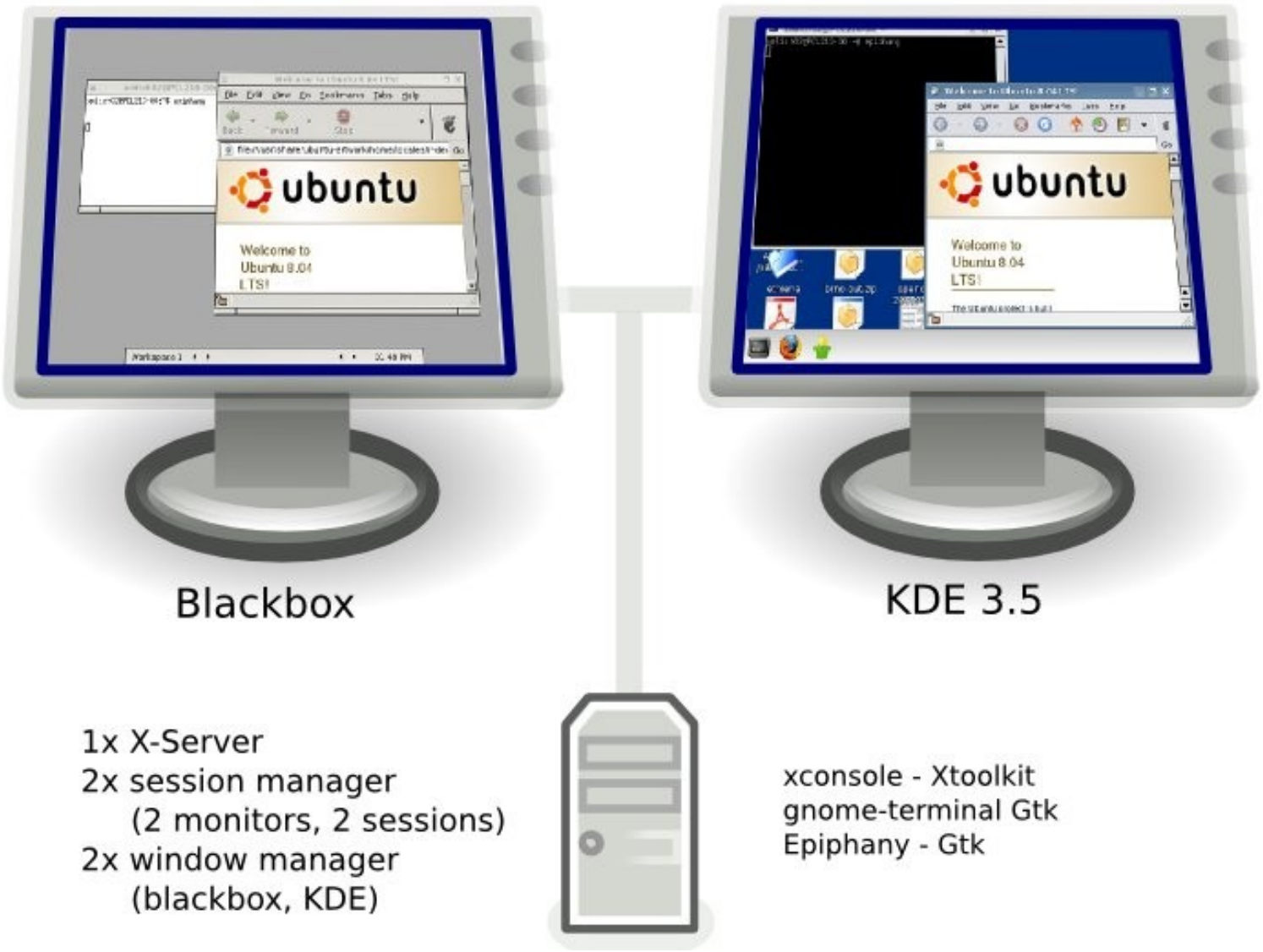
- Jednotlivá okna jsou ve **stromové struktuře**
- Hlavní okno je **správce oken** (window manager)
- Prvky mají relativní pozici od levého horního okna
- Každé okno se zpracovává v rámci X serveru samostatně

Principy a funkce - XServer 4 - příklad

- jedna session
- správce oken je KDE
  - vykreslování záhlaví a rámečků oken
  - obsahuje zvláštní aplikace (plocha / panel – zobrazuje seznam úloh, atd.)
- programy
  - napsané s použitím různých toolkitů (konsole je založený na KDE/Qt, epiphany používá Gtk)

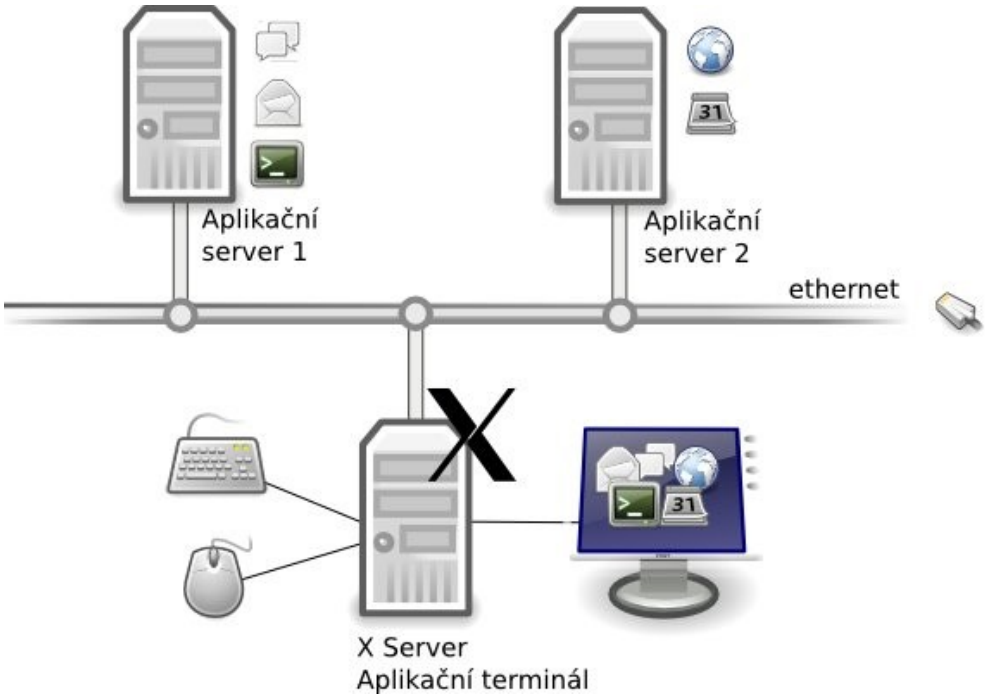


Principy a funkce - XServer 5



X Server – X Klient
vs.
Server – Klient

- X server beží na grafickém terminálu – stará se o vykreslování, myš a klávesnici
- Aplikace (výpočty/diskové operace) běží vzdáleně a připojuje se přes síť na X Server a tam se vykresluje

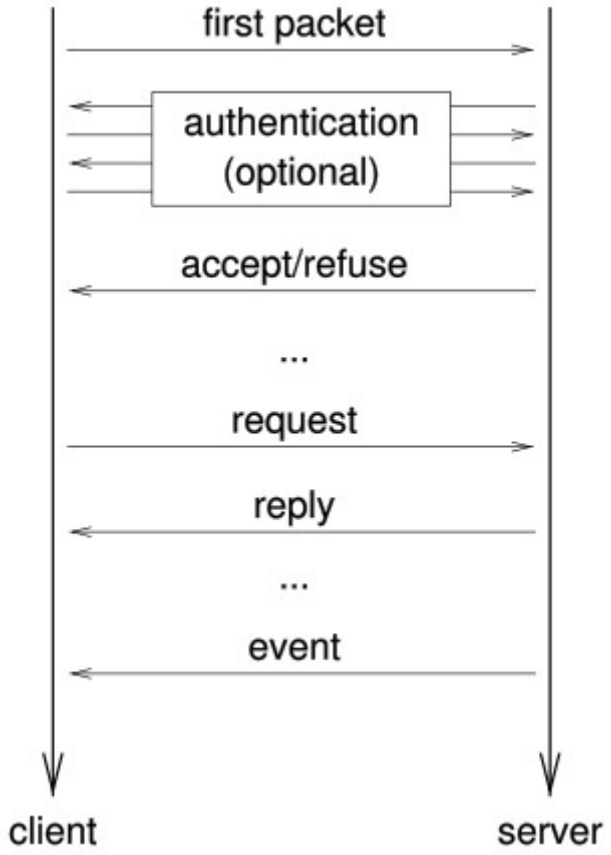


Principy a funkce - X Window Core Protocol

- musí podporovat každý XServer a X klient
- může běžet nad TCP/IP, rourou (pipe), sdílenou pamětí
- používají ho všechny aplikace (správce oken, správce sezení taky)
- XServery/Xaplikace různých výrobců jsou vzájemně kompatibilní
- Snaha o kompatibilitu na úrovni zdrojových kódů i na binární úrovni.

Principy a funkce - X Window Core Protocol 2

- Požadavky (requests)
  - zprávy posílané serveru
- Odpovědi (replies)
  - posílá je server jako odezvu na požadavek
- Události (events)
  - autonomní informování o události
  - například “key pressed” apod.
- Chyby (errors)
  - "chybové události"



Principy a funkce - X Window Core Protocol 3

- Okna
  - Dáno velikostí a geometrií
  - Stromová hierarchie
  - InputOutput (normální) vs. InputOnly (neviditelné – události)
- Pixmapy
  - kus paměti / automaticky se nevykresluje
  - Jde zobrazit do okna (double buffering)
- Grafický kontext (GC) a fonty
  - podobně jako ve windows
  - pamatuje si kam kreslí (okno/pixmapa), barvu pozadí, barvu popředí, font, atd.
  - fonty jsou uložené na severu nastavují se v rámci GC

Principy a funkce - X Window Core Protocol 4

- Resources and identifiers
  - 32-bit integer jednoznačně se odkazující na nějaký zdroj (Resource) na xserveru
  - Resource: Window, Pixmap, Font, Colormap, Graphic context
- Events
  - KeyPressed, etc.
  - Expose (Klient chce překreslit část okna – např. změnila se hodnota progressbaru a je nutné ho překreslit znovu)
- Atoms
  - z hlediska XServeru je 32 bitový int atomický

Principy a funkce - Rozšíření

Protokol lze rozšířit o další funkcionalitu.

- DPMS – řízení spotřeby
- MIT-SHM – přenos rastrových obr. přes sdílenou paměť
- SHAPE – neobdelníková okna
- RANDR – dynamická změna rozlišení a rotace displeje
- RENDER – antialiasing, průhlednost, atd.
- XKEYBOARD – podpora národních rozložení
- XINEREMA – podpora zobrazení přes více monitorů
- XVideo (XV) – hardwarová akcelerace videa

Principy a funkce - Užitečné nástroje

- Vzdálené připojení
  - ssh -X
  - xming (http://sourceforge.net/projects/xming), xwinlogon
- xnest
  - více xserverů současně
- xdpyinfo
  - seznam podporovaných rozšíření a další informace
- xwininfo
  - detailnější informace o okně
- glxinfo
  - informace o opengl rozšíření X serveru

Principy a funkce - Užitečné nástroje 2

- xrandr
  - otáčení displaye, nastavení rozlišení
  - Section "Screen"
    Option "RandRRotation" "true"
- xev
  - zobrazuje události X okna
- zenity/kdialog/xdialog
  - interakce s gui pomocí shellových skriptů
  - zenity --file-selection
  - kdialog --yesno "zajimaji vas tyhle kraviny?"
- xnee – záznam a přehrávání uživatelských akcí
- xrestop – prostředky alokované na xserveru

Principy a funkce - Užitečné nástroje 3

- xwit – nastavování parametrů oken
- xautomation – sada command line nástrojů (xte)

- http://en.wikipedia.org/wiki/List_of_widget_toolkits#On_Unix.2C_under_the_X_Window_System
- http://en.wikipedia.org/wiki/List_of_widget_toolkits#Cross-platform
-

- API zpřístupňující X Protokol v C
- Nejdůležitější funkce
  - XOpenDisplay(char *display_name), XCloseDisplay, XSetCloseDownMode
  - char *display_name="**hostname:number.screen_number**"
  - XLockDisplay, XUnlockDisplay
  - XAddConnectionWatch, XRemoveConnectionWatch
  - XCreateWindow, XCreateSimpleWindow, XDestroyWindow, XDestroySubwindows
  - _X11TransConnectDisplay, _X11TransGetConnectionNumber, _XSendClientPrefix, _X11TransGetConnectionNumber

- Grafické operace

body (XDrawPoint), čáry (XDrawLine), obdélníky (XDrawRectangle), mnohoúhelníky (XDrawSegments), kruhové výseče, kružnice a elipsy (XDrawArc), texty (XDrawString, XDrawImageString, XDrawText), kopie oken, pixmap (XCopyArea, XCopyPlane), obrázky (XPutImage), kurzory (XDefineCursor)

- Akce uživatele
  - KeyPress, KeyRelease, MappingNotify, FocusIn, FocusOut
  - ButtonPress, ButtonRelease, MotionNotify, EnterNotify, LeaveNotify

- Signalizace stavu
  - Expose, GraphicsExpose, NoExpose, ColormapNotify, VisibilityNotify
  - CirculateNotify, ConfigureNotify, CreateNotify, DestroyNotify, GravityNotify, MapNotify, ReparentNotify, UnmapNotify
- Zprávy od jiné aplikace
  - ClientMessage, PropertyNotify, SelectionClear, SelectionRequest, SelectionNotify

Programování XAplikací - Xlib 4 - Xlib vs. WinAPI

```c
#include <X11/Xlib.h>
#include <stdio.h>
#include <stdlib.h>
int main() {

  Display *dpy; Window rootwin; Window win; Colormap cmap; XEvent e;
  int scr; GC gc;

  if(!(dpy=XOpenDisplay(NULL))) {
    fprintf(stderr, "ERROR: could not open display\n");
    exit(1);
  }

  scr = DefaultScreen(dpy);
  rootwin = RootWindow(dpy, scr);
  cmap = DefaultColormap(dpy, scr);

  win = XCreateSimpleWindow(dpy, rootwin, 1, 1, 100, 50, 0,
    BlackPixel(dpy, scr), BlackPixel(dpy, scr));

  XStoreName(dpy, win, "hello");

  gc=XCreateGC(dpy, win, 0, NULL);
  XSetForeground(dpy, gc, WhitePixel(dpy, scr));
  XSelectInput(dpy, win, ExposureMask|ButtonPressMask);

  XMapWindow(dpy, win);

  while(1) {

    XNextEvent(dpy, &e);
    if(e.type==Expose && e.xexpose.count < 1) {
      XDrawString(dpy, win, gc, 10, 10, "Hello World!", 12);
    } else if(e.type==ButtonPress) {
      break;
    }
  }

  XCloseDisplay(dpy);
  return 0;
}
```

```c
#include <windows.h>
#include <stdio.h>
#include <string.h>

// Global variable

HINSTANCE hInst;

// Function prototypes.
int WINAPI WinMain(HINSTANCE, HINSTANCE, LPSTR, int);
LRESULT CALLBACK MainWndProc(HWND, UINT, WPARAM, LPARAM);

// Application entry point.

int WINAPI WinMain(HINSTANCE hInstance, HINSTANCE hPrevInstance,
  LPSTR lpCmdLine, int nCmdShow) {
    MSG msg; BOOL bRet; WNDCLASS wcx; HWND hWnd; UINT uResult;

    hInst = hInstance;

    wcx.style = CS_HREDRAW | CS_VREDRAW;
    wcx.lpfnWndProc = (WNDPROC) MainWndProc;
    wcx.cbClsExtra = wcx.cbWndExtra = 0;
    wcx.hInstance = hInstance;
    wcx.hIcon = LoadIcon(NULL, IDI_APPLICATION);
    wcx.hCursor = LoadCursor(NULL, IDC_ARROW);
    wcx.hbrBackground = GetStockObject(WHITE_BRUSH);
    wcx.lpszMenuName =  (LPCSTR) "MainMenu";
    wcx.lpszClassName = (LPCSTR) "MainWClass";

    if (!RegisterClass(&wcx)) return FALSE;

    hWnd = CreateWindow("MainWClass", "x:50, y:100", WS_OVERLAPPEDWINDOW, 50,
      100, 750, 150, (HWND) NULL, (HMENU) NULL, hInstance, (LPVOID) NULL);
    if (!hWnd) return FALSE;

    ShowWindow(hWnd, nCmdShow);
    UpdateWindow(hWnd);

    while( (bRet = GetMessage( &msg, NULL, 0, 0 )) != 0) {
        if (bRet == -1) {
            // handle the error and possibly exit
        } else {
            TranslateMessage(&msg);
            DispatchMessage(&msg);
        }
    }
    return (int) msg.wParam;
}
```

Suma sumárum:

- programování gui aplikací s Xlibem je hodně hardcore
- hodí se na low level věci
  - například čtečka pro slepce, analýza uživatelského rozhraní (odchytávání zpráv, apod.)
  - vlastní gui toolkit
  - hackování XServeru (akcelerace)

Budoucnost

- XCB (X C Binding) – menší komplexnost, blíže X protokolu

Programování XAplikací - Xtoolkit, motif

- rozšíření Xlibu o některé základní prvky (tlačítka, scrollbary)
- stále docela hardcore

Tvorba uživatelských rozhraní
Gtk

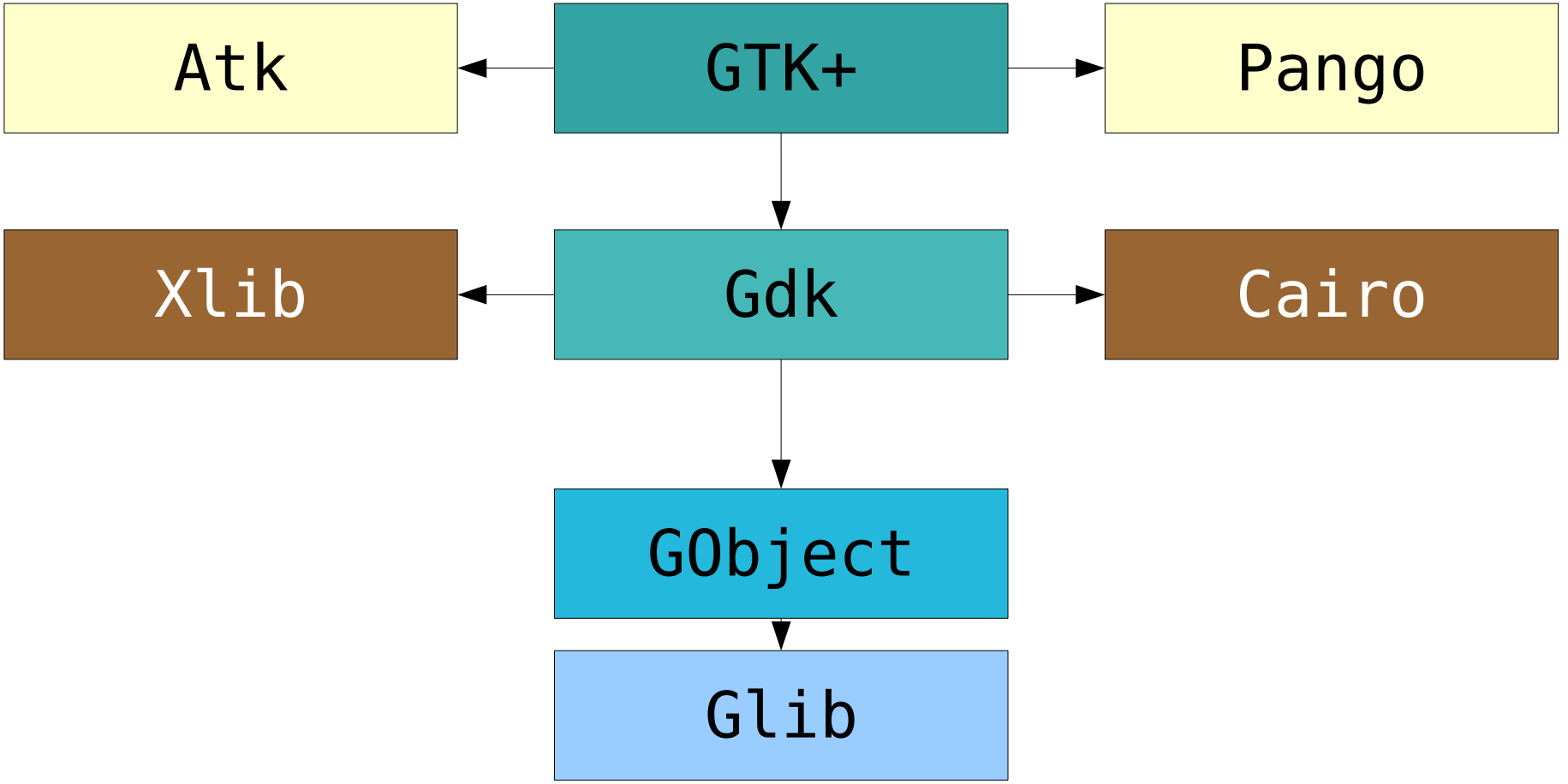
Jozef Mlích

Ústav počítačové grafiky a multimedií
Fakulta informačních technologií, Vysoké učení technické v Brně
Božetěchova 2, 612 66 Brno, Czech Republic
http://www.fit.vutbr.cz/~imlich/
imlich@fit.vutbr.cz

30.10.2014

- hodně používané kvůli LGPL
- Objekty v C.
- Architektura
- Jmenné konvence
- Udalosti / rozmístění prvků
- program se nakreslí v Glade / na zmenu stylu se pouzije GtkInspector

GTK+ Architektura



Glib Libraries

- **Glib** (základní typy/funkce/pomocné třídy např seznamy/pole/stromy)
- **GObject** (signaly, vlastnosti, správa paměti, dědičnost)
- **GModule** ("plugins")
- **GThread** (vlákna, synchronizace)
- **GIO** (virtualní souborové systémy, síťování, d-bus)

Jmenné konvence

- `g_` prefix

  např. `g_object_unref()`

- Makra - `G_` prefix

  např `G_OBJECT()`

- Gtk mají `gtk_`

  např `gtk_widget_show()`

- ne CamelCase!

GObject

- Ukazatel na objekt je vždy první parametr funkce (a pointer) is always passed as a first argument

  např. `gtk_widget_show (button1);`

- Počítaní referencí: `g_object_ref()`, `g_object_unref()`
- “konstruktory”: `g_xxx_new()`
- “destruktory”: `g_free()` or `g_object_unref()`

GObject

- pseudo-třídy:
  - konstructory, destruktory, OOP principy
  - vlastnosti, notifikace
- přetypování: `G_OBJECT()`, `G_CALLBACK()`, `GTK_CONTAINER()`

GObject – object hierarchy

```
GObject
+----GtkWidget
     +----GtkContainer
          +----GtkBin
               +----GtkButton
                    +----GtkToggleButton
                         +----GtkCheckButton
```

Glib – hlavní smyčka

- mainloop v hlavním vláknu, čeká na události
- “události == sinály”
- Odeslání: `g_signal_emit()`
- propojení: `g_signal_connect()`
- Zpracování: callback
- callback blokující: `g_signal_handler_block()`

Inicializace

- `gtk_init()`
- `gtk_widget_show()`
- `gtk_main()`
- `gtk_main_quit()`

Události

```
g_signal_connect (button, "clicked",
        G_CALLBACK (hello), NULL);
```

```
static void hello (GtkWidget *widget,
                   gpointer   data)
{
    g_print ("Hello World\n");
}
```

Události II

- nejčastěji:
  - `changed`
  - `clicked`
  - `focus`
  - `button-press-event`
  - `motion-notify-event`
  - `key-press-event`
- callback: `return {FALSE,TRUE};`

Vlastnosti - properties

- ```
  gtk_container_set_border_width
              (GTK_CONTAINER (window), 10);
  ```

- ```
  value = gtk_toggle_button_get_active
              (GTK_TOGGLE_BUTTON (toggle2));
  ```

- ```
  gtk_toggle_button_set_active
              (GTK_TOGGLE_BUTTON (toggle2),
               TRUE);
  ```
  ```

Layout containers

- Nepoužívají se absolutní souřadnice
- Flexibilní nastavení velikosti okna
- Neviditelné obdelníky pro zalamování
- `GtkContainer → GtkBox`
- `GtkContainer → GtkGrid`

- ```
  gtk_container_add (GTK_CONTAINER (window),
                     button);
  ```
  ```

Nástroje

- Glade
- Gtkinspector / css

Literatura

- http://en.wikipedia.org/wiki/X_Window_System
- http://en.wikipedia.org/wiki/X_protocol
- http://tronche.com/gui/x/xlib/
- http://www.kiv.zcu.cz/~luki/vyuka/stare-materialy/os/oslinux/2.0.31/sak4/xx.htm
- http://www.gtk.org/
- http://developer.gnome.org/gtk3/stable/
- Warkus, M.: Official GNOME 2 Developer's Guide, O'Reilly, 2004, ISBN: 1-59327-030-5
- Sheets, J.R.: Writing GNOME applications, Addison-Wesley, 2001, ISBN 0-201-65791-0
- Krause, A.: Foundations of GTK+ Development, Apress, 2007, ISBN13: 978-1-59059-793-4
- Nye, A., O'Reilly, T.: X Toolkit Intrinsics Programming Manual, OSF/Motif Edition, O'Reilly & Associates, 1990, ISBN 0-937175-62-5
- Scheifler, R., W., Gettys, J.: X Window System, The Complete Reference to Xlib, X Protocol, ICCCM, XLFD, Digital Press, 1990, ISBN 1-55558-050-5