

Vývoj v C++

...0100100100100..01010...
...0100100100100..01010...2
...0100100100100..01010...

FI MUNI, Brno 2008
Juraj Michálek

e-mail: michalek@atollon.com

URL: <http://georgik.asinus.org>

jabber: [georgik@jabber.e-region.net](jabber:georgik@jabber.e-region.net)



■ Blok 1. - vývoj softvéru

- modely vývoja, Agile, XP, PDCA

■ Blok 2. - interaktívne aplikácie, skriptovanie

- Allegro, SDL, ClanLib
- LUA, Mesa3D

■ Blok 3. - aplikácie s GUI

- QT, FLTK, GTK

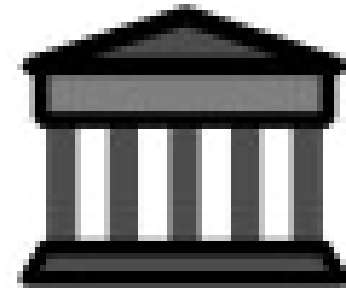
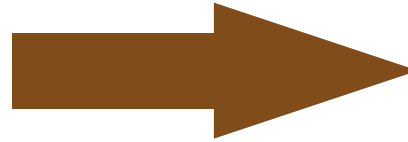
■ Blok 4. - server-klient

- Boost, ACE, JSON, XML, Axis

■ Blok 5. - vývojárske nástroje

- Eclipse, Subversion, Mono, Python....

Blok 1.



Vývoj softvéru

Prvé kroky

- Ako to vyzerá pri písaní programov?
- 1. programátor dostane skvelý nápad
- 2. naprogramuje dokonalý program
- 3. všetko si naprogramuje sám
- 4. zistí, že program je nepoužiteľný

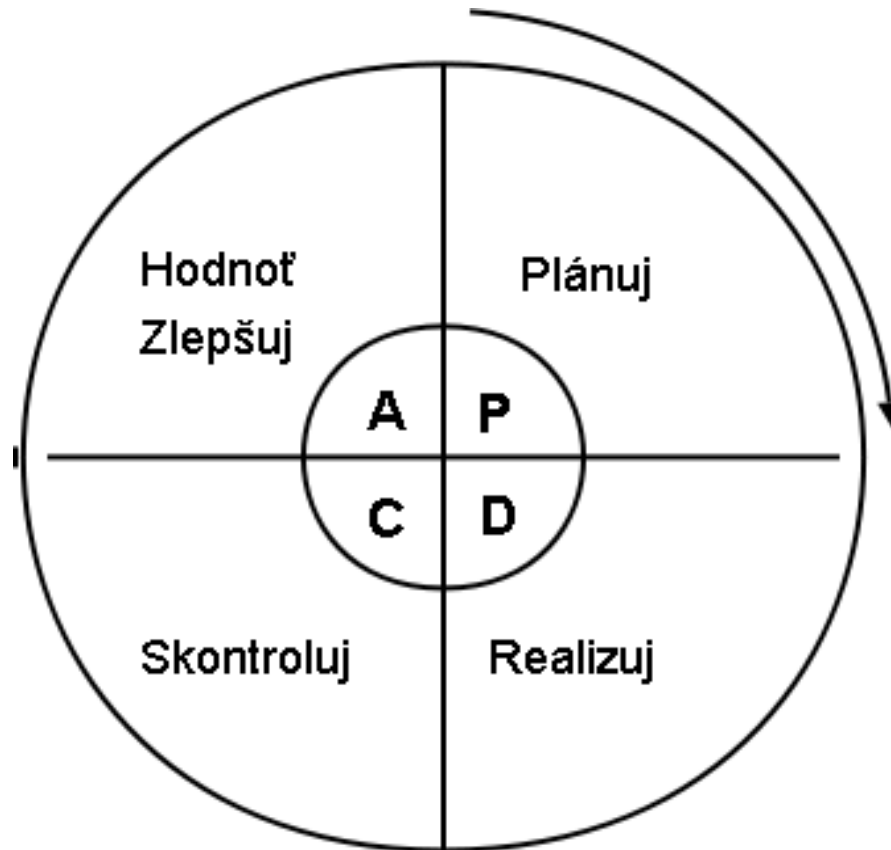
Na vlastných nohách

- 5. pokúsi sa program opraviť
- 6. zistí, že polovicu funkcií naprogramoval zle
- 7. preprogramuje zlé funkcie
- 8. zistí, že program je opäť nepoužiteľný
- goto 5.

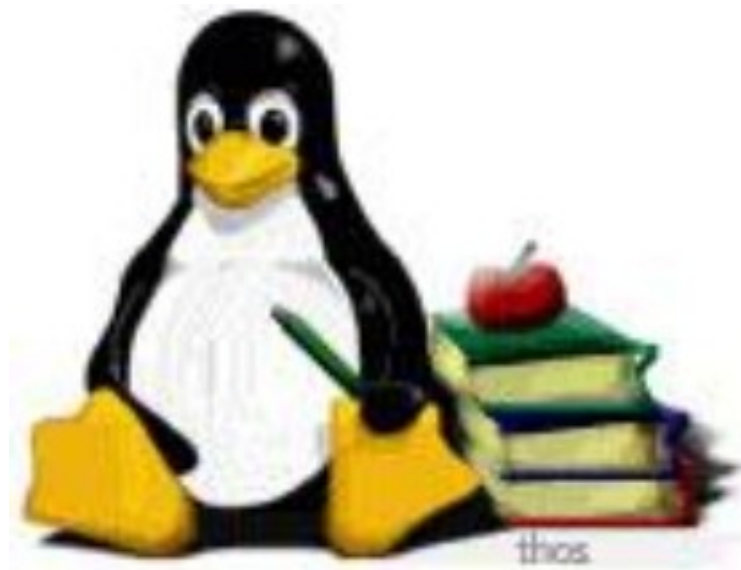
Modely vývoje

- **Agilné metódy**
- **Scrum**
- **Extreme Programming**

Deming wheel



Starndard Template Library



STL I.

- základ pre C++
- práca s reťazcami
 - ➔ `#include <string>`
- operácie nad zoznamami
 - ➔ `#include <list>`
 - ➔ `#include <deque>`

STL II.

- operácie s asociatívnymi poliami
- ➔ `#include <map>`
- špeciálne algoritmy (triedenie, výpis)
- ➔ `#include <algorithm>`
- zhrnutie: umožňujú urýchlenie vývoja
- ➔ <http://www.sgi.com/tech/stl/>

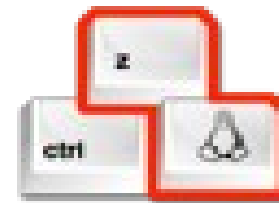
Otázka

- spracovanie vstupov
- klávesnica, myš
- zobrazenie výstupov
- obrazovka
- vystačíme so štandardnými funkciami?

Odpooved'



Blok 2.



Interaktívne aplikácie a skriptovanie

Interaktívne aplikácie

- rýchla reagovať na užívateľa
- musia sa prispôsobovať jeho potrebám
- dobrým príkladom je počítačová hra
- ukážeme si konkrétnu implementáciu

Vhodné knihnice

Allegro

Simple Direct Media Layer

Clanlib Simple Development Kit

Knižnica

Allegro

Allegro I.

- **Allegro Low Level Game Routines**
- **vznikla na platforme Atari ST**
- **Atari pod tlakom platformy PC zaniklo**
- **našťastie nezanikli myšlienky Allegra**

Allegro II.

- bolo prepísané pre Borland C
- adaptované na DJGPP
- podpora pre DirectX a Xserver

Allegro - aplikácia

- spracovanie klávesnice, myši, joysticku
- spracovanie obrazu, zvuku
- jednoduché GUI
- ovládanie časovačov
- používanie dátových súborov

Programovanie I.

- inicializácia Allegra

- ➔ `allegro_init();`

- ukončenie Allegro

- ➔ `allegro_exit();`

- za funkciou main je nutné uviesť makro

- ➔ `END_OF_MAIN();`

Inicializácia grafiky

- nastavenie grafickej hĺbky

- `set_color_depth(int depth)`

- inicializácia grafického prostredia

- `set_gfx_mode(int card,
 int w, int h,
 int v_w, int v_h)`

BITMAP

- štruktúra na uloženie obrazu
- obrazovka je BITMAP
- ➔ `BITMAP *screen`
- vytvorenie obrazovej pamäte
- ➔ `BITMAP *bmp = create_bitmap(x, y);`
- vymazanie obrazovej pamäte
- ➔ `clean(bmp);`

Grafika

- vykreslenie bodu do pamäte

- ➔ `putpixel (bmp, x, y, color)`

- načítanie obrazu zo súboru

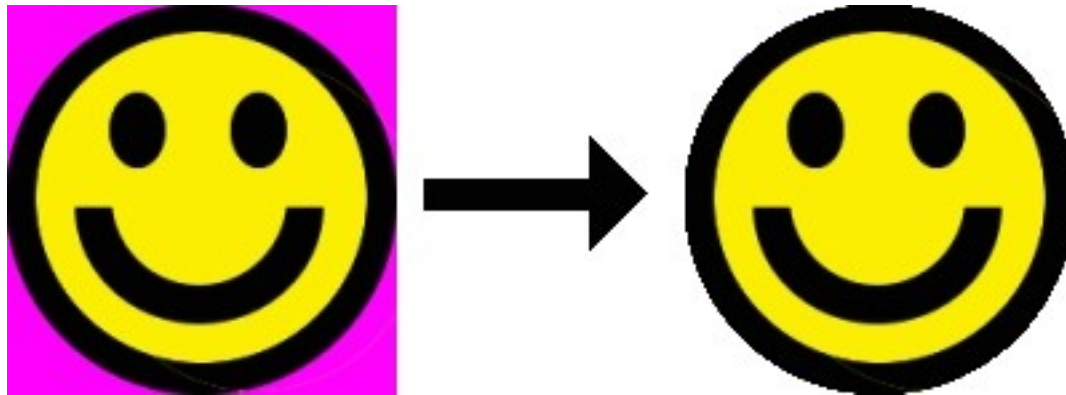
- ➔ `BITMAP *load_bitmap (filename,
palette);`

- kopírovanie medzi pamät'ami

- ➔ `blit (bmp, screen, 0, 0, 0, 0,
320, 200);`

Sprite

- ▣ okolie obrazu je transparentné
- ➔ `draw_sprite(screen, bmp, x, y);`



Grafika

- Uvoľnenie obrazovej pamäte
- ➔ `destroy_bitmap (bmp) ;`



Klávesnica I.

- zavedenie ovládača klávesnice
 - + `install_keyboard()`
- odstránenie ovládača klávesnice
 - + `remove_keyboard()`
- načítanie klávesy
 - + `readkey()`
- vymazanie zásobníka kláves
 - + `clear_keybuf()`

Klávesnica II.

- zistenie stlačenia klávesy
 - ➔ `int keypressed()`
- test na stlačenie konkrétne klávesy
 - ➔ `key [KEY_typ]`



Časovač I.

- časovač zaist'uje rovnakú rýchlosť
- inštalácia časovača
 - + `install_timer()`
- odstránenie časovača
 - + `remove_timer()`

Časovač II.

- časovač predstavuje ako počítadlo
- ➔ `volatile int counter;`
- v pravidelných intervaloch je volaná určitá funkcia
- ➔ `void my_timer_handler()`
- inštalácia funkcie volenej v pravidelných intervaloch
- ➔ `install_int_ex(my_tim, speed);`

Časovač III.



Myš

■ inštalácia a odstránenie ovládača

➔ `install_mouse()`

➔ `remove_mouse()`

➔ `mouse_x` // X súradnica

➔ `mouse_y` // Y súradnica

➔ `mouse_z` // poloha koliečka

➔ `mouse_b` // stav tlačítok

Text

- jednoduchý výpis textu
- ➔ `textprintf(bmp, font, x, y, color, text);`



Ďalšie vlastnosti

- prehrávanie samplov a MIDI
- ukladanie súborov do datafile
- jednoduchá 3D grafika - polygony
- jednoduché GUI

Knižnice nad Allegrom

- Allegro má pomerne veľa funkcií
- Allegttf - práca s True Type fontami
- AllegGL - práca GL grafikou
- DUMB - práca sa audiom (Vorbis/Ogg)

Kompilácia

- Konfiguračný skript

- ➔ `allegro-config`

- Linux, Unix, BeOS, QNX, MacOS, Windows:

- ➔ `g++ foo.cc -o foo`
 ``allegro-config --libs``

Zdroj

■ **vývoj knihnice**

➔ <http://alleg.sourceforge.net>

Allegro.cc

- Game development community network
- Matthew J. Leverton
- ➔ <http://www.allegro.cc>

{ allegro.cc }

game developing community network

Knižnica



SDL

- **Simple Direct Media Layer**
- **knižnica vznikla na Linuxe**
- **pre veľký úspech bola prenesená na ďalšie systémy**

Aplikácia

- spracovanie klávesnice, myši, joysticku
- spracovanie obrazu, zvuku
- ovládanie časovačov
- podpora udalostí a vlákien
- mplayer (Linux Movie Player)

Rozdiely

- kvalitne spracovaný systém udalostí
- zložitejší prístup ku grafike
- prepracovanejšie grafické funkcie
- SDL samo o sebe je low level knižnica
- používa sa s nadstavbami

Programovanie

- inicializácia systému
 - ➔ `SDL_init (flags)`
- ukončenie systému
 - ➔ `SDL_quit ()`

Grafika I.

- inicializácia grafického prostredia
 - ➔ `SDL_SetVideoMode(width, height, video_bpp, videoflags)`
- obrazové dáta v pamäti
 - ➔ `SDL_SWSURFACE`
- obrazové dáta vo video pamäti
 - ➔ `SDL_HWSURFACE`

Grafika II.

- štruktúra na uloženie obrazu

- `SDL_Surface`

- vytvorenie obrazovej pamäte

- `surface =`

```
SDL_CreateRGBSurface (SDL_SWSURFACE,  
width, height, 32, rmask, gmask,  
bmask, amask);
```

Grafika III.

- **vymazanie obrazovej pamäte**
 - ➔ `SDL_FillRect (screen, NULL, color)`
- **Uvoľnenie obrazovej pamäte**
 - ➔ `SDL_FreeSurface (SDL_Surface
*surface) ;`

Grafika IV.

- načítanie obrázu zo súboru

- ➔ `SDL_Surface *SDL_LoadBMP (const char *file)`

- kopírovanie medzi pamäťami

- ➔ `SDL_BlitSurface (src, srcrect, dst, dstrect);`

Klávesnica

- stlačenie klávesy vyvolá udalosť
- na zistenie stlačenej klávesy je nutné odchytiť udalosť
- ➔ `SDL_PollEvent (SDL_Event *event)`
- test na stlačenie konkrétne klávesy
- ➔ `event . key . keysym . sym`

Časovač

- časovač implementovaný ako počítadlo
- ➔ `static int counter;`
- inštalácia funkcie volenej v pravidelných intervaloch
- ➔ `SDL_SetTimer(interval, tick);`

Myš

- zistenie stavu myši

- ➔ `SDL_GetMouseState (*x, *y);`

Text

- jednoduchý výpis textu nie je implementovaný



Ďalšie vlastnosti

- priama podpora GL knižníc
- zabudovaný mechanizmus pre vlákna
- spolupráca s jazykmi
 - `Ada, Eiffel, Java, Lua, Perl, PHP, Pike, Python, Ruby`
- 130 rozširujúcich knižníc

Knižnice

- **SDL je samo o sebe pomerne t'azko pouzitel'ne**
- **SDL_mixer - multikanalovy mixer**
- **SDL_image - spracovanie obrazovych formátov**
- **SDL_net - práca so sieťou**

Zdroje

- **vývoj**

- ➔ <http://www.libsdl.org>

- **portál s hrami**

- ➔ <http://www.lgames.org>

Kompilácia

- konfiguračný skript
- ➔ `sdl-config`
- Linux, Windows, BeOS, Mac OS
- ➔ `g++ foo.cc -o foo`
 ``sdl-config --libs``

Knižnica



Clanlib game SDK

- mamutia knižnica
- ClanLib game Simple Development Kit
- kompletne naprogramované v C++
- vytvorené ako komplexné API

Aplikácie

- spracovanie 2D a 3D grafiky
- spracovanie zvuku, I/O
- spracovanie fontov (ttf)
- obsahuje nadstavbu pre tvorbu GUI

Programovanie

- inicializácia systému
 - ➔ `CL_SetupCore::init()`
- ukončenie systému
 - ➔ `CL_SetupCore::deinit()`

Grafika

- inicializácia grafického prostredia

- ➔ `CL_Display::set_videomode(width, height, depth)`

- vytvorenie plátna

- ➔ `CL_Canvas canvas(width, height);`

GUI

- GUI definition files (.gui)
- definícia GUI oddelená od kódu
- za behu programu je možné nahrávať GUI popisy
- popis má štruktúru kódu v jazyku C

OpenGL

- **CL_Textures** -- základná správa textúr
- **CL_SetupGL** -- nastavenie GL vlastností
- **CL_OpenGL** -- efekty OpenGL

Kompilácia

```
➔ g++ foo.cc -o foo -LclanApp
```

Ďalšie vlastnosti

- prehrávanie samplov a MIDI
- podpora pre JavaScript
- implementovaný MikMod - prehrávanie music modules
- podpora prehrávania OggVorbis
- priama podpora sieťovania, správa signálov

Zdroj

■ **vývoj**

➔ <http://www.clanlib.org>

LUA

- silný skriptovací nástroj
- volanie skriptov z programu
- volanie programu zo skriptov
- jednoduchá syntax jazyka
- ➔ <http://www.lua.org>



Knižnica

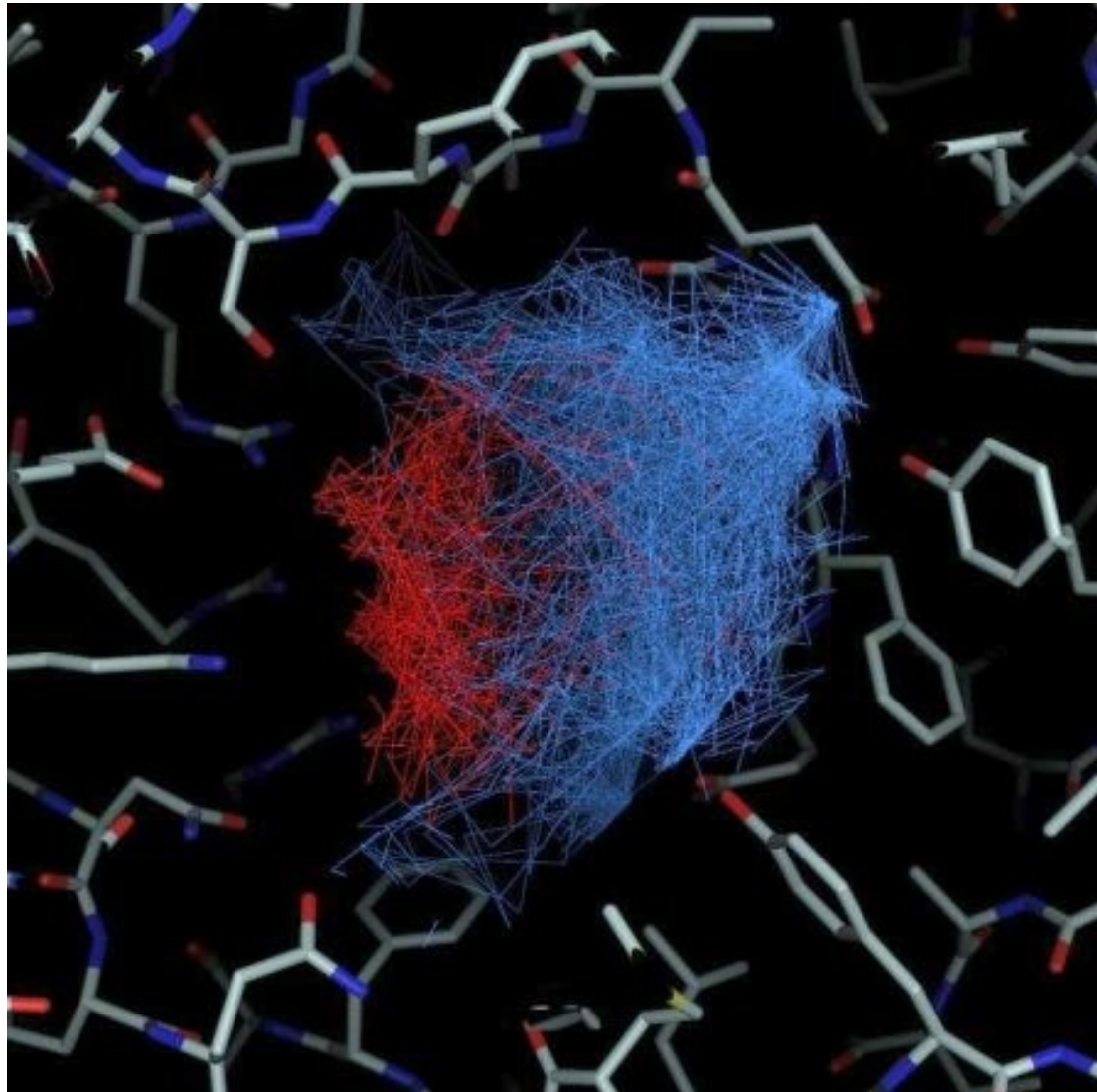
■ Mesa



Mesa

- <http://mesa3d.sourceforge.net>
- open source implementácia OpenGL
- vysoko-úrovňová knižnica
- grafické programy
- grafické servery a clustery

Dino



Games for Linux

- podpora hier pre Linux
- port a code, save the game
- archív 192 hier, 35 knižníc
- ➔ <http://games.linux.sk>



Blok 3.



Graphic User Interface
GUI

Graphic User Interface

- okná, tlačítka, zoznamy, menu
- Jednoduchá definícia rozhrania
- editor grafického rozhrania

Vhodné knihovny

QT

FLTK

GTK

QT



Code less.
Create more.
Deploy everywhere.

Troltech:
<http://www.troltech.com>

QT

- produkt firmy Trolltech
- licencie: open source aj komrčné
- knižnica vznikla v roku 1996
- podporuje C++ a Javu

Platformy

- **MS/Windows**
- **Unix/X11 - Linux, Sun Solaris, HP-UX, Compaq Tru64 UNIX, IBM AIX, IRIX, Mac OS X**
- **Embedded - Linux platforms with framebuffer support.**

Code less.

Create more.

Deploy everywhere.

C++
Java



Windows
Mac OS
Linux/X11
Windows CE
Embedded Linux
S60 (coming soon)

Moduly

- **Network Module**
- **OpenGL Module**
- **SQL Module**
- **Workspace (MDI) Module**
- **XML Module**
- **Canvas Module**
- **Table Module**

Designer

- nástroj na tvorbu GUI
- rozvrhnutú plochu ukladá do XML súboru
- z XML definície sa generuje kód programu



Kompilácia

- uloženie projektu
- vytvorenie Makefile z projektu
- `qmake -o Makefile test.pro`
- kompilácia
- `make`

Linguist

- nástroj na preklad aplikácii
- umožňuje oddeliť kód od textov
- prekladateľ a programátor pracujú samostatne

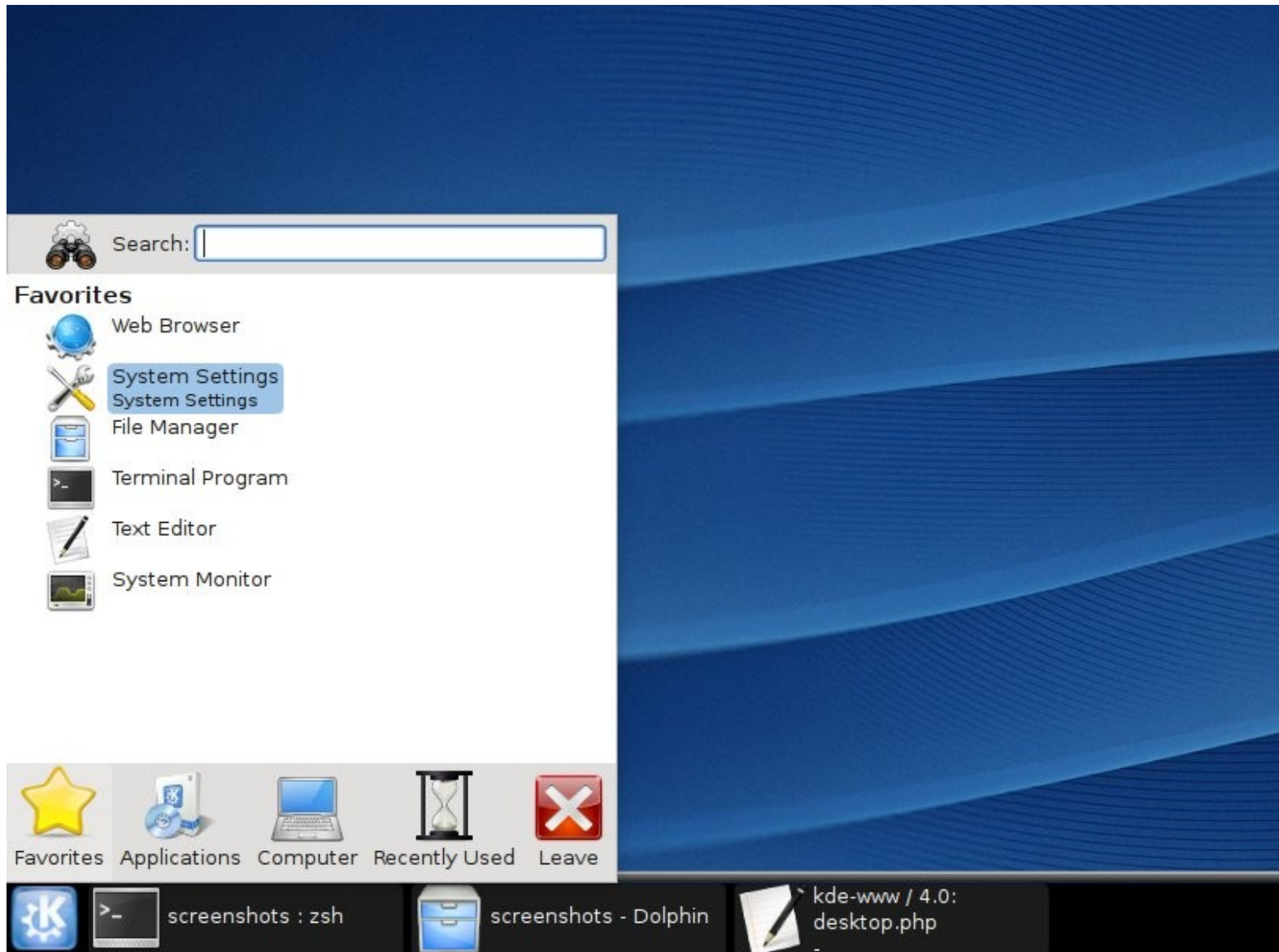
Assistant

- **prepracovaný systém nápovedy**
- **ukážky kódu k funkciám**
- **stručné a jasné návody**

Amélia



KDE



Google Earth



FLTK



<http://www.fltk.org>

Knižnica FLTK

- **The Fast Light Tool Kit**
- **Autor: Bill Spitzak**
- **venoval sa vývoju rozhraní pre Sun Microsystems**
- **veľa myšlienok zjednodušil a vložil do kitu**

Platformy

- Unix, Linux
- Microsoft Windows
- MacOS
- s podporou OpenGL a GLUT

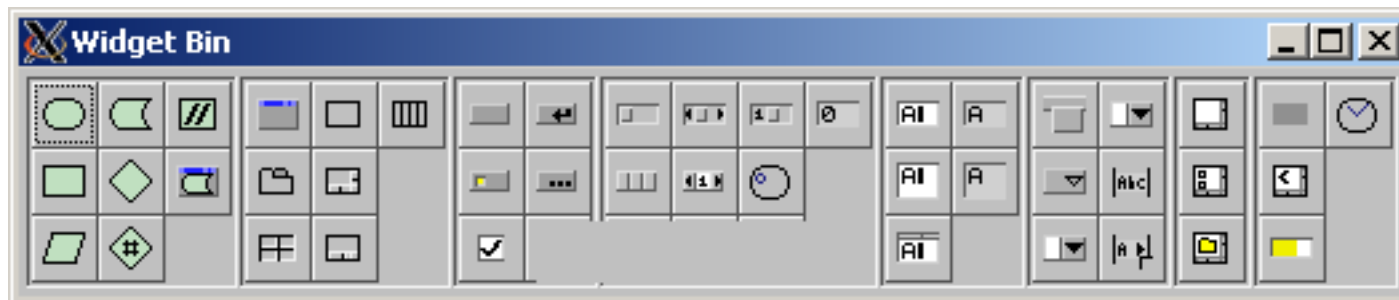


Vlastnosti

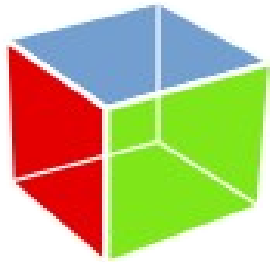
- **kompletný objektový návrh**
- **optimalizované na statické kompilácie**
- **Nízko-úrovňová kompatibilita - 10% odlišnosti**
- **Emacsové skratky pre ikony**

FLUID

- **Fast Light User-Interface Designer**
- **jednoduchý designer na tvorbu GUI**



Knižnica



The GTK+ Project

➔ <http://www.gtk.org>

GTK I.

- **GIMP ToolKit**
- **najbežnejšie používaná knižnica pre GUI**
- **pôvodne vznikla ako súčasť GIMPu**
- **je na nej postavené obrovské množstvo projektov**

GTK II.

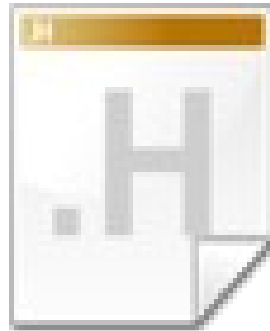
- **Peter Mattis, Spencer Kimball, Josh MacDonald**
- **GNU Network Object Model Environment (GNOME)**
- **C++, Guile, Perl, Python, TOM, Ada95,**
- **Objective C, Free Pascal, Eiffel**
- **komplexne riešená knižnica**

Glade

- GUI editor pre GTK+
- Glade 3
- ➔ <http://glade.gnome.org>



Blok 4.



Klient – server

Klient-server

Boost

ACE

JSON, XML

Apache Axis2C

Boost

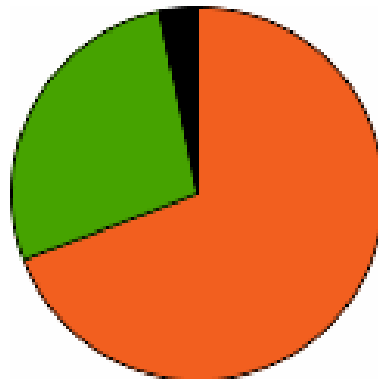


- <http://www.boost.org>
- “...one of the most highly regarded and expertly designed C++ library projects in the world.”
- ➔ Herb Sutter and Andrei Alexandrescu

Boost - projekt

LANGUAGES

C++	63%
HTML	25%
Other	3%



Boost C++ Libraries, updated 29 Aug 2008

more at [chloh](#)

PROJECT COST

This calculator estimates how much it would cost to hire a team to write this project from scratch.

Include	<input type="text" value="Markup And Code"/>
Codebase	5,264,026 LOC
Effort (est.)	1567 Person Years
Avg. Salary	<input type="text" value="\$55000"/> /year
	\$86,183,655

Boost C++ Libraries, updated 29 Aug 2008 more at [chloh](#)

Jam



- náhrada Makefile
- rozšíriteľné funkcie
- automatická detekcia header files
- prenositeľné: Unix, Windows, VMS, Mac
- malý CPU overhead
- build obrovských projektov
- škálovateľné
- (pozn. citlivé na medzery)

Boost Test Library

- Test everything that could possibly break
- jednoduchá tvorba tescasov
- TestSuite – TestCase
- test je spustiteľný program
- bjam integruje spúšťanie testov



TestCase

```
using namespace boost::unit_test_framework;
```

```
/**  
 * Test basic marshaller functionality  
 */  
void testServerQuery()  
{  
    ServerQuery query = ServerQuery();  
  
    query.session = "112";  
    query.userId = "janko";  
  
    BOOST_CHECK_EQUAL( query.sessionAsChar(), "112" );  
    BOOST_CHECK_EQUAL( query.userIdAsChar(), "janko" );  
}
```



jam + TestCase

■ Jamfile (Jamroot)

➔ `import testing ;`

➔ `run tests/testServerQuery.cpp server ;`

■ Shell

➔ `$ bjam`

➔ `...patience...`

➔ `...found 2 targets...`

➔ `...updating 2 targets...`

➔ `*** test passed ***`

Návrhové vzory a C++

- POSA - <http://posa3.org/>
- Douglas C. Schmidt
- ➔ <http://www.cs.wustl.edu/~schmidt/>
- Vanderbilt University

ACE

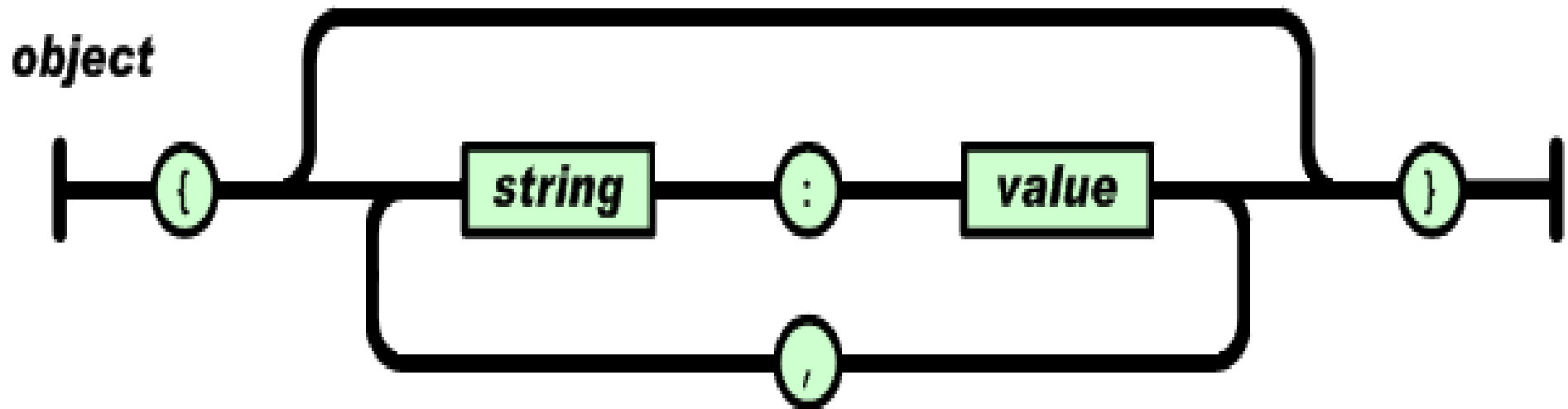
- **The ADAPTIVE Communication Environment**
- <http://www.cs.wustl.edu/~schmidt/ACE.html>
- **Open source C++ framework**
- **demultiplexing**
- **reactor, proactor**
- **serializácia**
- **acceptor, connector**
- **stream**



JSON



- JavaScript Object Notation
- <http://www.json.org/>



JSON – příklad

```
{  
  "firstName": "Janko",  
  "lastName": "Hraško",  
  "address": {  
    "streetAddress": "Botanická 42",  
    "city": "Brno",  
    "postalCode": 602 00  
  },  
  "phoneNumbers": [  
    "0118 999 881 999 119 725 3",  
    "999"  
  ]  
}
```

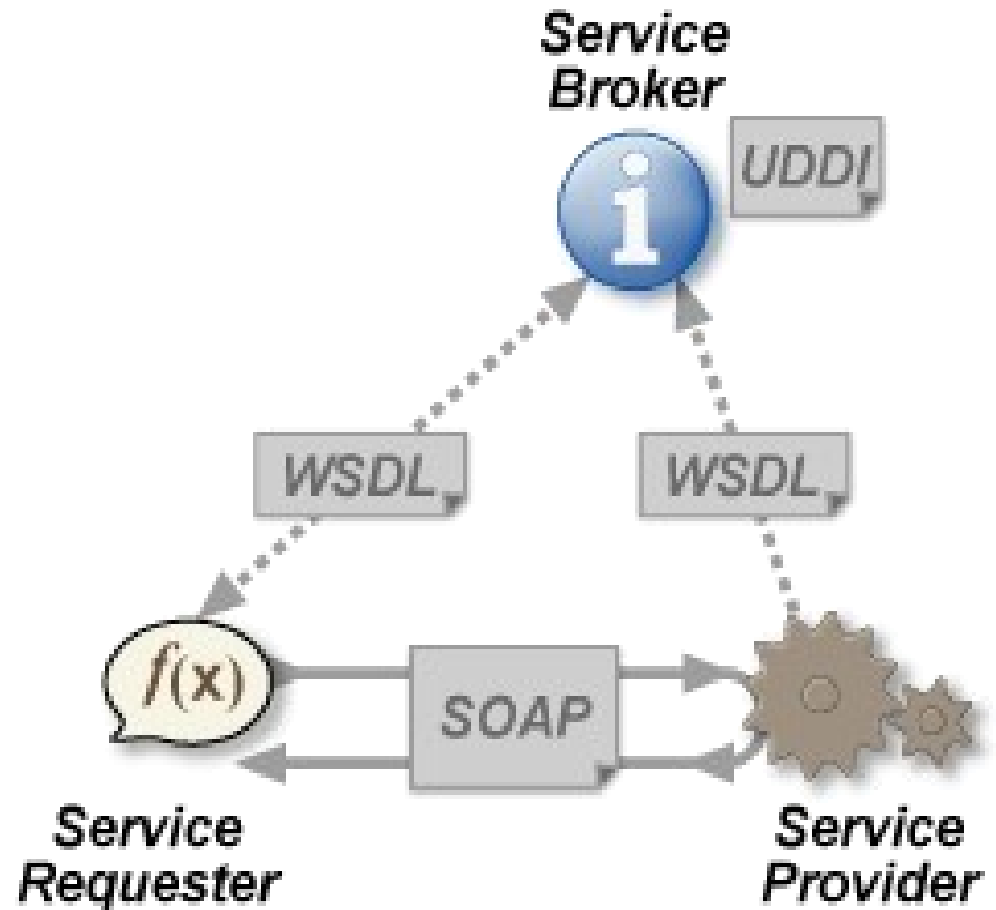

XML

- **libxml2**
- **MIT licencia**
- <http://xmlsoft.org>
- **C++ binding pre libxml2**
- <http://libxmlplusplus.sourceforge.net/>



Web Services

- definované W3C
- Web API
- protokol SOAP
- XML Based



Axis C++

- Apache Web Service Project
- obe strany: klient, server
- Implementácie C, C++, Java
- verzia pre C++
- ➔ <http://ws.apache.org/axis/cpp/index.html>



Blok 5.



**Chut'ovky a
programátorské nástroje**

Eclipse

- výkonné multiplatformové IDE
- ➔ <http://www.eclipse.org>



Eclipse Debugger

- využíva GDB
- debug projekty
- ➔ spustenie C++ aplikácie
- ➔ pripojenie k bežiacemu procesu
- ➔ post-mortem analýza (core dump)
- pozn.: meno projektu nesmie obsahovať medzeru

Subversion - SVN

- verzovací nástroj
- jednoduchý na používanie
- <http://subversion.tigris.org>
- repository, check-out, commit, merge
- <http://www.asinus.org/wiki/VideoTutorialSubversion>



S U B V E R S I O N

Eclipse - vývoj

- C/C++ - CDT plugin

- ➔ <http://www.eclipse.org/cdt/>

- Subclipse plugin - SVN

- ➔ <http://subclipse.tigris.org>

- PyDev

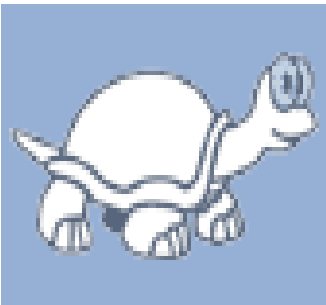
- ➔ <http://pydev.sf.net>



Tortoise SVN

■ SVN klient pre Windows

➔ <http://tortoisesvn.tigris.org>



TortoiseSVN

Trac

- **Source Control & Project Management**
- ➔ <http://trac.edgewall.org>
- **integruje: Wiki, Ticket system, SVN**
- **množstvo plug-inov**
- ➔ <http://trac-hacks.org/>



Projekt

- **Google Code**

- ➔ <http://code.google.com/opensource>

- **SourceForge**

- ➔ <http://sourceforge.net>



- **FreshMeat**

- ➔ <http://freshmeat.net>



- **otvorit' program svetu - Open Source**

- **niekto môže vo vašej práci pokračovať**

- ➔ <http://atanks.sf.net>

Ohloh

- porovnanie projektov
- rôzne metriky
- ➔ <http://www.ohloh.net>



Include	<input type="text" value="Markup And Code"/>
Codebase	385,818
Effort (est.)	102 Person Years
Avg. Salary	\$ <input type="text" value="55000"/> year
\$ 5,598,652	



Platformy

- Minimalist GNU For Windows

- ➔ <http://www.mingw.org>

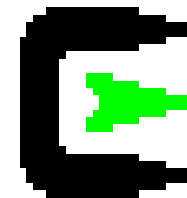
- DJGPP od DJ Delorie

- ➔ <http://www.delorie.com>

- Cygwin

- ➔ <http://www.cygwin.com>

- ➔ <http://cygwinports.dotsrc.org/>



Pomocné nástroje

- **formátovanie kódu**
 - ➔ `indent`
- **záplaty na kód**
 - ➔ `diff`, `patch`
- **ladenie**
 - ➔ `strace`, `gdb`, `cgdb`

Doxygen

- generovanie dokumentácie pre
 - ➔ C, C++, Java, Python
 - ➔ PHP, C#, D
- použitie
 - ➔ `doxygen -g project.config`
 - ➔ `doxygen project.config`



Asinus IT Group

- podpora open source projektov
- hostovanie projektov
- technológie:
 - + Python, Ruby, Java, SVN, Trac
- kontakt: team@asinus.org
- napr.: FlexGarden <http://www.flexgarden.net>

ASINUS

C#, .NET a Mono

- open source implementácia .NET
- Rozhranie napr. GTK#
- ➔ <http://www.mono-project.com>





Python

- silný objektovo orientovaný jazyk
- platformovo nezávislý
- integrovateľný s C a C++
- komunikácia s OpenOffice.org
- aplikačný server Zope, Django
- ➔ <http://www.python.org>
- ➔ <http://www.py.cz>



Twisted

- udalost'ami riadený sieťový framework
- podpora TCP, UDP, SSL/TLS, multicast
- Unix sockets
- veľké množstvo protokolov
- ➔ HTTP, NNTP, IMAP, SSH, IRC, FTP...
- ➔ <http://twistedmatrix.com/trac/>



PyGame

- kombinácia sily Pythonu
- a rýchlosti SDL (kód v C)
- ➔ <http://www.pygame.org>



OpenOffice.org

- UNO bridge – Java, C++, Python, Perl
- rozhranie na komunikáciu s OpenOffice.org
- generovanie dokumentov
- dopĺňanie údajov z databáz
- ➔ <http://udk.openoffice.org>



se-radio.net

- Software engineering radio
- zaujímavý a kvalitný podcast
- ➔ <http://www.se-radio.net>



A je tu záver

Otázky?

Myšlienka

Existuje mnoho ciest ako vytvoriť program.

Zvoľte si tú, ktorá vás naučí najviac.

Ďakujem za pozornosť

...0100100100100..01010...
...0100100100100..01010...2
...0100100100100..01010...

FI MUNI, Brno 2008
Juraj Michálek

e-mail: michalek@atollon.com
URL: <http://georgik.asinus.org>
jabber: [georgik@jabber.e-region.net](jabber:georgik@jabber.e-region.net)

