

# Vývoj v C++

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

**FI MUNI, Brno 2007**  
**Juraj Michálek**

e-mail: [michalek@atollon.com](mailto:michalek@atollon.com)  
blog: <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

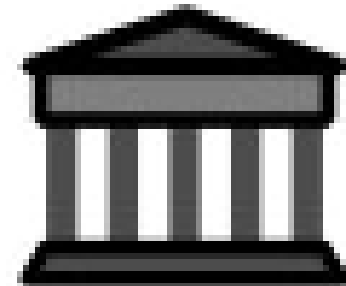
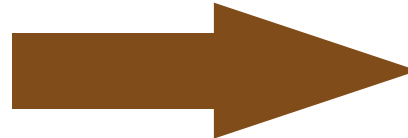
## ■ Blok 4. - server-klient

- JSON, XML, Webservice - SOAP

## ■ 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**

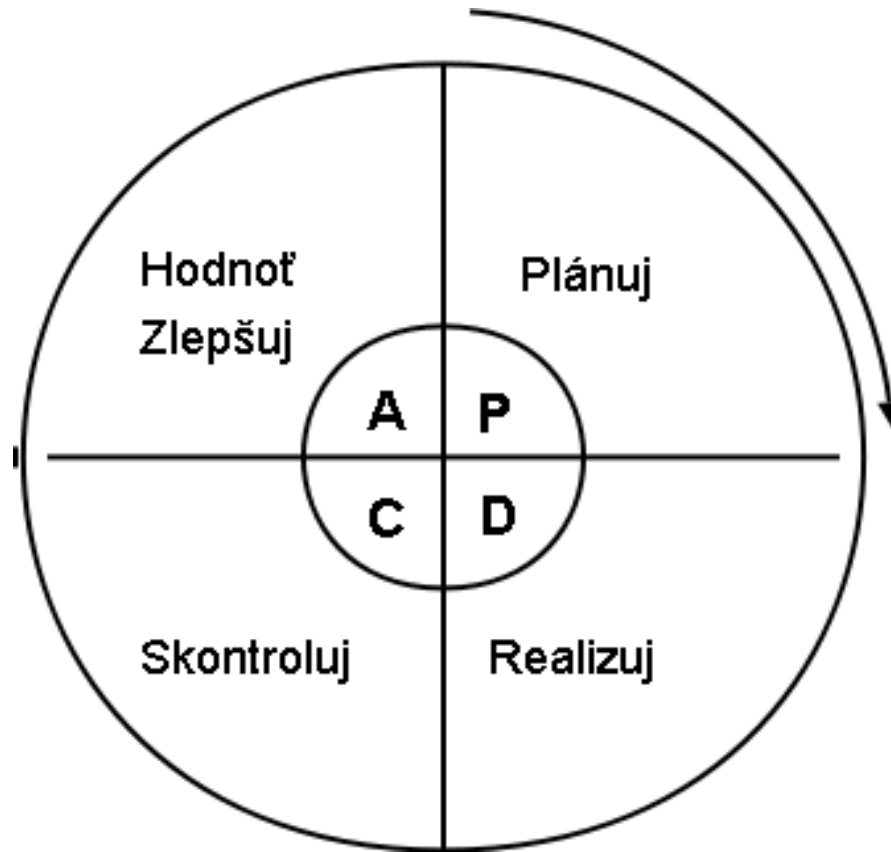
# Na vlastných nohách

- 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.

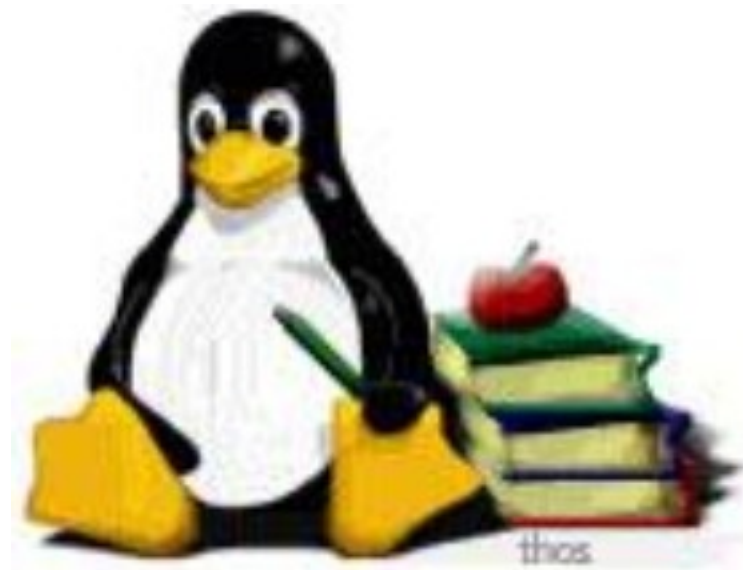
# Modely vývoje

- **Agilné metody**
- **Scrum**
- **Extreme Programming**

# Deming wheel



# Štandardné knižnice





# Štandardné knižnice I.

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

# Štandardné knižnice II.

- operácie s asociatívnymi poliami
  - ➔ `#include <map>`
- špeciálne algoritmy (triedenie, výpis)
  - ➔ `#include <algorithm>`
- zhrnutie: umožňujú urýchlenie vývoja

# Otázka

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

# Odpooved'



# Spracovanie vstupu

- **stlačil používateľ**
  - + **aípku hore a vpravo?**
  - + **CTRL+ALT+DEL+Any key+Benykey?**
- **v akej súradnicovej sústave pracuje myš?**
- **má myš aj koliesko? v akej polohe?**
- **(nechýba používateľovi koliesko ;-)**

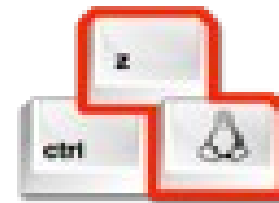
# Spracovanie výstupu

- vykresliť obraz v 16-bit alebo 32-bit?
- ktorú časť obrazu prekresliť?
- ako nakresliť kurzor, aby po ňom nezostávali stopy?
- aký grafický akcelerátor je k dispozícii?

# Ťažký je život ;-)

- všetko sa predsa naprogramovať nedá!
- veľa programátorov riešilo podobné problémy
- a preto vzniklo niekoľko kvalitných knižníc

# 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

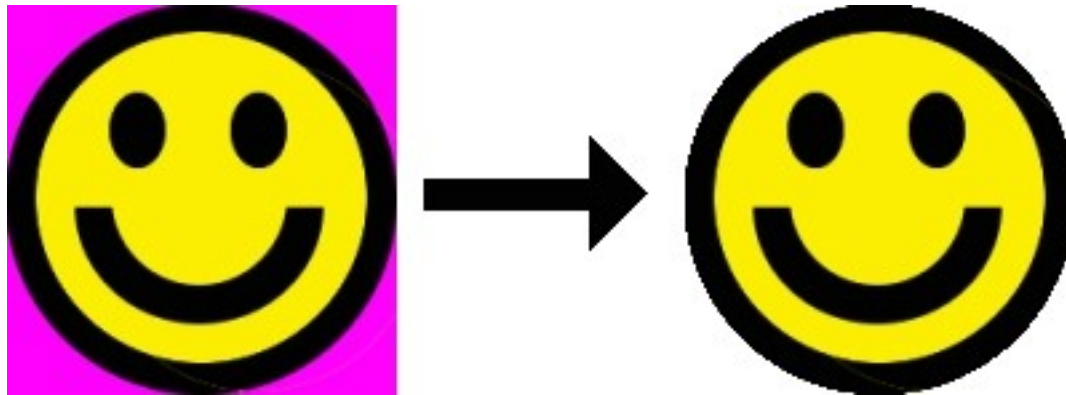
- ➔ `BITMAP *load_bitmap(filename,  
palette);`

- 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 zaisťuje 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



# Kni~nica



# 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 obrazu 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 použiteľné**
- **SDL\_mixer - multikanálový mixér**
- **SDL\_image - spracovanie obrazových 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

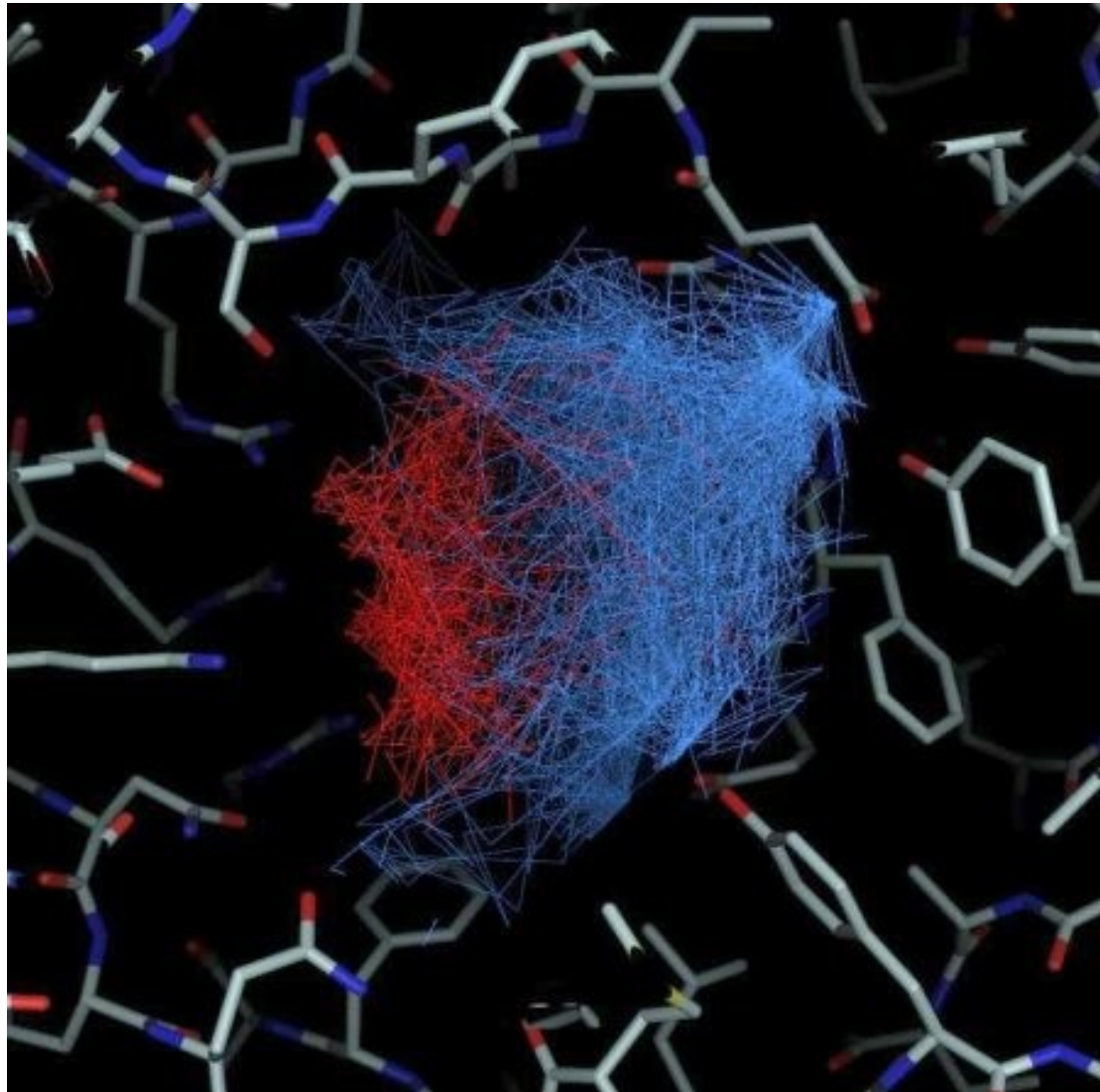
## ■ Mesa



# Mesa

- **open source implementácia OpenGL**
- **vysoko-úrovňová knižnica**
- **grafické programy**
- **grafické servery a clustery**

# Dino



# Zdroj

■ **vývoj**

➔ <http://mesa3d.sourceforge.net>



# Games for Linux

- podpora hier pre Linux
- port a code, save the game
- archív 184 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



# QT

- **produkt firmy Trolltech**
- **bezplatné použitie pre nekomerčné využitie**
- **knižnica vznikla v roku 1996**
- **je postavená na C++**

# Platformy

- **MS/Windows - 95, 98, NT 4.0, ME, 2000, XP**
- **Unix/X11 - Linux, Sun Solaris, HP-UX, Compaq Tru64 UNIX, IBM AIX**
- **IRIX and a wide range of others**
- **Macintosh - Mac OS X**
- **Embedded - Linux platforms with framebuffer support.**

# 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

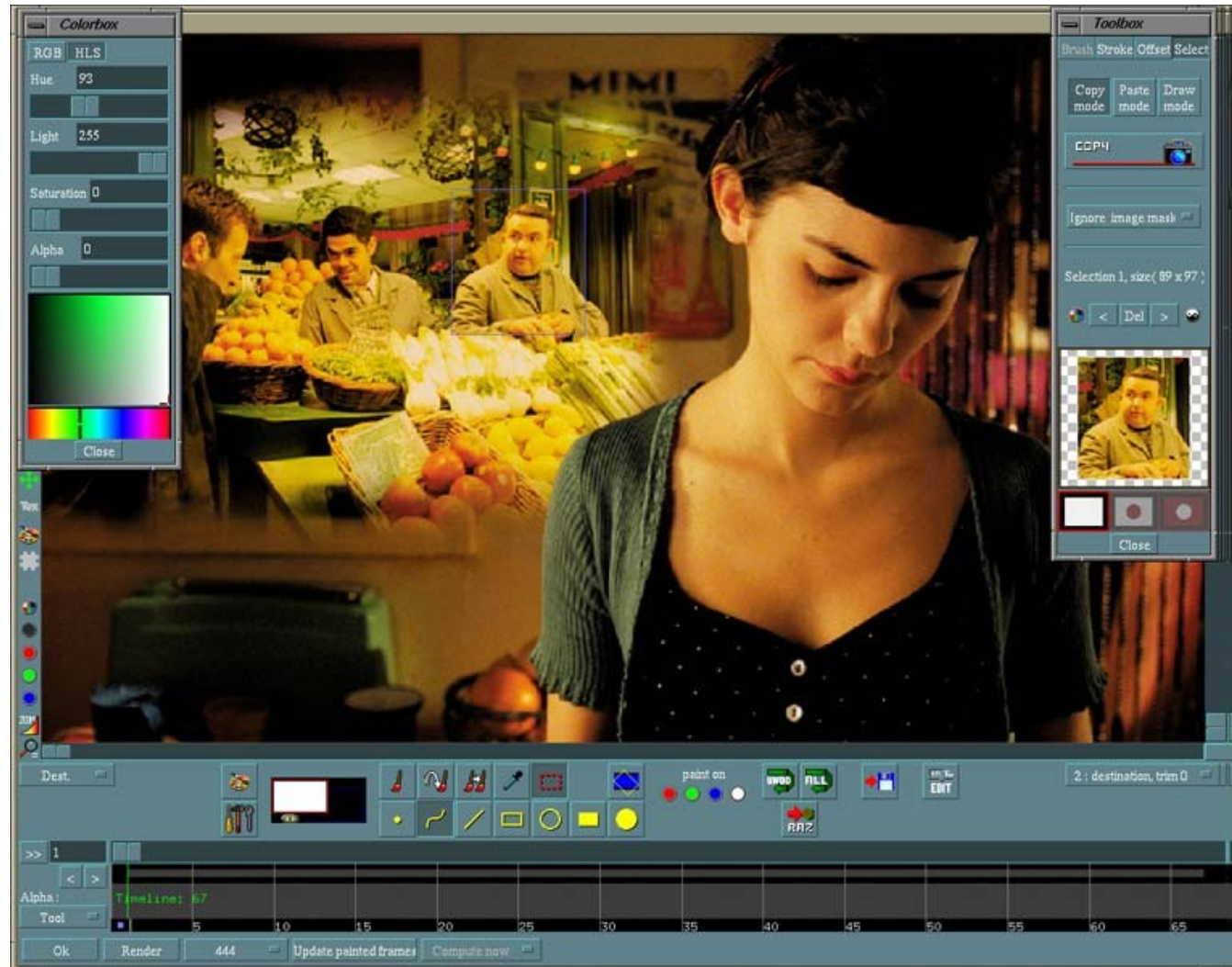
- **prepracovaný systém nápovedy**
- **ukážky kódu k funkciám**
- **stručné a jasné návody**

# Zdroj

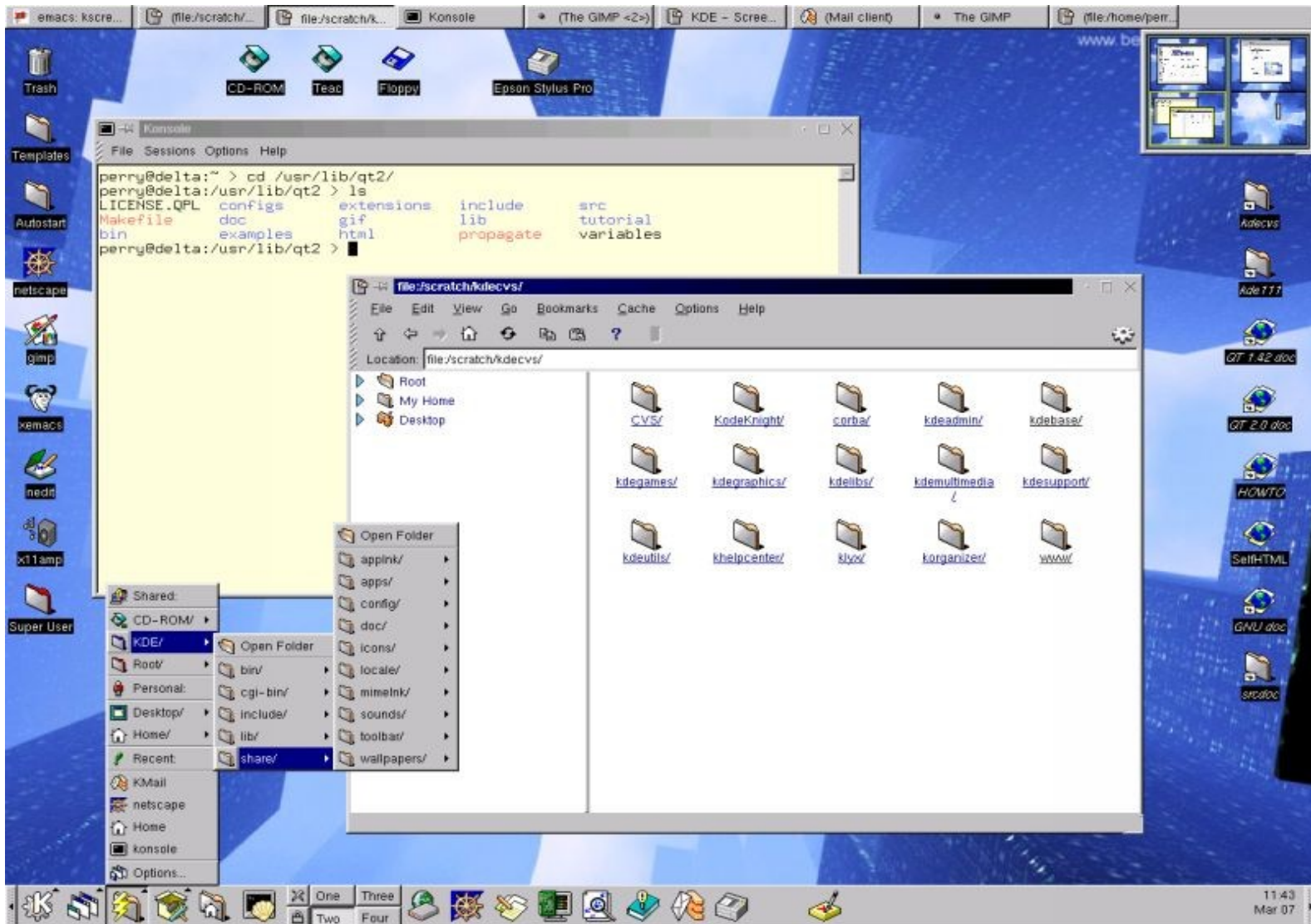
■ **Trolltech:**

➔ <http://www.trolltech.com>

# Amélia



# KDE





# Qtopia





**FLTK**



# Knižnica FLTK

- **The Fast Light Tool Kit**
- **Autor: Bill Spitzak**
- **venoval sa vývoju rozhraní pre Sun Microsystems**
- **veľa myšlienok zjednodušil a vložil do kitu**
- **o projekt sa momentálne stará vývojový tím FLTK**

# Platformy

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

# Vlastnosti

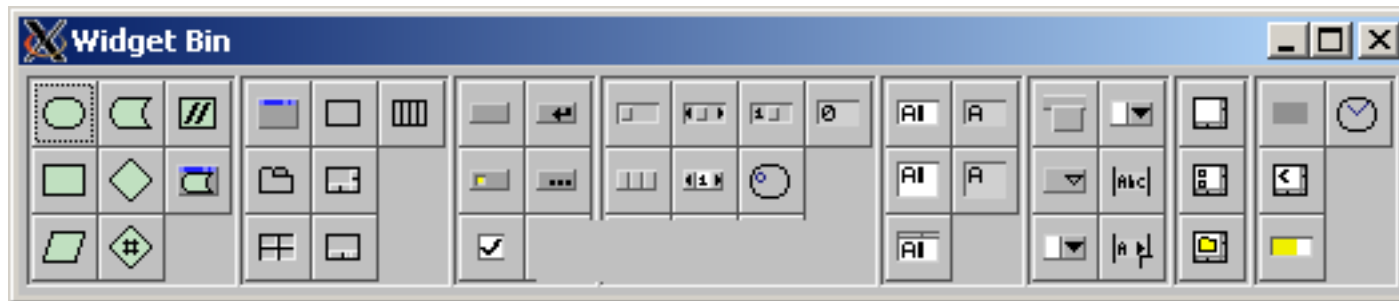
- **kompletný objektový návrh**
- **optimalizované na statické kompilácie**
- **Nízko-úrovňová kompatibilita - 10% odlišnosti**
- **Emacsové skratky pre ikony**

# Spracovanie udalostí

- pohyb myšou
- zmena stavu tlačítka
- zachytávanie klávesnice

# FLUID

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



# Zdroj

■ **vývoj**

➔ <http://www.fltk.org>

# Knižnica





# 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**

# Zdroj

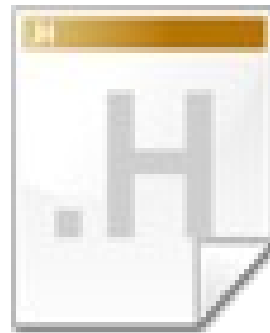
- **GTK vs. GTK+**
- **vývoj**
- ➔ <http://www.gtk.org>

# Glade

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

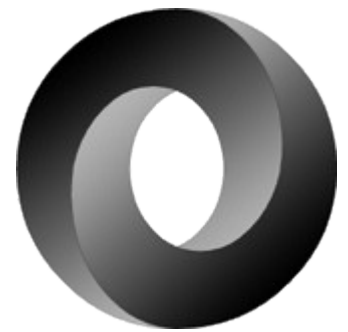


# Blok 4.

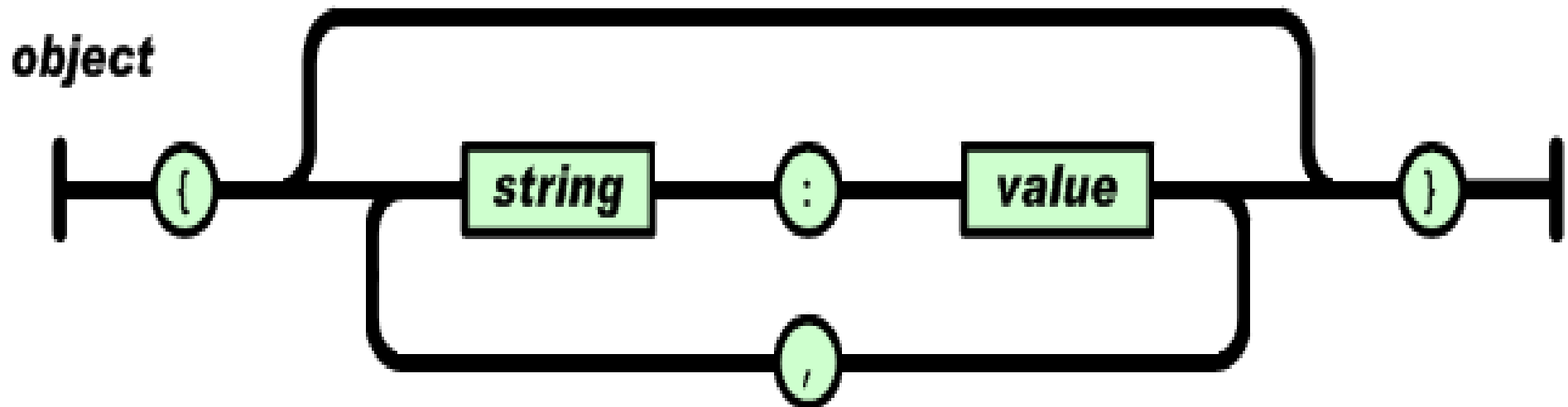


**Klient – server**

# 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>



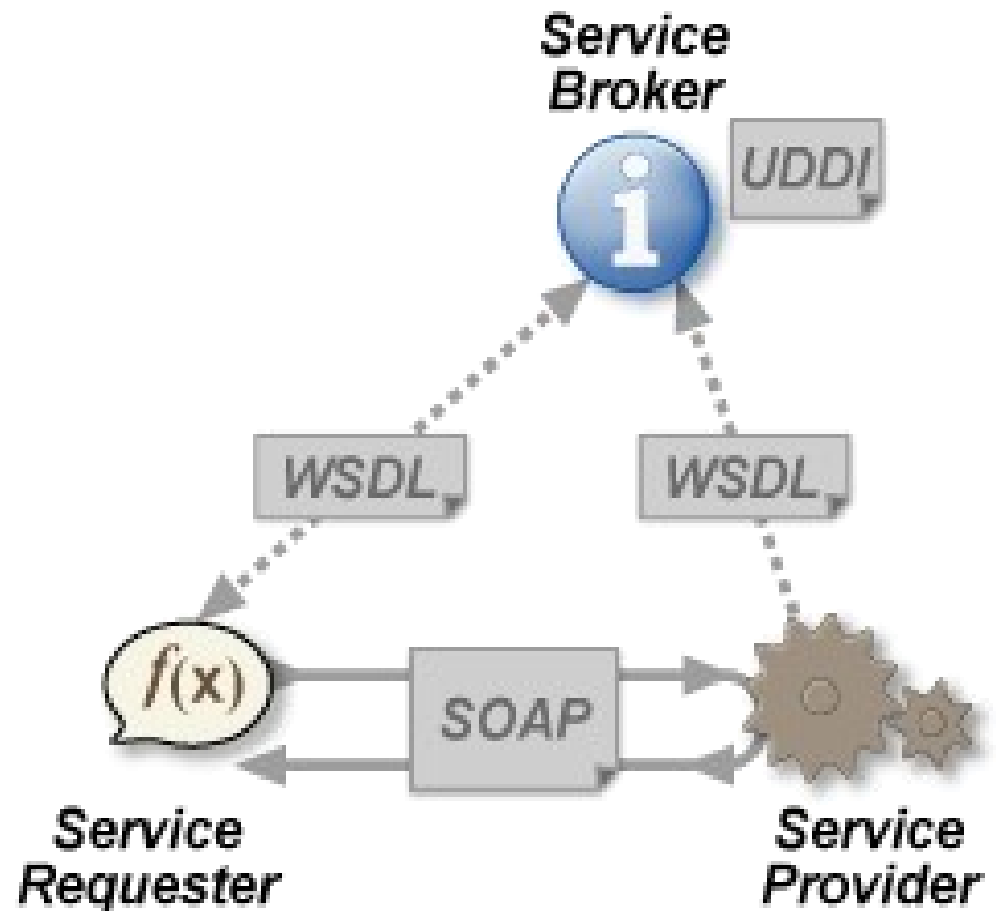


# libxml++

- C++ binding pre libxml2
- <http://libxmlplusplus.sourceforge.net/>

# Web Services

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

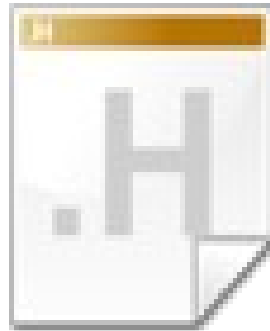


# Axis C++

- 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**

# Projekt

- **SourceForge**

- <http://sourceforge.net>



- **FreshMeat**

- <http://freshmeat.net>



- **otvoriť program svetu - Open Source**

- **niekto môže vo vašej práci pokračovať**

- <http://atanks.sf.net>

- **Dôležitá je kritika verejnosti**

# Platformy

- **Minimalist GNU For Windows**

- ➔ <http://www.mingw.org>

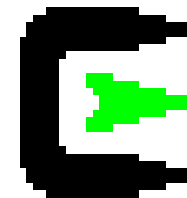
- **DJGPP od DJ Delorie**

- ➔ <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`





# Subversion - SVN

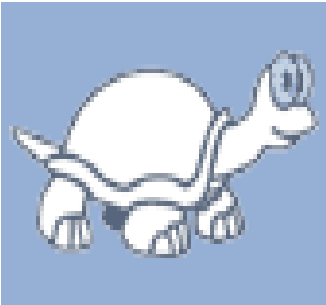
- verzovací nástroj
- jednoduchý na používanie
- <http://subversion.tigris.org>
- repository, check-out, commit, merge
- <http://www.asinus.org/wiki/VideoTutorialSubversion>



S U B V E R S I O N

# 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/>



# Asinus IT Group

- podpora open source projektov
- hostovanie projektov
- technológie:
  - Python, Ruby, Java, SVN, Trac
- kontakt: [team@asinus.org](mailto: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>



# Eclipse

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



# Eclipse - vývoj

- **C/C++ - CDT plugin**

- ➔ <http://www.eclipse.org/cdt/>

- **Subclipse plugin - SVN**

- ➔ <http://subclipse.tigris.org>

- **PyDev**

- ➔ <http://pydev.sf.net>





# 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.**

**Zvolte si tú, ktorá vás naučí najviac.**

# Ďakujem za pozornosť

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

**FI MUNI, Brno 2007**  
**Juraj Michálek**

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

