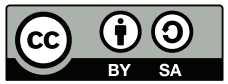
Open source programování

Otevřené vývojové prostředí

Petr Baudiš ⟨pasky@ucw.cz⟩

MFF UK 2011



Outline

Úžasný nový svět...

Otevřené vývojové prostředí

- Tvořeno programátory... hlavně pro programátory?
- Excelentní podpora pro systémové programátory
- Kolísavá kvalita dokumentace, ale přístup ke zdrojákům
- Menší důraz na IDE

O čem dnes

- C, C++ toolchain
- Základní knihovny
- Dokumentace
- Skriptovací jazyky
- Verzovací systémy

Outline

GNU Compiler Collection

- **gcc**, jeden z prvních a úhelných kamenů GNU
- Standardy — C89, C99, C++98, C(++)1x; Objective C, Fortran, Java, Ada

Rozšíření

- typeof, long long, `x ? : y`, `case 1 ... 5`, `0b1011`
- Atributy funkcí (aj.) — noinline, pure a const, constructor, atd.
- Atomické operace, thread-local proměnné, vektorové typy
- Inline assembler

Model překladu C

- Preprocessing: `gcc -E`
- `.c` → `.o`: Překlad, `gcc -c`
- `.o` → spustitelný soubor: Linkování, `gcc`
- `.c` → `.o`: Taky stačí `gcc`
- Jednoduché použití: `gcc -Wall -O3 -g -o soubor soubor.c`

- ELF: Univerzální formát binárních souborů (objekt, executable, core dump)

Tvorba knihoven (shared objects)

- Verzování: major.minor.patchlevel, API vs ABI, verzované symboly
- Viditelnost symbolů: `-fvisibility=hidden`, `__attribute__((visibility ("default")))`, version scripty
- Relokace: Vyhněte se externím proměnným, uvnitř knihovny používejte lokální definice
- Jednoduché použití: `gcc -Wall -O3 -g` **`-shared -fPIC`** `-o soubor.so soubor.c`

GNU Binutils

- Linker `ld`: Obvykle jen backend pro gcc; linker skripty
- Assembler as: Vyrábí `.c` z `.s` místo `.c`
  AT&T synaxe! (vs. nasm)
- Dumpery: `nm`, `objdump`, `readelf`

- Konkurence: elfutils

GNU Make

Automatický překlad, přestaví právě věci, které jsou potřeba

Makefile

```
OBJS=soubor1.o soubor2.o

all: program
program: $(OBJS)
  $(CC) $(LDFLAGS) -o $@ $^

clean:
  rm -f $(OBJS)
```

```
make clean
make
```

- Úskalí: Rekurzivní make, paralelní make, automatický dependency tracking

GNU Autotools

- M4: Univerzální makroprocesor
- GNU automake: "Zjednodušení" tvorby Makefiles
- GNU libtool: Portabilní výroba knihoven
- GNU autoconf: Automatická detekce přítomných featur a knihoven, compile-time konfigurace; `./configure`

- Buildovací návody knihoven: pkg-config
- Konkurence: `Makefile.lib` apod., cmake

GNU Gettext

- Jak na i18n, l10n?
- Internationalization: Systém charsetů a locales
- Localization: Překlad textové komunikace s uživatelem

- Ve zdrojáku anglický string obalený makrem _()
- Pro každý jazyk katalog zpráv s překlady
- Kostra automaticky generovaná ze zdrojáku
- Generuje separátní binární soubor,
  runtime lookup na základě `$LC_MESSAGES`
- Podpora pro kontext, plurály atd.

GNU Debugger

- **gdb** — hlavně C, ale i spousta dalších jazyků
- Spuštění + breakpoint nebo analýza coredumpu
- Trasování, vypisování hodnot atd.
- Watchpointy, podmíněné breakpointy, změny hodnot, ...
- Interface: Rozhovor, TUI, GUI (ddd)
- Jednoduché použití: `break` a `run` a `cont`, `bt` a `frame`, `print` a `display`, `next` a `step`, `list`
- Low-level: `disas`, `info reg` a `x`, `nexti` a `stepi`

- `ptrace()` syscall

IDE

- Eclipse
- RHIDE
- KDevelop
- EMACS nebo vim!

Outline

GNU libc

- **glibc** — C runtime (ne C++), POSIXové API a příbuzní
- Standardy — Cx9, POSIX.*, SysV/BSD
- Částečná koevoluce s libiberty a GNUlib
- Charsets a locales, gettext runtime, třídění a vyhledávání, matchování globů a regulárních výrazů, I/O nad streamy i deskriptory, soubory a sockety, terminály, signály a IPC, procesy, job control, syslog, name resolution, matematické funkce, datum a čas, control flow, dynamický linker, proměnné prostředí, charakteristiky systému, kryptografické funkce
- Multi-threading (pthreads: NPTL, (mrtvé) LinuxThreads)
- Zajímavé featurky: I/O (vektorové, asynchronní, mmapové, dyn. alokované, ...), dočasné soubory, `backtrace()`, NSS, customizace `printf`, rozšíření paměťového alokátoru, obstacks
- Často GNU rozšíření pro reentrantní verze; `strverscmp()`, hledej `_GNU_SOURCE`

Systémové knihovny

- libevent
- libnih
- GLib
- libucw

Terminálové knihovny

- Termcap a terminfo
- GNU Readline
- NCurses
- SLang

Omalovánkové knihovny

- SDL — "low-level" grafika, I/O, zvuk, ...
- Cairo — vektorová grafika, mnoho výstupů
- GTK — okénka Cčkově (event a callback)
- Qt — okénka C++kově (signal a slot), i non-GUI věci

Outline

. . . ostatních projektů

- Manuálové stránky (linux-manpages)
- GNU info (pinfo!)
- Web : –(
- Use the Source, Luke

Generování dokumentace

Docbook

- Dokumentace v (rozumném) XML formátu,
  export do spousty výstupních formátů (HTML, PDF, man, ...)
- Preprocesory (asciidoc, markdown, ...)

Doxygen

- Referenční programátorská dokumentace
- Z komentářů přímo v kódu
- Automatické cross-reference

Outline

Shell

- GNU bash, zsh, (dash)
- GNU coreutils
- POSIX (aktivní drive; `$POSIXLY_CORRECT`)
- Roztodivná rozšíření

Další

- Perl: There is more than one way to do it
- Python: There should be one — and preferably only one — obvious way to do it
- Scheme: Tradiční skriptovací jazyk GNU
- Tcl: Hordy zombies
- Lua, CLisp, Ruby, PHP, ...

- SWIG: Bindingy C funkcí do různých skriptovacích jazyků
- Naopak: Problematické, nutno ručně

- flex a bison — scanner a parser (generátor C kódu)

Outline

Tradiční

- RCS (a SCCS) — jednotlivé soubory
- CVS — síťové RCS, které umí dávkově zpracovávat celý adresářový strom
- Subversion — pořádný VCS/SCM, ale centralizovaný

Distribuované

Git

It's simplest to think of the state of your Git repository as a point in a high-dimensional "code-space", in which branches are represented as n-dimensional membranes, mapping the spatial loci of successive commits onto the projected manifold of each cloned repository. — `http://tartley.com/?p=1267`

- Git — nejrozšířenější(?), idiosynkratický, mocný
- Mercurial — přátelštější (možná)
- Bazaar — nejpřátelštější
- Fossil — vyšperkovaný, (zatím) nerozšířený

Děkuji za pozornost

Příště: Gitový tutorial (SU1!)