

CPAN, tvorba a šíření softwaru v jazyce Perl

Petr Pajas
Hejnice 2010

Služba CPAN

CPAN = Comprehensive Perl Archive Network

Slouží ke sdílení modulů (knihoven) a dalších nástrojů pro Perl

Webové rozhraní poskytuje

- úložiště (264 mirrorů po celém světě)
- vyhledávání
- stahování a prohlížení balíčků
- zobrazení dokumentace + možnost anotace
- bug tracker (rt.cpan.org)
- testování kvanty dobrovolníků na různých platformách
- hodnocení modulů (review, rating)
- diskuzní fórum
- on-line diff a grep na danou distribuci
- upload, informace pro vývojáře, správa jmenných prostorů

Instalace balíčků

Ručně: stažením z webu a aplikováním „svaté trojice“ (viz dále)

- problém: závislosti (bývá jich hodně)

Automaticky: pomocí nástroje cpan

- Při prvním spuštění se zeptá na několik otázek
 - výběr mirroru, vlajky pro "svatou trojici")
- Umí najít distribuce pro daný modul, a všechny závislosti, stáhnout je, rozbalit, zkompilovat, otestovat, nainstalovat
- Vše jedním příkazem, např.

```
cpan -i Module::Starter
```

- Samotné cpan spouští CPAN shell
- Při havárii instalace máte možnost instalaci dokončit ručně

Typická struktura balíčku

<code>blib/</code>	- dočasný cílový adresář při sestavování
<code>lib/</code>	- vaše knihovny
<code>inc/</code>	- pomocné knihovny (např. nutné pro instalaci)
<code>t/</code>	- self-testy
<code>Changes</code>	- sem zapisujete podstatné změny
<code>Makefile.PL</code>	- program pro vytvoření Makefilu
<code>Build.PL</code>	- program pro vytvoření Build skriptu
<code>MANIFEST</code>	- seznam souborů pro distribuci
<code>META.yml</code>	- generovaný soubor pro indexer CPANu
<code>README</code>	- stručný popis balíčku (vyžadováno CPANem)

„Svatá trojice“

Svatou trojici tvoří následující **čtyři** :-) příkazy:

- **Varianta A** - distribuce založené na `ExtUtils::MakeMaker` nebo `Module::Installer`
 - `perl Makefile.PL`
 - `make`
 - `make test`
 - `make install`
- **Varianta B** - distribuce založené na `Module::Build`
 - `perl Build.PL`
 - `./Build`
 - `./Build test`
 - `./Build install`

Varianta A je tradiční, B je v Perlu relativně nová, má ještě mouchy, pomalu se však začíná prosazovat.

Než začneme nový projekt

- Podívat na CPAN a web, zda někdo už nenapsal něco, co by šlo použít
- Rozmyslet si strukturu
 - třídy (moduly)
 - samostatné distribuční balíčky
- Zvolit vhodný prefix jmen modulů
 - inspirovat se na CPANu
- V případě pochybností je lepší se zeptat
 - `modules@perl.org`
- Přečíst si doc `perlstyle`

Vývojový cyklus

- Vytvoření skeletonu
 - rozhraní k C knihovně na základě hlavičkových souborů => h2xs případně swig
 - jinak module-starter (viz dále)
- Úprava skeletonu
- Tvorba testů (adresář t/)
- Vývoj kódu (programování, ladění, další testy, však víte)
- Příprava distribučního balíčku
 - perl Makefile.PL (nebo ./Build.PL a dále ./Build místo make)
 - make manifest disttest distsign dist
- upload na CPAN
- odezva CPAN indexeru, dobrovolných testerů, a časem i uživatelů
- Bug tracker: rt.cpan.org

Module::Starter

- Příklad použití:

```
module-starter --module=Muj::Modul \
               --author="Petr Pajas" \
               --email="pajas@matfyz.cz" \
               --license=perl
```

- umí vytvořit skeleton pro všechny běžné sestavovací moduly:
--builder=ExtUtils::MakeMaker|Module::Install|Module::Build
- vytvoří rozumnou kostru budoucí distribuce
 - lib/Muj/Modul/* .pm, t/..., README, kostru dokumentace v POD
- je konfigurovatelný (~/.module-starter)
- rozšiřitelný o pluginy (např. Module::Starter::{Smart,PBP})
- do skeletonu lze snadno ručně migrovat i existující projekty

Makefile.PL (ExtUtils::MakeMaker)

```
use strict;  
use warnings;  
use ExtUtils::MakeMaker;
```

```
WriteMakefile(  
    NAME          => 'Audio::aKodePlayer',  
    AUTHOR        => q{Petr Pajas <pajas@matfyz.cz>},  
    VERSION_FROM  => 'lib/Audio/aKodePlayer.pm',  
    ABSTRACT_FROM => 'lib/Audio/aKodePlayer.pm',  
    LICENSE       => 'perl',  
    BUILD_REQUIRES => {  
        'Test::More' => 0,  
    },  
    dist          => { COMPRESS => 'gzip -9f', SUFFIX => 'gz', },  
    clean         => { FILES => 'Audio-aKodePlayer-*' },  
);
```

Makefile.PL (Module::Install)

```
use inc::Module::Install;

name      'Audio-aKodePlayer';
all_from  'lib/Audio/aKodePlayer.pm';
author    q{Petr Pajas <pajas@matfyz.cz>};
license   'perl';

build_requires 'Test::More';

auto_install;

WriteAll;
```

Build.PL (Module::Build)

```
use strict;
use warnings;
use Module::Build;

my $builder = Module::Build->new(
    module_name      => 'Audio::aKodePlayer',
    license          => 'perl',
    dist_author      => q{Petr Pajas <pajas@matfyz.cz>},
    dist_version_from => 'lib/Audio/aKodePlayer.pm',
    build_requires => {
        'Test::More' => 0,
    },
    add_to_cleanup    => [ 'Audio-aKodePlayer-*' ],
    create_makefile_pl => 'traditional',
);

$builder->create_build_script();
```

Tvorba testů

- užívá se testovacích modulů `Test`, `Test::More`, apod.
- plán testů (abychom věděli, že vše doběhlo)
 - `use Test::More tests => 23;`
- volání funkcí vaší knihovny a kontrola výstupů pomocí funkcí testovacího modulu, např.

```
BEGIN { use_ok( 'Some::Module' ); }  
ok($got eq $expected, $test_name);  
is ($got, $expected, $test_name);  
is_deeply($got_complex_structure,  
          $expected_complex_structure, $test_name);  
SKIP: {  
    skip $why, $how_many unless $missing_feature;  
    ok(..); is(...);  
}
```

- samotné testování pomocí `make test` (resp. `./Build test`)

Upload na CPAN

- <http://pause.cpan.org>
- založení účtu na CPANu (zdarma)
- přečtěte si pravidla na pause.cpan.org
- předpokládá se znalost `perlmodinstall`, `perlmodlib`, `perlmodstyle`, and `perlnewmod`
- upload z webové stránky
- server PAUSE si soubor
 - převezme a zkontroluje
 - prohledá (vyhledá jména balíčků), zaindexuje
 - zveřejní
 - během několika hodin jej převezmou i mirrory
 - o převzetí a stavu indexace vám zašle e-mail

Různé

- `Module::Starter::PBP` - podle mne nic nepřináší
- `Test::Perl::Critic` - dobrá věc, pokud souhlasíte s PBP
- `ExtUtils::XSpp` – `xsp` je varianta `h2xs` pro C++
- `Devel::CheckLib` – zjistí, zda v systému existuje daná C knihovna a hlavičky
- `cpanp` z modulu CPANPLUS
 - varianta `cpan` shellu
 - komfortnější (zavolá za vás `sudo`, snazší konfigurace, atp.)
 - výrazně pomalejší (netuším proč)
- Vývoj pro Windows
 - ActivePerl fy ActiveState používá tzv. Perl Package Manager a binární, předkompilované balíčky
 - zazipovaný `blib/plus` metadatový XML souboru `.ppd`
 - vytvoří se pomocí `make ppd,zazipovat blib/` ale musíte už sami

Různé II

- Developer releases: číslo verze tvaru X.YY_ZZ
- Dokumentace k modulu – POD
 - automaticky se vygeneruje man
 - CPAN zobrazí pěkné HTML
 - ActivePerl zobrazí HTML help
- Co když modul požaduje velká data (model)
 - raději neukládat na CPAN, autoři to nevidějí rádi
 - URL uvést v README, v dokumentaci
 - naučit Makefile.PL/Build.PL data stáhnout pomocí LWP
 - kam s nimi nechat raději na uživateli
 - konfigurace cesty pomocí .pm souboru a ENV

CPAN testers

- CPAN testers
- zkouší jen sestavit a provést testy
- reporty vám chodí průběžně e-mailem
- Jak se jím stát:
 - Spustíte cpan shell

```
install CPAN::Reporter  
o conf init test_report
```
 - To je vše – chyby při instalaci modulů se nyní budou automaticky hlásit autorům

To je vše.
Děkuji.