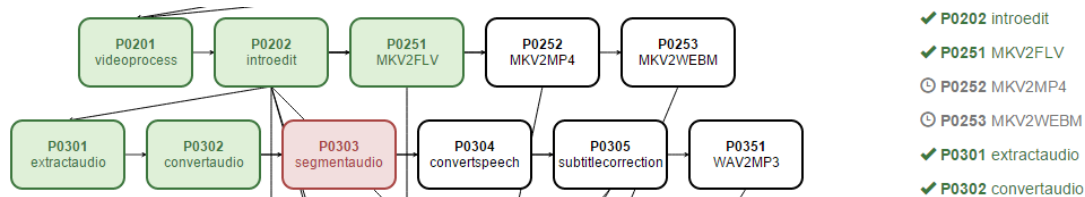


Prednasky.com – propojení s video servery FIT

Bc. Pavel Černý*



Abstrakt

Cílem této práce je vytvoření komplexního systému pro automatizované zpracování videí z přednášek. Práce je rozdělena na dvě hlavní podúlohy. První je věnována serverové části – Bash frameworku, který je tvořen z několika oddělených atomických procesů. Vstupem tohoto bloku bude záznam přednášky a výstupem bude video s úvodní částí, titulky a synchronizovanou prezentací. Druhou důležitou částí je internetová aplikace, která bude sloužit pro přehrávání, editaci, vytváření procesů a administraci.

Klíčová slova: Bash framework — Atomické úlohy — Přednášky — Video server — Zpracování videa — Tokenové zpracování — Responzivní design

Příložené materiály: —

*xcerny49@stud.fit.vutbr.cz, Faculty of Information Technology, Brno University of Technology

1. Úvod

Tento projekt se skládá ze dvou částí – zpracování videí na straně serveru a z webové aplikace. To je také jeden z důvodů proč jsem si jej vybral. S tvorbou webových aplikací mám již dřívější zkušenosti a hlavně mě tato práce baví. Serverová část byla pro mě taktéž velmi atraktivní kvůli samotnému problému úlohy – komplexně a zároveň uživatelsky přívětivě nastavit a zpracovat videa, což byla určitá výzva.

Záznamy z přednášek se aktuálně zpracovávají tím způsobem, že jsou publikovány na video serveru FIT, kde si je může student pouze stáhnout – bez jakýchkoli dalších dat. K záznamu je pouze na začátek přidán obrázek, kde je upozornění o kopírování a dále je provedena konverze do správného formátu. Vždy tedy bylo nutné, si záznam před zhlédnutím stáhnout.

Zpracování záznamu přednášek je velmi variabilní – můžeme mít jednoduché video, které se pouze překonvertuje do požadovaného formátu, ale v opačném případě je nutné přidat intro, detekovat plátno, syn-

chronizovat prezentaci a to vše nastavitelné s nebo bez kontroly správce.

Základní myšlenka je taková, že na serveru jsou spuštěny programy ve smyčce, které zpracovávají jednotlivé kroky při kompletování videa. Velký důraz byl kladen na počáteční požadavky:

- atomicita jednotlivých procesů,
- persistence,
- přenositelnost,
- rozšiřitelnost,
- bezpečnost,
- konfigurace.

Jako vhodné řešení jsem zvolil vytvoření Bash frameworku a to z toho důvodu, že většina použitých programů jsou unixového typu pracující se standardním vstupem a výstupem. Mohl bych zvolit řešení naprogramování aplikace v některém z kompilovaných jazyků, která bude spouštět jednotlivé úlohy, ale toto řešení by bylo obtížně rozšiřitelné už jen z toho důvodu, že při změně je zapotřebí program přeložit. Dále jsem

přemýšlel o řešení ve skriptovacích jazycích, ale od tohoto jsem také opustil ze dvou důvodů – psát program v Bashi mi přišlo více přímočaré a za druhé, nenalezl jsem vhodný framework v ostatních skriptovacích jazycích, který by byl vhodný k řešení daného problému.

Pokud shrnu úkoly frameworku, tak bude provádět základní operace nad složkami a soubory (vytvoření, mazání, změna), spouštění programů a zpracování jejich vstupů a výstupů. Nejvíce procesorového času zabere samotný běh jednotlivých externích programů – z tohoto důvodu není relevantní rychlost samotného Bashe. Pokročilé skriptovací jazyky (Perl, Python, PHP, atp.) navíc podporují mnoho funkcí, které bych nevyužil. Posledním důvodem bylo doporučení od vedoucího práce – zpracovat framework v Bashi z důvodů snadné přenositelnosti.

Při hledání již existujících řešení ohledně tokenového zpracování jsem nenalezl nic vhodného. Pro Bash existuje mnoho pomocných knihoven s užitečnými funkcemi, ale nic komplexního, co by se blížilo k mému problému. Využil jsem některé části, z těchto knihoven:

- Bash Shell Function Library (bsfl)¹ – inspirace pro logování.
- Bashinator² – inspirace logování, chybové zpracování, zpracování chyb.
- Shell Script Framework tool (shesfw)³ – zpracování parametrů.

Nicméně ani mimo jazyk bash jsem nenalezl nic uspokojivého, co se zabývá tokenovým zpracováním. Při implementaci jsem čerpal hlavně z [1, 2].

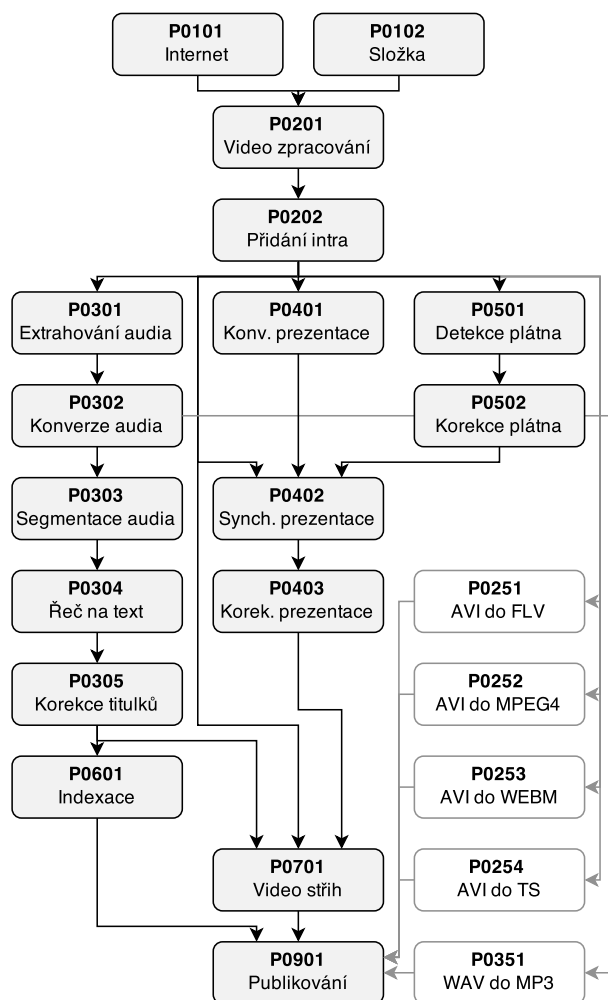
Funkce frameworku by měly poskytnout uživateli maximální komfort při obsluze. K dispozici jsou tyto možnosti:

- Spuštění všech procesů.
- Zastavení všech procesů po zpracování aktuální úlohy.
- Ukončení všech procesů ihned.
- Restart všech procesů.
- *Obdobně pro jednotlivé procesy.*
- Vytvoření adresářové struktury pro data tokenu.
- Vytvoření kostry nového procesu
- Zobrazení stromu závislostí procesů od tokenu
- Zobrazení běžících procesů

¹Viz <https://code.google.com/p/bsfl/>.

²Viz <http://www.bashinator.org/>.

³Viz <https://code.google.com/p/shesfw/>.



Obrázek 1. Diagram procesů – konfigurace možného propojení.

2. Tokenové zpracování úloh

Každá úloha ke zpracování je reprezentována *tokenem* – souborem, fyzicky uloženým na disku. Tento token bude “probublávat” přes jednotlivé atomické procesy. Pokud bude mít proces na vstupu token, zpracuje svou část a přepošle tento token jinému procesu.

Procesy musí mít předem dohodnuté rozhraní předávání vstupů a výstupů – v našem případě se jedná o textové soubory, které obsahují cestu k cíli.

Pomocí nastavení můžeme upravit chování zpracování. Například přeskočení nebo stále opakování úlohy (využito při čekání na vstup od uživatele).

3. Procesy

Procesy mají pevně daný formát *proces-ID_proces-název*.

Každý token “probublává” strukturou procesů až k poslednímu členu *publish*, který video zveřejní. Struktura je zobrazena na obrázku č. 1. Jedná se o základní nastavení, které lze jednoduše měnit změnou konfigurace procesu. Framework podporuje i šablony těchto

Obrázek 2. Ukázka uživatelského rozhraní – administrace spouštění procesů.

konfigurací. Při zpracování videí z přednášek budeme používat tři základní šablony:

1. Zpracování všeho – videa, audia, prezentace, převod řeči na text. (Toto je základní nastavení)
2. Zpracování videa, audia a převodu řeči – využijeme v případě, pokud máme k dispozici dva zdroje videa – plátno a přednášející. Tato šablona bude pro přednášejícího.
3. Zpracování videa a prezentace – šablona pro plátno.

Konfigurace jsou “kaskádovitě” uspořádány podle načítání a mohou se přepisovat.

1. Globální konfigurace – základní nastavení Frameworku.
2. Globální konfigurace tokenu – volitelná, načítá se pro každý proces.
3. Základní konfigurace procesu – povinná, u každého procesu.
4. Uživatelská konfigurace procesu – volitelná.

Framework podporuje vytváření šablon z jakéhokolí sestavení. Tyto šablony jsou vytvářeny druhou a čtvrtou konfigurací.

Každý proces je spuštěn v nekonečné smyčce a uchovává si tyto fronty tokenů:

- Fronta čekajících tokenů.
- Fronta tokenů, které skončily s chybou.
- Fronta dokončených tokenů.

Bash framework podporuje spuštění více instancí jednoho procesu. V takovém případě bylo nutné uzamykání fronty čekajících procesů, aby nedošlo ke zpracování jednoho tokenu více-krát.

4. SGE

V průběhu vývoje vznikl požadavek na podporu SGE⁴. Pokud je použití SGE povoleno, framework zaobaluje veškeré příkazy a odesílá je do fronty pro zpracování. Ve většině případů se jedná o neblokující operaci. S každou úlohou se odesílá příkaz pro případ chyby ve skriptu, který by token přenesl do chybové fronty.

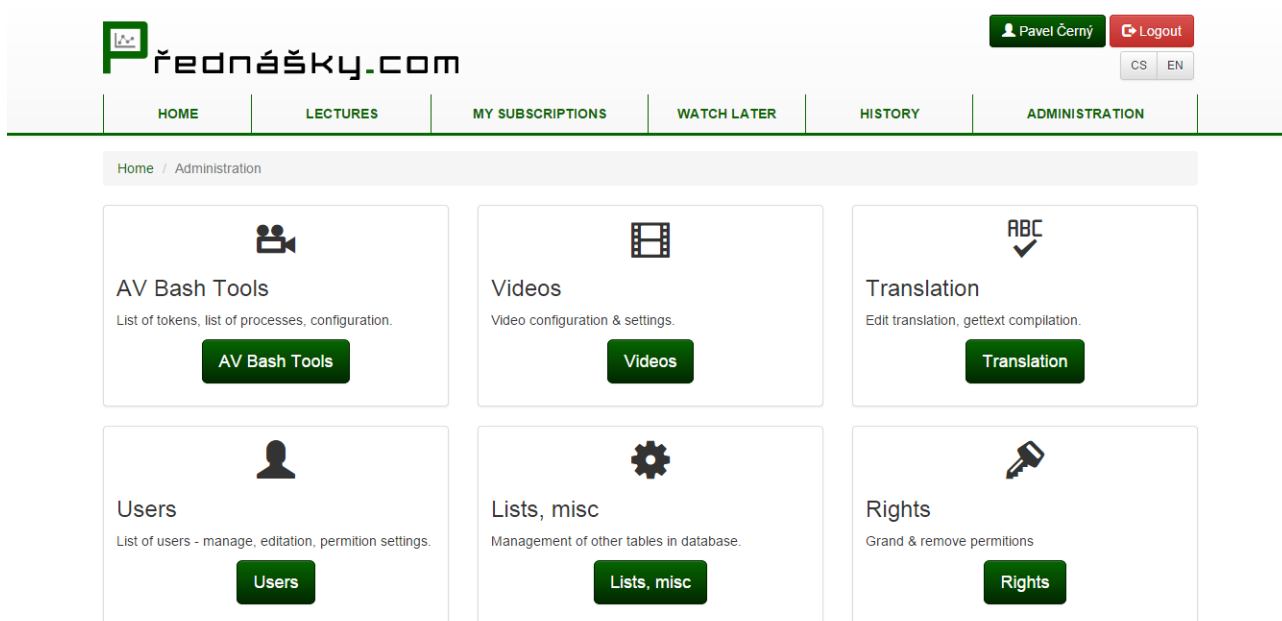
Pokud v rámci jednoho procesu odesíláme více příkazů na zpracování, které jsou na sobě závislé, jsou nastaveny parametry tak, aby druhý příkaz na první počkal. S posledním příkazem se odesílá skript, který v případě úspěchu spustí přeposlání tokenu dalším procesům. Pokud SGE vypneme, procesy spouští příkazy klasicky – synchronně.

5. Webové rozhraní

Tato část projektu zastřešuje Bash framework. Pro jednotlivé skupiny uživatelů poskytuje maximální komfort pro používání. Systém bude obsluhovat tyto skupiny uživatelů:

- **Běžný návštěvník** – poskytuje přehrávání veřejných videí a vyhledávání.

⁴Sun Grid Engine, nově Oracle Grid Engine (OGE) – systém pro správu dávkového zpracování úloh.



Obrázek 3. Ukázka uživatelského rozhraní – administrace.

- **Přihlášený uživatel** – studentovi navíc poskytuje uložení videí k pozdějšímu přehrání, “sledování” předmětu pro nové videa, přidávání komentářů a historii přehrávání.
- **Učitel** – systém nabízí zejména správu nově nahraných přednášek, kde je potřeba doplnit informace k videu a zkontrolovat nastavení před zpracováním. Proces, který čeká na vstup od učitele je P0201. Systém nabídne i řadu “předpřipravených” šablon nastavení. Jakmile učitel schválí video, token (popřípadě více tokenů, pokud má záznam dva zdroje nahrávání) pokračuje dále ve *stromu procesů*. Popřípadě může učitel nahrát své vlastní video do systému nebo pouze zadat URL s videem o kterou se postará proces P0101. Zpracování se zastaví dále u procesu P0701, kde bude mít učitel možnost *vystráhnout* část videa a vytvořit z ní samostatný úsek.
- **Administrátor** – má dozorovou kontrolu nad všemi procesy, může kontrolovat a nastavovat tokeny a k nim přiřazené videa. Ukázku vygenerovaného stromu procesů dle aktuální konfigurace pro daný token můžeme vidět pod titulkem práce – vygenerováno za pomoci js frameworku Raphaël⁵. Administrátor dále přiřazuje učitelům předměty a nastavuje výčtové tabulky v databázi.
- **Super administrátor** – může navíc spouštět

a vypínat procesy (viz obrázek č. 2) a spravovat překlady.

Webové rozhraní splňuje určité standardy responzivního designu – k tomu nám napomáhá framework Bootstrap⁶ – (viz obrázek č. 4). Dále jsem čerpal informace z [3].

6. Technologie

Systém je naprogramován v jazyce PHP. Tento skriptovací jazyk jsem si vybral z důvodů dřívějších zkušeností, široké škále podpory, doplňků a snadné dostupnosti. Aplikace je napsána s důrazem na oběktový přístup. Při implementaci jsem čerpal z [4].

Jako základ jsem si zvolil mikro-framework **Slim**⁷. Mezi jeho přednosti, které využívám, patří:

1. **Routování** – Mapování volání funkcí na libovolné HTTP požadavky a adresy, včetně volitelných částí cesty.
2. **HTTP** – Jednoduchá abstraktní vrstva požadavku a odpovědi nad protokolem HTTP.
3. **Middleware** – Snadné aplikování práv dříve než dojde ke spuštění aplikace.
4. **Cache** – Pokročilá práce s HTTP cache – (Etag, Expires, Last Modified).
5. **Dependency Injection** – Podpora funkcí, atributů a návrhového vzoru Singleton.

Dále bylo nutné zvolit šablonovací systém. Framework Slim podporuje základní zobrazovací funkce, ale

⁵Raphaël framework usnadňuje práci s vektorovou grafikou. Využívá k tomu SVG, každý objekt je vytvořen také jako DOM objekt. Viz <http://raphaeljs.com/>.

⁶Vyvinuto firmou Twitter – sada nástrojů pro práci s HTML, CSS a JavaScriptem. Viz <http://getbootstrap.com/>.

⁷Viz <http://www.slimframework.com/>.

ty dostatečné nebyly. Vybíral jsem z mnoha šablonovacích systémů a do užšího výběru se dostaly tyto:

- Dwoo
- Open Power Template
- RainTPL
- Smarty
- Twig

Nakonec jsem se rozhodl pro **Dwoo**⁸, který nejvíce odpovídá mým potřebám – podpora dědičnosti šablon, množství funkcí, cachování, výkonnost.

Pro práci s formuláři jsem si zvolil framework **PFBC**⁹, který podporuje integraci s frameworkem Bootstrap. Umožňuje rozsáhlá nastavení – vykreslování, atributů, validací a HTML5 prvků.

V administraci používám framework **XCRUD**¹⁰, který velmi usnadňuje práci s tabulkovými daty (základní operace vytvoření, čtení, úprava a smazání) – podporuje širokou škálu nastavení zobrazení, bezpečnosti, operací nad daty včetně Ajaxového zpracování (ukázkou můžeme vidět na obrázku č. 2 – tabulka ve spodní části).

Pokud pracuji přímo nad databází, používám další framework – **RedBeanPHP**¹¹. Jedná se o objektově relační manažer. Velmi usnadňuje práci ohledně jednoduchých operací – vytvoření, editace, mazání a hledání dat. Samozřejmě podporuje také zadávání libovolných dotazů.

Jako databázový systém jsem si zvolil **MySQL**, kvůli jednoduchosti a podpory funkcí, které využívám – zejména cizích klíčů a transakcí (formát úložiště InnoDB). Framework je oddělen od databáze a může pracovat zcela bez ní, což byl jeden z požadavků na systém. Z tohoto důvodu je nutné provádět jednosměrnou synchronizaci videí a tokenů, kterou má na starosti PHP – pro aktuální zobrazení dat.

7. Závěr

Tato práce se zabývá vytvořením systému, který na jedné straně usnadní zpracování samotných záznamů a na straně druhé atraktivní učení z těchto materiálů, ať už se jedná o samotné snadné přehrávání nebo diskuse mezi studenty a učiteli u videí.

Za největší přínos bych považoval nasazení a skutečné využití tohoto projektu – v plánu je nasazení na FIT, dále zpracování projektu jako celku.

⁸Viz <http://dwoo.org/>.

⁹PHP Form Builder Class. Viz <http://www.imavex.com/pfbc3.x-php5/>.

¹⁰Viz <http://xcrud.com/>.

¹¹Viz <http://redbeanphp.com/>.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Curabitur iaculis molestie velit. Donec ac lorem odio. Curabitur vestibulum imperdiet eros, eget gravida tortor blandit in. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Suspendisse sit amet aliquam orci, sed sagittis turpis. Ut metus sapien, feugiat quis justo vel, cursus accumsan nunc. Nullam lacinia scelerisque pulvinar. Donec augue sem, molestie eu erat id, condimentum congue felis. Maecenas rutrum lacus nec enim gravida sodales. Pellentesque lobortis fringilla commodo. Donec faucibus blandit tortor, in imperdiet augue. Curabitur semper arcu vel lorem porta, a placerat neque finibus. Aliquam quis urna fringilla, iaculis nulla eget, eleifend nibh. Quisque quis molestie neque.

Obrázek 4. Ukázka zobrazení na mobilních zařízeních.

Projekt se nachází ve fázi vývoje – nyní pracuji na webové části – hotovo je přihlašování, propojení s Bash frameworkem, odhrouhlení videí učitelem, překlady, administrace uživatelů a práv. Celkově se jedná o stránky využívané administrátory a učiteli. Serverová část je dokončena. Zbývá naprogramovat stránky využívané návštěvníky – přehrávání videí, kategorizace, historie, odběry a obsluha emailů.

V budoucnu mám v plánu dále pokračovat na vylepšení tohoto projektu. Například se může jednat o zpracování záznamů z konferencí, lepší sledování v průběhu zpracování, zdokonalování webového rozhraní nebo rozšíření projektu mezi další fakulty.

Závěrem bych rád poděkoval vedoucímu své práce – panu doktoru Igoru Szökemu, za cílené vedení, ochotu na konzultacích a při tvorbě práce.

Literatura

- [1] Cameron Newham, J. Vossen, Carl Albing, and Jp Vossen. *Bash Cookbook: Solutions and Examples for Bash Users (Cookbooks (O'Reilly))*. O'Reilly Media, Inc., 2007. ISBN: 0596526784.
- [2] Dave Taylor. *Wicked Cool Shell Scripts: 101 Scripts for Linux, Mac OS X, and Unix Systems*. No Starch Press, San Francisco, CA, USA, 2004. ISBN: 1593270127, 9781593270124.
- [3] Ethan Marcotte. *Responsive web design*. A Book Apart, New York, New York, USA, 2011. ISBN: 9780984442577.
- [4] Hasin Hayder. *Object-Oriented Programming with PHP5: Learn to Leverage PHP5's OOP Features to Write Manageable Applications with Ease*. Packt Publishing, 2007. ISBN: 1847192564, 9781847192561.