

Jméno a příjmení: Filip Kalous

Login: xkalou03

## 1) Zadání

Mým úkolem bylo vytvořit skript v jazyce PHP, který analyzuje hlavičkové soubory jazyka C (přípona .h), a vytvoří XML strukturu funkcí nalezených v daných souborech. K načtení souboru a následnému vypsání skript využívá dvou knihoven jazyka PHP, a to *XMLWriter* a *RecursiveDirectoryIterator*.

## 2) Implementace

Celý program je napsán v kuse bez rozdělení na podprogramy, až na 2 výjimky. Ty tvoří funkce pro vypsání nápovědy *help()*, a funkce figurující v kontrole parametrů, která bude popsána níže.

## 3) Zpracování parametrů

Kontrola správného zadání parametrů probíhá hned na začátku skriptu. K této kontrole skript využívá funkce *getopt()*, která získá argumenty z příkazové řádky, přičemž nezáleží na pořadí, jak jsou argumenty zadány. Při této kontrole se v případě špatných argumentů rovnou ukončuje program s příslušnou chybou, ať už jde o špatně zadané soubory pro vstup a výstup nebo nedodržení požadavků parametrů.

## 4) Procházení souborů

Jak už bylo zmíněno výše, pro práci se soubory program využívá dvou knihoven. Pokud je parametrem programu zadán přímo soubor, knihovna pro rekurzivní procházení souboru se nevyužívá. V případě, že je vstupem je celá složka, program projde aktuální složky i podsložky a uloží cesty k souborům.

Původně jsem chtěl pro zapsání výstupu programu do XML souboru využívat knihovnu *SimpleXML*, ale *XMLWriter* se pro své funkce v tomto programu hodí mnohem lépe. Celý výstupní formát se nejdříve ukládá do paměti, a po dodatečných úpravách se zapíše do XML souboru nebo se pošle na standardní výstup.

## 5) Hledání funkcí v souborech

Hlavní částí celého programu jsou regulární výrazy, které jsou použity, jak pro upravení vstupního souboru, tak pro hledání definic a deklarací funkcí. V případě nalezení definice funkce se regulárními výrazy hledají i parametry této funkce. I když napsat regulární výraz přesně podle požadavků zadání, které se pak lehce změnili při zveřejnění referenčních výstupů k testům, nebyla příjemná práce, řešit tento problém např. přes konečný automat nebo LL gramatiku jsem raději nezvolil.

## 6) Závěr

Skript byl náležitě otestován testovací sadou vydanou k zadání. Celý skript byl psán a testován na školním serveru Merlin již od začátku, z důvodu vyhnutí se nekompatibility výsledného skriptu při psaní na jiném operačním systému s podporou jiné verze PHP. Celý skript prošel testy bezchybně.