

Analýza úlohy

Cílem projektu bylo vytvořit skript, který zvýrazní části vstupního textového souboru, přičemž zvýraznění je realizováno přidáním formátovacích značek podobných html přímo do textu. Zvýrazňované části odpovídají regulárním výrazům ve formátovacím souboru.

Úlohu lze rozdělit na následující podúlohy: zpracování argumentů, zpracování formátovacího souboru, transformace regulárního výrazu do podoby podporované v PHP 5, vytvoření formátovacích značek pro styl zvýraznění, vložení formátovacích značek do vstupního textu a zápis zvýrazněného textu do výstupního souboru. Nebylo implementováno žádné rozšíření.

Zpracování argumentů

Argumenty jsou zpracovávány po jednom, neprobíhá kontrola zda existují soubory zadané pomocí argumentů `--input`, `--format` a `--output`, ta je odložena až do chvíle kdy se má z daného souboru číst nebo do něj zapisovat. Každému argumentu je přidělena proměnná, která slouží k vyloučení vícenásobného zadání stejného argumentu.

Zpracování formátovacího souboru

Formátovací soubor je čten po řádcích. Pokud řádek není prázdný, dojde k jeho rozdělení na regulární výraz a styl zvýraznění. Regulární výraz ani styl zvýraznění nesmí být prázdným řetězcem jinak je formátovací soubor považován za chybný. Kontrola správnosti regulárního výrazu a stylu se provádí až při jejich úpravě.

Čtení řádků z formátovacího souboru řídí chod zbytku programu, ostatní úkony se provádí pro každý řádek zvlášť a program je úspěšně ukončen když je přečten celý formátovací soubor.

Transformace regulárního výrazu

Formát regulárních výrazů specifikovaný v zadání neodpovídá regulárním výrazům používaným v PHP 5. Převod je uskutečněn pomocí konečného automatu, který čte vstupní regulární výraz znak po znaku a vytváří odpovídající regulární výraz pro PHP 5. Konečný automat je implementován ve funkci `convert_regex()` a očekává regulární výraz obalený v závorkách (vstupní výraz je obalen závorkami předem). Toto umožňuje rekurzivní volání `convert_regex()` nad uzávorkovaným podvýrazem, ale zabraňuje přečtení jediného znaku po negaci (není uzávorkován). Proto je implementována další funkce `read_one()`. Jedná se modifikovaný konečný automat z `convert_regex()`, který přečte jeden neuzávorkovaný znak.

Konečný automat z `convert_regex()` má následující stavy: `init` je počáteční stav, na vstupu je očekávána otevírací závorka, `read` je výchozí stav pro čtení znaků, `escape` je stav do kterého se automat dostane po přečtení symbolu `%`, `negate` je stav do kterého se automat dostane po přečtení symbolu `!`, z tohoto stavu je volána funkce `read_one()` pokud je potřeba znegovat neuzávorkovaný výraz.

Po návratu z funkce `convert_regex()` je zkонтrolováno, jestli byl přečten celý vstupní regulární výraz a jestli se neobjevila zakázaná kombinace znaků (návratová hodnota `null`)

Formátovací značky

K jednotlivým stylům zvýraznění jsou postupně přidávány formátovací značky do proměnných `$open` a `$close`, přičemž otevírací značky se vkládají ve stejném pořadí v jakém jsou styly uvedeny a uzavírací v opačném. U značek s parametry je kontrolován formát parametru.

Vkládání značek

Do proměnné `$output` je zkopirován vstupní text.

Pomocí funkce `preg_match()` je nalezena pozice ve vstupním textu, která odpovídá regulárnímu výrazu. Do vstupního textu se formátovací značky nevkládají, ten slouží pouze pro vyhledávání. Značky se vloží před a za najitou část textu do `$output`, přičemž k těmtu hodnotám jsou přičteny délky již vložených značek, které jsou uchovány v poli `$offset`. Po vložení značek je pole aktualizováno.

Prohledávání pokračuje dokud jsou nacházeny části textu odpovídající regulárnímu výrazu nebo dokud není zpracován celý vstupní text.

Výstup

Nakonec je obsah proměnné \$output vypsán do souboru zadaného parametrem --output, až v této chvíli se ověřuje, zda lze daný soubor otevřít a zapsat do něj.