

Popis zadání

Účelem úlohy C Header Analysis je nalezení hlavičkový souborů, nalezení v nich funkcí jazyka C a zpracování informací o funkcích do formátu XML. Mezi informace patří návratový typ funkce, jméno funkce, počet parametrů funkce, typy parametrů funkce a poznámka, zda má funkce proměnný počet parametrů, či nikoliv. Výsledkem zpracování úlohy je skript ve formátu PHP5, který je možno spouštět s různým počtem parametrů.

Popis parametrů

Skript na začátku očekává, že uživatel zadá parametr vstupu, který udává relativní nebo absolutní cestu k souboru nebo adresáři. Pokud je zadána cesta k souboru, skript zpracovává pouze tento soubor. Pokud je zadána cesta k adresáři, skript vyhledá veškeré hlavičkové soubory v tomto adresáři a ve všech jeho podadresářích. Pokud není vstupní parametr zadán, prohledá se adresář, ve kterém je skript spuštěn. Zadáním výstupního parametru uživatel udává cestu k souboru, do kterého se zapíše informace o funkcích. Pokud není uživatelem zadán, informace se vypíše na standardní výstup.

Parametrem `no-inline` skript přeskočí funkce, které obsahují identifikátor `inline` a tudíž je nevypíše na výstup.

Informace o funkcích ve formátu XML je standardně vypsána bez odsazení a jednotlivé atributy nejsou odřádkovány. Pokud by uživatel chtěl XML dokument zpřehlednit, `pretty-xml` může určit velikost odsazení atributů a také odřádkování mezi atributy. Je-li zadán, ale není uvedena velikost odsazení, budou atributy odsazeny 4 mezerami.

Dále je možné zadat parametr `max-par` s libovolným číslem. Toto číslo říká, že budou zpracovány pouze funkce, které mají méně nebo stejně parametrů jako je zadané číslo. Číslo se pouze pevně dané parametry.

Parametrem `remove-whitespace` skript odstraní všechny přebytečné bílé znaky u návratovém typu funkce a u typů parametrů funkce a nahradí je mezerou. Pokud není zadán, ve výsledném dokumentu jsou návratové typy a typy parametrů tak, jak jsou zapsány v souboru.

Pokud uživatel zadá `no-duplicates` a budou-li v jednom souboru stejně pojmenované funkce, ve výsledném dokumentu bude vypsána pouze první z nich. Pokud není zadán, vypíše se všechny funkce bez ohledu na jméno.

Posledním parametrem, který je možné zadat je `help`, který vypíše na standardní výstup nápovědu ke skriptu, která informuje uživatele o jednotlivých parametrech skriptu. Není dovoleno zadat parametr pro tisk nápovědy s některým z dalších parametrů. Tento případ není dovolen a skript se ukončí s chybou. Stejně tak pokud uživatel zadá výše popsané parametry ve špatném tvaru, bude to bráno jako chyba.

Popis implementace

Skript je rozdělen na zpracování parametrů z příkazové řádky a na dvě funkce, které je dále zpracovávají. Pro ověření, zda byl konkrétní parametr zadán, slouží pole typů `bool`. Ověření správnosti zadaných parametrů a nalezení konkrétního parametru kontrolují regulární výrazy. Pomocí funkce `getopt`, jsou získány hodnoty z parametrů, které předávají hodnoty. Všechny získané informace jsou předány funkci `rek_search`, která zajišťuje prohlížení adresářů nebo souborů. Pokud je na vstupu zadána přímo cesta k souboru, nekontroluje se jeho přípona. Pokud je zadán adresář, funkce `rek_search` se rekurzivně opakuje.

Pro každý odpovídající soubor se spouští funkce `zprac_file`, která zpracovává soubor a tiskne výsledný XML dokument. Nalezení funkce, návratového typu funkce, jména funkce, typů parametrů a proměnného počtu parametrů je zpracováno pomocí regulárních výrazů. Nalezení návratového typu funkce spočívá v nalezení posloupnosti všech znaků před jménem funkce. Při hledání parametrů funkce se předpokládá, že musí být mezi kulatými závorkami. Potom dochází k jejich oddělení na základě čárky funkcí `explode`, která je uloží do pole. Výsledný typ se získá podobně jako návratový typ samotné funkce. Bílé znaky před a po

návratových typech jsou odstraněny pomocí funkce `trim` a regulárními výrazy. Pokud je parametr funkce typu `void` nebo nejsou zadány žádné parametry, musí být funkce brána jako funkce bez parametrů a tudíž nesmí být vypsán atribut `param` u XML dokumentu. Speciálním případem je pak ukazatel na `void`, který musí být brán jako regulérní typ parametru. Má-li funkce proměnný počet parametrů, bude u atributu `varargs` napsáno „yes“, v opačném případě „no“. Ověření, zda ve funkci existují, probíhá hledáním tří teček, které je identifikují. Dále musí být snížen počet parametrů o 1, poněvadž se nejedná o parametr a nesmí být ani vypsán atribut `param`. V případě, že typ parametru je pole, bude nalezeno a dodatečně připojeno k zbývajícím atributům.

Zadání úlohy říká, že mají být ignorovány deklarace funkcí v komentářích, řetězcích a v makrech. Skript tomu zabráňuje tak, že dopředu ze souborů vymaže komentáře, makra a řetězce. Zde mohl nastat problém při správném vyhledání maker a komentářů, které jsou rozděleny „\“ na více řádků. Řešením je vymazání všech takovýchto znaků, které přímo sousedí se znaky konce řádků. Tato implementace by neměla mít vliv na nalezení funkcí, poněvadž by se tento znak neměl vyskytovat v deklaraci funkce.

Pro případ spuštění s `no-duplicates` je k dispozici pole, které je postupně plněno jmény funkcí v pořadí výskytu v souboru. Pokud se již funkce s daným jménem v poli vyskytuje, nebude jméno do pole přidáno a funkce nebude vypsána v XML dokumentu.

Skript musí být spouštěn s interpretem `php -d open_basedir=""`.