

Popis implementace od spuštění skriptu až po ukončení:

Po uvedených konstantách a globálních proměnných se spouští první funkce.

zpracuj_param() Skript po spuštění zpracuje parametry příkazového řádku. Ošetří případné chyby jako vícekrát zadaný parametr nebo parametr, `-help` zadaný současně s nějakým jiným parametrem. Je-li zadán parametr `-help`, zavolá se funkce `help()` pro výpis nápovědy a skript končí. Pokud parametr/y nevyhovuje, zavolá funkce `zpracuj_chybu()` s návratovým kódem 1 ke které se dostanu později.

Po úspěšném zpracování parametrů si otestujeme parametr `-input`. Zapišeme hlavičku pomocí funkce:

hlavicka_xml() ve které je: `<?xml version="1.0" encoding="utf-8"?> a <functions dir="'+el_dir+'">` Za `el_dir` se dosadí patřičný element dle zadání. V případě zadání hlavičkového souboru soubor ihned analyzujeme funkcí `analyzuj()`, v případě složky nebo pokud není `-input` zadán voláme funkci `xml()`.

xml() Prohledává rekurzivně adresářovou strukturu. Pokud najde hlavičkový soubor, volá funkci `analyzuj()`, pokud adresář zanoří se do adresáře a volá funkci `xml()`. Prohledávání končí, pokud vyčerpáme všechny soubory a složky z adresáře který zadal uživatel.

analyzuj() Analyzuje hlavičkový soubor a výsledek zapisuje do XML dokumentu. Musí se předpřipavit soubor:

predpriprav_soubor(), která odstraní ze souboru komentáře, makra, bílé znaky na začátku a konci souboru. Zkontrolujeme zanoření:

zanoreni() Odstraní zanoření Příklad:

```
{{{.struct struktura;.}}{...}}}; int main(void typ){retur n
10;} Potřebujeme odstranit:{{{.struct truktura;.}}{...}}}
```

Dále pokračuje funkce `analyzuj()`, která rozdělí funkce v souboru na seznam funkcí pomocí `split` podle `'.'`. Iterujeme přes seznam funkcí. Zjišťujeme elementy do XML souboru jako název funkce, návratová hodnota funkce, proměnný počet parametrů. Zbývá určit parametry. Ty se rozdělí pomocí `split` podle `'.'` do seznamu parametrů. Iterujeme přes parametry. Zjišťujeme typ parametru.

Získané elementy postupně zapisujeme do XML dokumentu. Starají se o to funkce `zapis_el_function()`, `zapis_el_param()`, `zapis_el_k_function()` viz pomocné funkce. Ve funkci `analyzuj()` provádíme operace takové, aby vyhovovaly použitým parametrům. Např. při zadání parametru:

`-no-inline` se přeskočí všechny funkce deklarované specifikátorem `inline`

`-max-par=n`. Pokud je tento parametr zadán voláme funkci:

predpocitej_param() Spočítá a vrátí počet parametrů.

Podle parametru `n` ihned víme, zda tato funkce vyhovuje uživatelem definovanému parametru či ne. Pokud nevyhovuje, nezapisujeme tuto funkci do XML dokumentu.

`--no-duplicates` Jméno každé funkce se přidává do seznamu. Při zadání tohoto parametru se zkontroluje seznam a pokud se tam stejnojmenná funkce už nachází, nezapisujeme ji do XML dokumentu.

`--remove-whitespace` U návratového typu a typu parametru nahradí všechny bílé znaky za mezeru a zredukuje počet mezer na nejméně možný.

Při ukončení skriptu končí skript s návratovým kódem 0, v případě chyby s návratovým kódem odpovídajícím chyby.

Pomocné funkce:

otevri_a_precti_soubor() Otevře soubor, uloží do proměnné, zavře soubor. Ošetřuje vyjímky.

zapis_a_uloz_soubor() Otevře soubor, uloží proměnnou do souboru, zavře soubor. Ošetřuje vyjímky.

zapis_el_function() Parametrem funkce jsou elementy `el_file`, `f_nazev`, `f_var_args`, `f_navrat`, které budou použité při zápisu do XML dokumentu.

```
<function file="'+el_file+' name="'+f_nazev+' varargs="'+f_var_args+'
rettype="'+f_navrat+'">
```

zapis_el_param() Parametrem funkce je element `f_typ`, který je použit při zápisu do XML dokumentu

```
<param number="'+str(c_parametru)+' type="'+f_typ+' />
```

zapis_el_k_function() Zapiše element `</function>`

zpracuj_chybu() Parametrem je číslo chyby. Zapiše chybu na `STDERR` a ukončí skript s odpovídajícím návratovým kódem.

napoveda() Výpis nápovědy.

Pozn. u všech zápisů do XML dokumentů se testuje parametr `-pretty-xml=k`, který provede se patřičné odsazení.