Task documentation CST: C Stats in english for IPP 2011/2012
Name and surname: Martin Riša
Login: xrisam00

Use of script

Perl-script cst.pl(further only script) counts statistics of valid C programming language files. Script counts occurrences of keywords, identifiers, operators, specified pattern and counts total length of comments. What is counted is specified by command line arguments(further only arguments), *-k* for keywords, *-o* for operators, *-i* for identifiers, *-w=pattern* for *pattern* occurrences counting and *-c* for total length of comments counting. Input files are specified by argument *–input=fileordir,* where *fileordir* argument represents full path to any file or folder on local filesystem. If folder path is given in input argument, script recursively passes through subfolders and processes every file in them which has *".c"* or *".h"* extension unless *--nosubdir* argument is set. In case *--nosubdir* argument is set script searches for such files only in given folder. If path to a file is given script processes only that file. Output is specified by argument *--output=file*, where file represents the name of the output file. Argument *-p* causes that output does contain only names of files with specified statistic result. Every line of output file except last one consists of pairs of filename and counted statistic. Last line of output file contains sum of statistics for each processed file. Script prints help if argument *--help* is set.

Implementation specification

- **Shared implementation for different statistics**

Script parses arguments using function *Long:Getopts()* whereby *Long:Bundling* option is set, this causes that long arguments could not be set by short prefix *'-'*. Script is able to count one statistic in one run so arguments *-k,-i,-o,-c* and *-w* are exclusive. At the beginning script creates list of files and fills it with full filenames of files to be processed. Full filenames are retrieved by use of function *CWD:cwd()* concatenated with relative filepath. This list is sorted alphabetically by full filenames or by filenames only if *-p* argument is set. Filenames are extracted from full filenames by use of function *File:Basename()*. Files are processed one by one. At first preprocessor constructions are omitted from processing unless argument *-w=pattern* is set. Preprocessor constructions of language C are enumerated in C-language standard.

- **Keywords counting**

At first script deletes string literals. In case string literal is written on more than one line and it doesn't have proper ending syntax error 123 is returned. Afterward block comments are deleted from processing as well as line comments. Then script counts occurrences of keywords delimited by zero-space character *\b*. Keywords are part of C-language standard and can be found in script in array called *KEYWORDS*.

- **Length of comments counting**

In the beginning every string literal is replaced by string with same length as original string literal but consisted only of spaces. Then file array is concatenated to single string and length of block comments is counted whereby all found block comments are replaced with single space. Afterward resulting string is split to a file array by lines backwards. Then length of line comments is counted.

- **Operators counting**

String literals are omitted from processing , validation of string literals is the same as in keywords counting. Afterward every block comments are replaced by single space as well as line comments. Then script searches for user defined types. Which are stored in list of defined types. Parsing user defined types can recognize defining of structure types and renaming of previously defined types. After all types are recognized from processing are omitted declarations of pointers which are recognized from list of defined types followed by star character. Afterward all occurrences of string "..." is omitted from processing. Then the occurrences of supported operators are counted.

- **Identifiers counting**

String literals are omitted from processing , validation of string literals is the same as in keywords counting. Afterward every block comments are replaced by single space as well as line comments. Keywords are omitted also from processing by replacing every occurrence of keyword by single space. Afterward numeric constants are omitted from processing as well as character literals. Identifiers are counted by regular expression defined in C-language standard from remaining parts of processed file.

- **Occurrences of pattern counting**

Search for pattern occurrence is done within whole file including preprocessor constructions. Pattern can be any sequence of characters.