

Tým xhricma00 varianta vv-BVS

Marek Hric
xhricma00

Mikuláš Lešiga
xlesigm00

Roman Andraščík
xandrar00

Adam Veselý
xvesela00

November 27, 2024

Rozdelenie bodov

xhricma00: 25%
xlesigm00: 25%
xandrar00 :25%
xvesela00: 25%

Rozšírenia

ORELSE
UNREACHABLE
BOOLTHEN
FOR
WHILE
FUNEXP

Rozdelenie prace :

Marek Hric :

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum

Mikuláš Lešiga :

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum

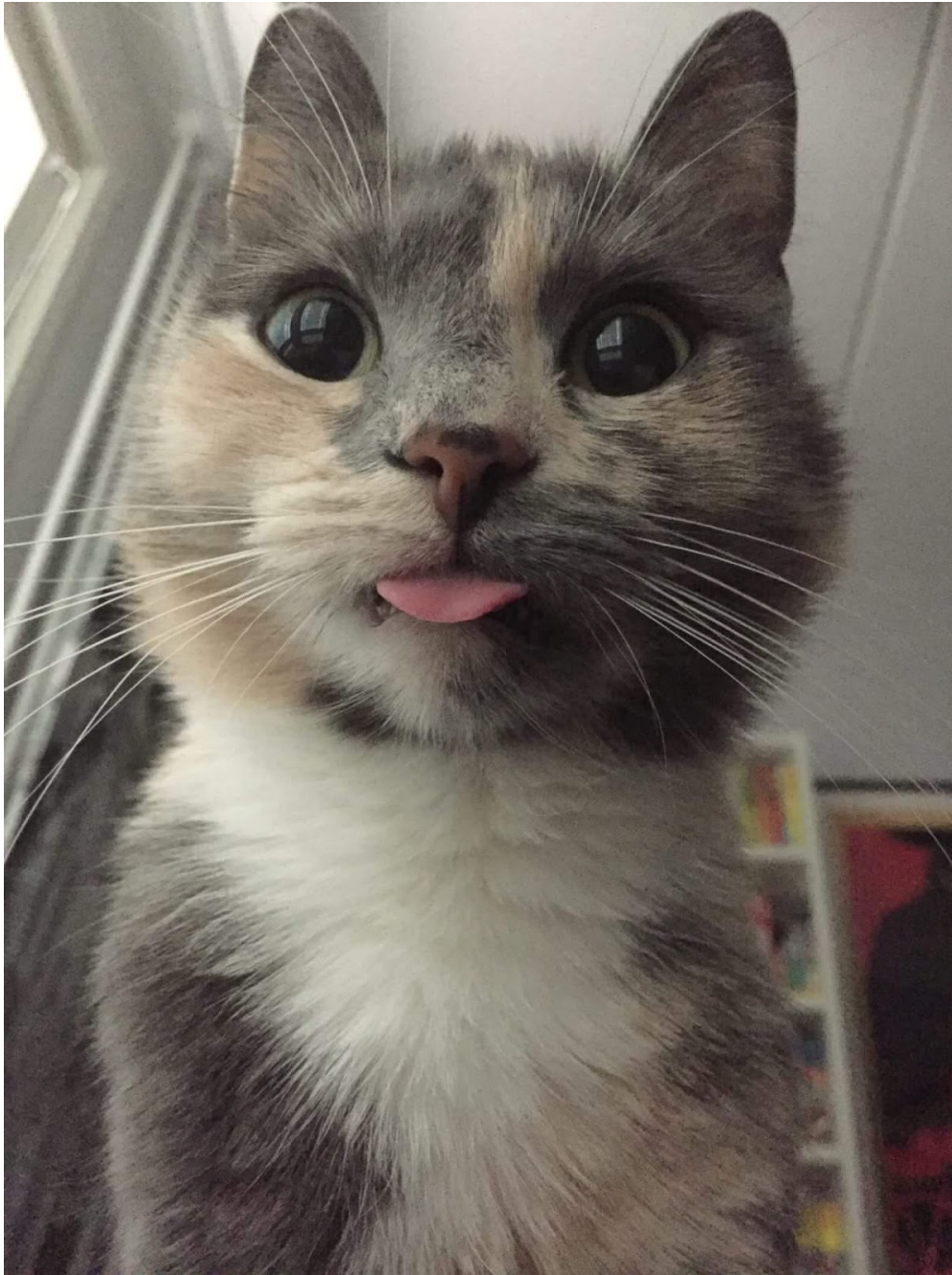
Roman Andraščík :

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum

Adam Veselý :

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum

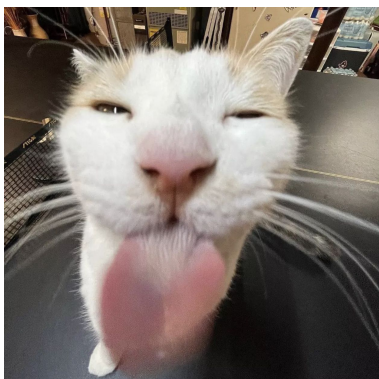
Diagram konečného automatu :



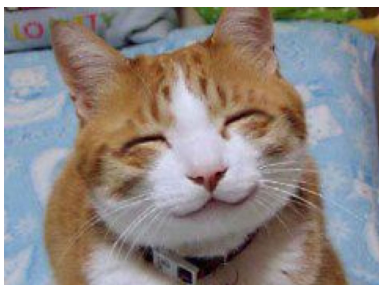
LL-gramatika :

1. aaaa
2. bbbb
3. cccc
4. dddddddd

LL-tabulka :



Precendecna-tabulka :



Lexikalna analyza

Riesenie lexikalnej analyzy sme zacali vytvorenim diagramu deterministickoho konecneho automatu. Nasledne sme na jeho zaklade zacali vypracovavat implementaciu. Implementacia sa nachadza v subore *scanner.c*, ktory pracuje s tokenmi deklarovanymi v subore *token.h*. Hlavnou funkciou *scanner.c* je funkcia *get_token*. Pre ulahcenie prace a prehľadnosti kodu sme si deklarovali niekoľko makier, ktoré sú extenzívne používané v hlavnej funkcii. Funkcia *get_token* berie postupne znaky z standardneho vstupu a vytvára token. Tokenu je priradený jeho typ a hodnota ktorá mu odpovedá. Funkcia začína určovaním jednoznakových tokenov, ktoré vie určiť hneď na začiatku. Pokračuje identifikáciou komentárov, ktoré následne ignoruje. Po identifikácii komentárov zisťuje či sa jedná o ID alebo Keyword, pri kľúčových slovách sa následne určuje aj ich typ. Ak sa nejedná ani o jedno pokračuje kontrolou dátových typov pri ktorých ukladá aj ich hodnoty.

Syntakticka analyza

Riesenie syntaktickej analyzy sme započali vytvorením LL gramatiky, LL tabulky a precedencnej tabulky. Následne na ich základe sme vypracovali subor *parser.c* a *exp_parser.c*. Tieto subory pracujú s uzlami deklarovanými v subore *ast.h*. Spustenie syntaktickej analyzy započne zavolaním funkcie *Parse()*. Táto funkcia postupne prechádza cez tokeny a priradzuje ich do uzlov pomocou ktorých postupne tvorí abstraktný syntaktický strom na základe LL gramatiky. Subor *parser.c* ďalej riadi aj precedencnú analýzu volaním funkcií zo suboru *exp_parser.c*. Tento subor vytvorí strom výrazov, ktorý je následne pripojený do syntaktického stromu.

Semanticka analyza

Semantická analýza je implementovaná v suboroch *sem_anal.c*, *symtable.c*, *sem_anal.h* a *symtable.h*. Spustenie semantickej analyzy započne zavolaním funkcie *analyse()*. Táto funkcia prechádza vytvoreným AST a postupne určuje či vyhovuje pravidlám jazyka IFJ24. Vyhovujúce funkcie sú následne vložené do tabulky symbolov, ktorá je deklarovaná v subore *symtable.h*.

Generovanie kodu

Generator je implementovaný v suboroch *codegen_priv.h*, *codegen.h* a *codegen.c*. Spustenie generácie kodu započne zavolaním funkcie *codegen()*.

Datove struktury

Circular Buffer

Implementovane v suboroch *circ_buff.c* *circ_buff.h*.

Implementacia Circular Buffer je vyuzita hlavne v casti Scanner kde sluzi na bezpreblemove ziskavanie dat a ich naslednu validaciu. Na pracu so scannerom ho neskor vyuzivaju aj casti Parser a Expression Parser. Struktura obsahuje klasicke funkcie *circ_buff_init*, *circ_buff_free*, *circ_buff_enqueue*, *circ_buff_dequeue*, *circ_buff_is_empty*.

Dynamic String

Implementovane v suboroch *dyn_str.c*, *dyn_str.h*.

Implementacia dynamickeho retazca je vyuzita hlavne v Scanner casti programu kde sprostredkuvava validaciu a uschovavanie dat, neskor je pouzita aj v casti Codegen kde sluzi na ulahcenie validacie dat. Struktura dynamickeho retazca obsahuje klasicke funkcie *dyn_str_init*, *dyn_str_grow*, *dyn_str_append*, *dyn_str_append_str* a *dyn_str_free*.

Stack

Implementovane v suboroch *stack.c*, *stack.h*.

Implementaciu nasho zasobniku vyuzivame v Expression Parser casti programu. Struktura zasobniku je implementovana s klasickymi funkciami *stackInit*, *stackPush*, *stackPop*, *stackIsEmpty*, *stackClear* a *stackGetTop*. Zasobnik sme zvolili pre jeho optimalny pristup k datam a zachovanie jednoduchosti kodu.