**Vysoká škola polytechnická Jihlava**

Katedra technických studií   
Programovací jazyky a překladače

**Překladač z jazyka daného gramatikou do mezikódu ve formě čtveřic**semestrální práce

Autor práce: Tomáš Cink  
Jihlava 2019

Obsah

[Zadání 3](#_Toc9791986)

[1 Analýza problému 5](#_Toc9791987)

[2 Teoretický a technický návrh řešení 6](#_Toc9791988)

[2.1 Lex 6](#_Toc9791989)

[2.2 Yacc 6](#_Toc9791990)

[2.3 Makefile 9](#_Toc9791991)

[3 Testování 11](#_Toc9791992)

[4 Závěr 18](#_Toc9791993)

Seznam obrázků

[Obrázek 1 - Gramatika (zdroj: zadání semestrální práce) 3](#_Toc9791994)

[Obrázek 2 - Lexikální analýza (vlastní) 6](#_Toc9791995)

[Obrázek 3 - Ukázka ze souboru sp.y (vlastní) 7](#_Toc9791996)

[Obrázek 4 - Práce s parametrem 8](#_Toc9791997)

[Obrázek 5 - Výpis čtveřic s pozicemi a symboly 9](#_Toc9791998)

[Obrázek 6 - Makefile (vlastní) 10](#_Toc9791999)

[Obrázek 7 - Výpis nápovědy 11](#_Toc9792000)

[Obrázek 8 - Testovací soubor 1 11](#_Toc9792001)

[Obrázek 9 - Test č.1 mezikód (vlastní) 12](#_Toc9792002)

[Obrázek 10 - Testovací soubor 2 (vlastní) 13](#_Toc9792003)

[Obrázek 11 - Test č.2 mezikód (vlastní) 13](#_Toc9792004)

[Obrázek 12 - Test 2 základní trasování (vlastní) 14](#_Toc9792005)

[Obrázek 13 - Testovací soubor 3 (vlastní) 15](#_Toc9792006)

[Obrázek 14 - Úplné trasování testu 3 (vlastní) 16](#_Toc9792007)

[Obrázek 15 - Test č. 3 mezikód (vlastní) 17](#_Toc9792008)

Seznam tabulek

[Tabulka 1 - Syntaxe a sémantika (zdroj: zadání semestrální práce) 3](#_Toc9792009)

[Tabulka 2 - Parametry (zdroj: zadání semestrální práce) 4](#_Toc9792010)

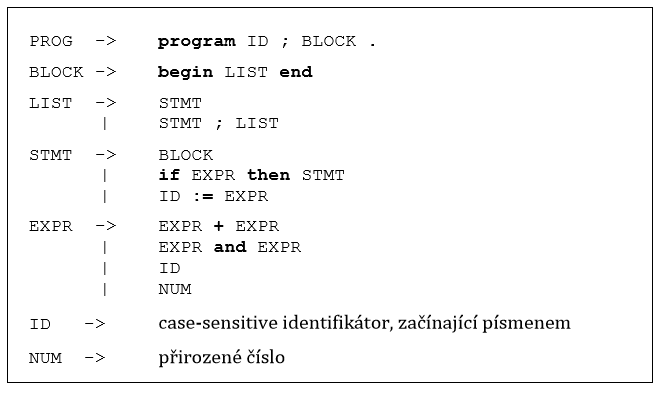
Seznam použitých zkratek

LEX Lexikální analýza

YACC Yet another compiler compiler

# Zadání

Navrhněte, vytvořte, otestujte a zdokumentujte překladač programovacího jazyka, generovaného následující gramatikou:



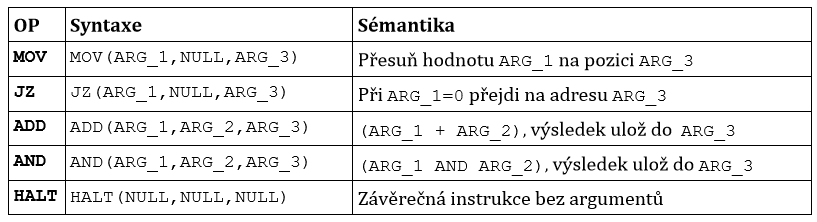
Obrázek - Gramatika (zdroj: zadání semestrální práce)

a generujícího intermediální kód (mezikód) ve formě čtveřic (quadruples) s tímto obecným formátem:



a následující syntaxí a sémantikou:

Tabulka - Syntaxe a sémantika (zdroj: zadání semestrální práce)



**Poznámky ke specifikaci jazyka a způsobu implementace:**

1. Všechny „neviditelné“ mezery (white spaces) jsou ignorovány
2. Jazyk povoluje samostatně stojící celořádkové komentáře, předznamenané středníkem:

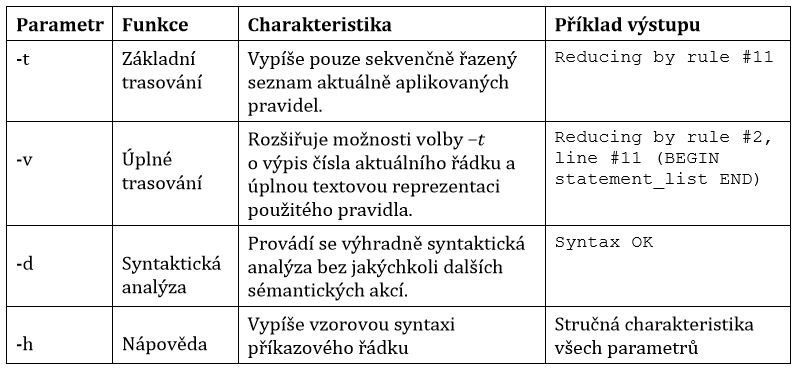
Kód 1

; Toto je příklad komentáře

Kód 2

1. Výsledný překladač bude realizován pomocí standardních generátorů (např. typu lex a yacc) ve formě dvou souborů, lexikálního a syntaktického analyzátoru. Překlad projektu zajistěte prostřednictvím utility make. Ke zdrojovým kódům dodejte nejméně 3 testovací příklady.
2. Překladač pracuje se standardním vstupem a výstupem (stdin, stdout)
3. Zajistěte, aby překladač po nalezení jakékoli syntaktické chyby nepřerušil svoji práci, ale zpracoval i zbytek vstupního kódu. Na konci vypíše, pokud možno co nevýstižnější informaci o chybách, které se v průběhu překladu vyskytly.
4. Spustitelný soubor, realizující Váš překladač, bude v závislosti na parametrech příkazového řádku pracovat v následujících režimech:

Tabulka - Parametry (zdroj: zadání semestrální práce)



# Analýza problému

Hlavním krokem k analýze je rozvrhnout tokeny potřebné v gramatice. V zadání jsou zvýrazněné a jsou to slova *program, begin, end, if, then, and* a symboly *„;“* pro zaznačení komentáře, *„.“* pro označení konce programu, *„+“* ke sčítání dvou expression a přiřazení pomocí *„:=“*. Dále lze zadat libovolnou proměnnou značenou v ID anebo numerické hodnoty značené NUM. Tyto tokeny jsou vytvořeny v lexikální analýze.

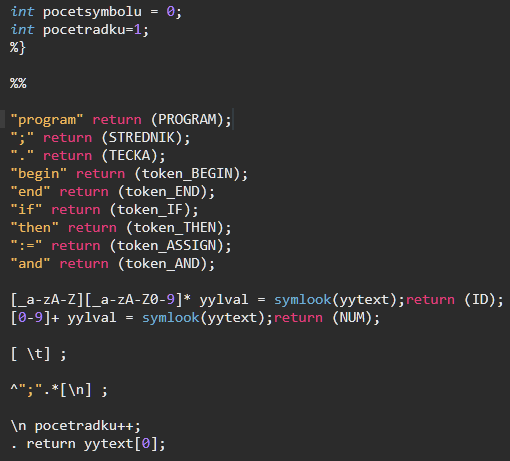
Pro syntaktickou analýzu lze využít zadanou gramatiku, kde jsou zvolené neterminály *PROG*, *BLOCK*, *LIST*, *STMT*, *EXPR* ve kterých se dále využívá terminálů ID a NUM a též tokenů z lexikální analýzy.

# Teoretický a technický návrh řešení

V této části budou popsány definice z lexikální a syntaktické analýzy

## Lex

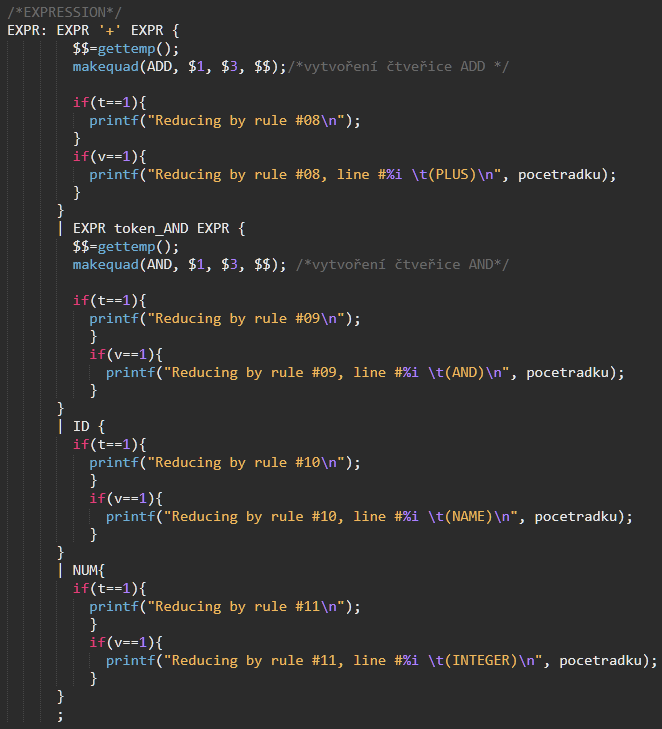
Tato část má obsažené deklarace knihoven pro přístup ke vstupním výstupním operacím a pomocným deklaracím ze souboru *sp.h*. Také jsou zde definovány deklarace pro tokeny, proměnné, číselné hodnoty, ignorování mezer a ošetření komentářů.



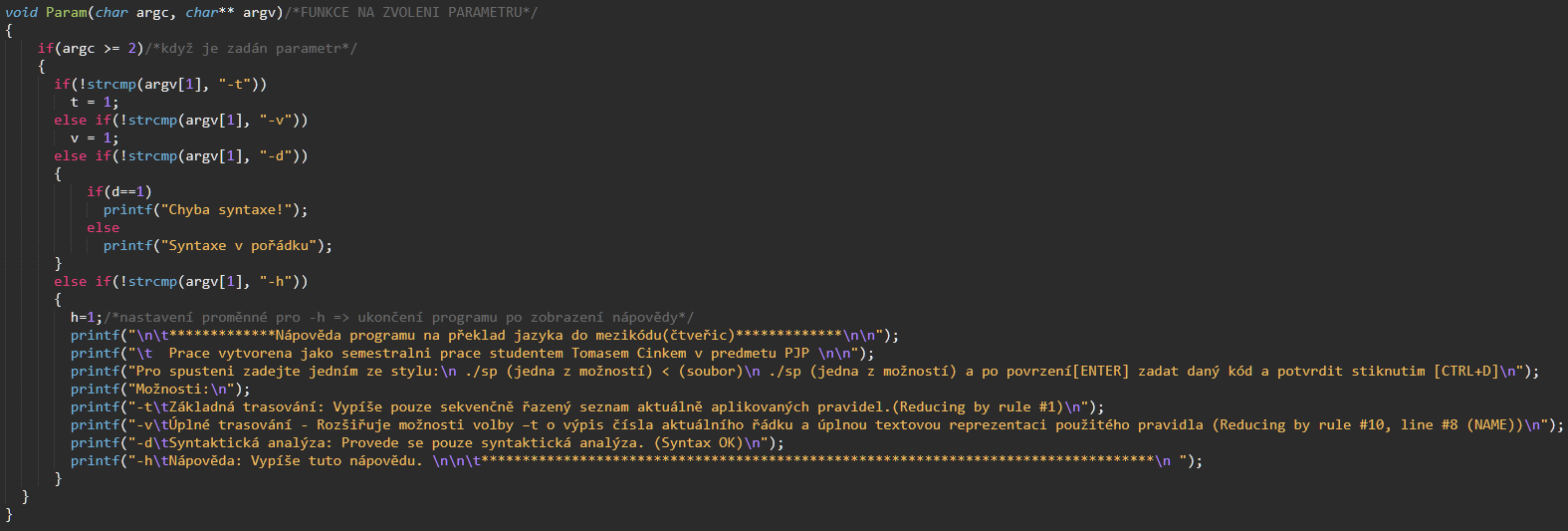
Obrázek - Lexikální analýza (vlastní)

## Yacc

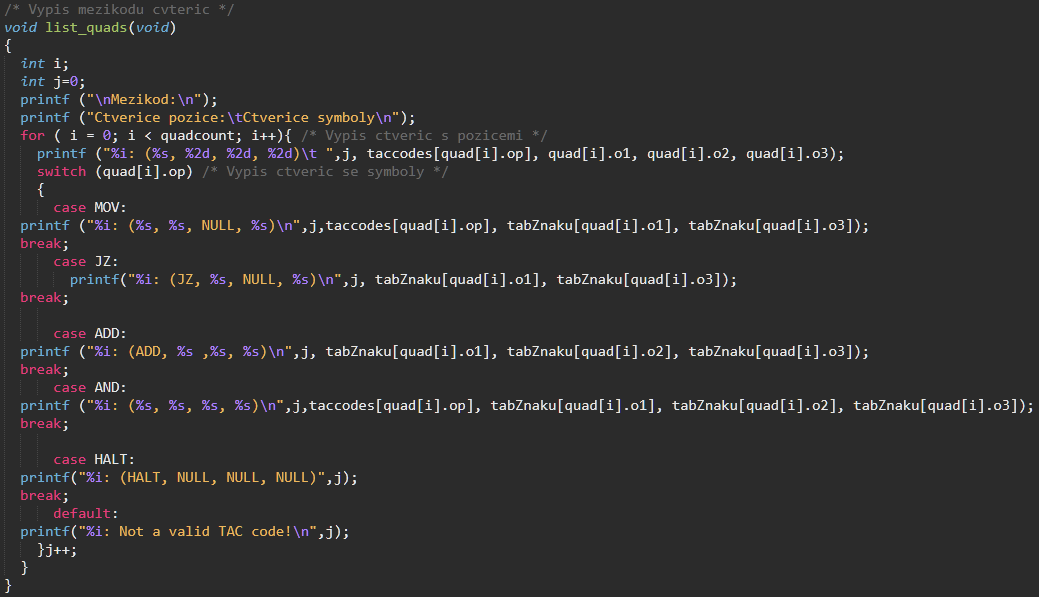
V tomto souboru jsou definovány potřebné knihovny, pomocné proměnné, gramatiku definovanou v zadání, zachycení chyb, ošetření parametrů pro vstup a výpis mezikódu a tabulky symbolů.



Obrázek - Ukázka ze souboru sp.y (vlastní)



Obrázek - Práce s parametrem



Obrázek - Výpis čtveřic s pozicemi a symboly

## Makefile

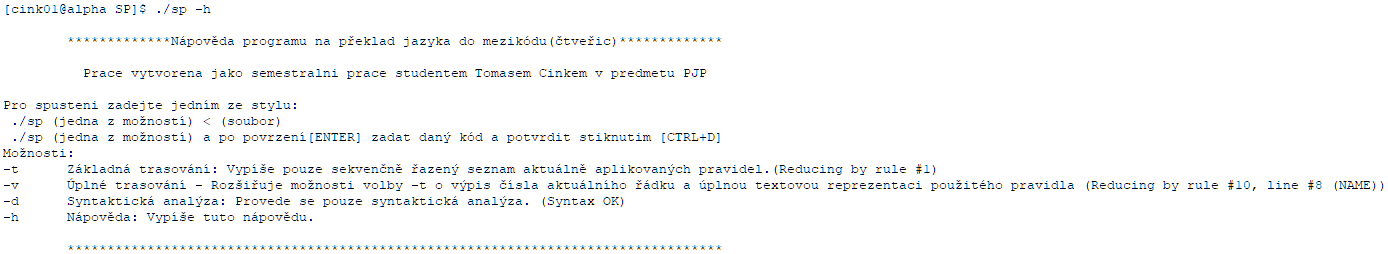
Pro zjednodušení kompilace spustitelného souboru slouží soubor *Makefile*, který provede všechny potřebné příkazy za nás. Dále obsahuje přidané příkazy jako *make help*. Který zobrazí nápovědu ke spuštění programu.



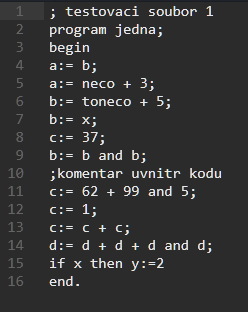
Obrázek - Makefile (vlastní)

# Testování

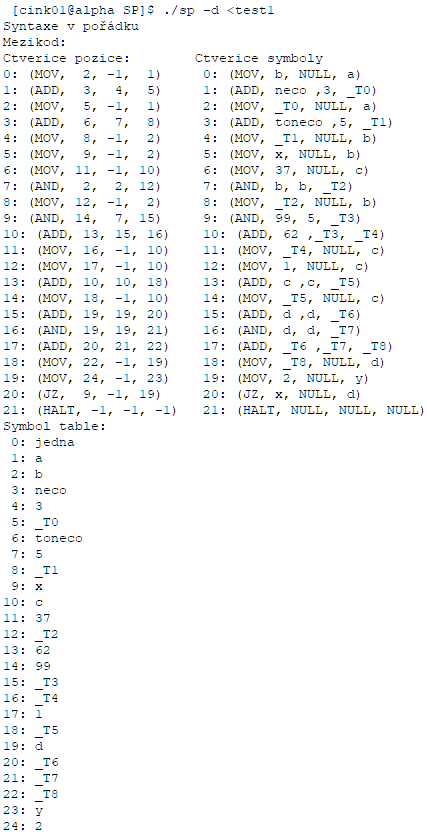
Pro testování jsou vytvořeny tři testovací soubory, které budou využiti k testování funkčnosti a ukázce práce s parametry.



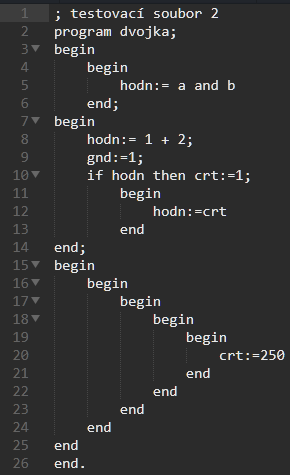
Obrázek - Výpis nápovědy



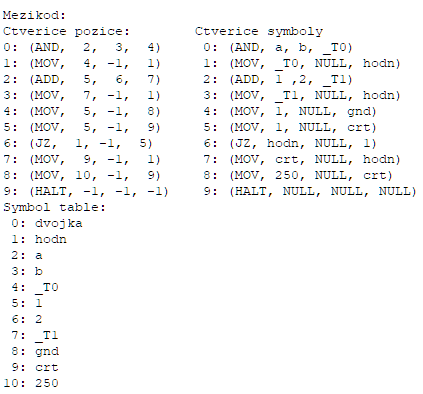
Obrázek - Testovací soubor 1



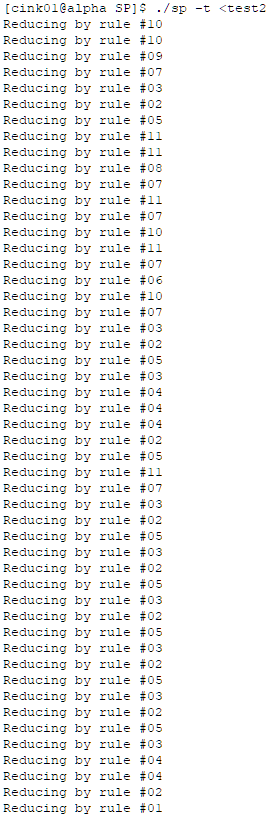
Obrázek - Test č.1 mezikód (vlastní)



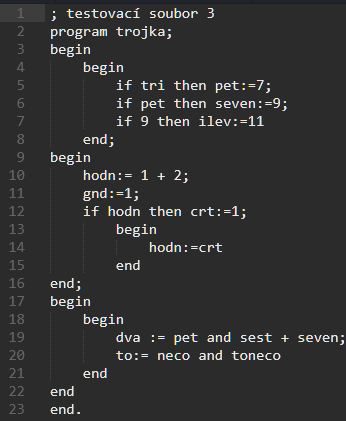
Obrázek - Testovací soubor 2 (vlastní)



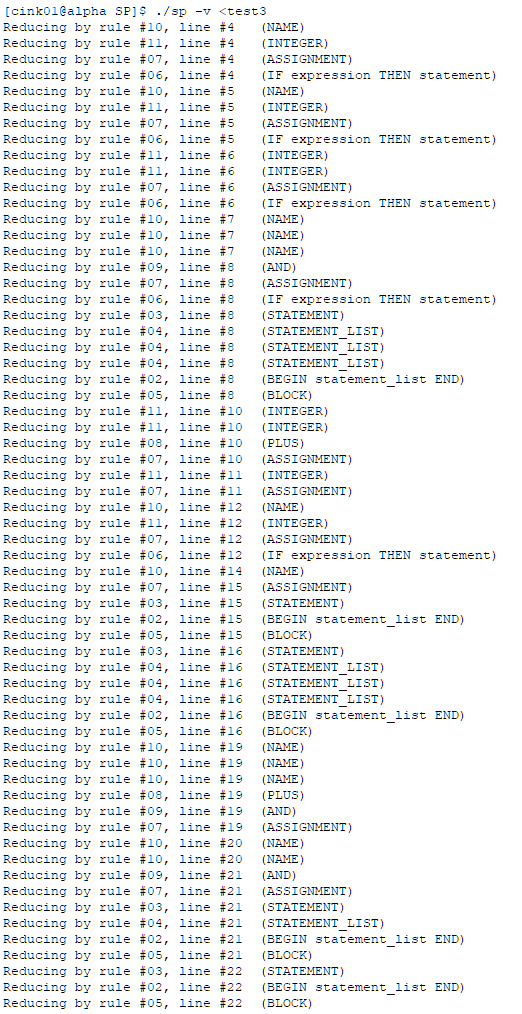
Obrázek - Test č.2 mezikód (vlastní)



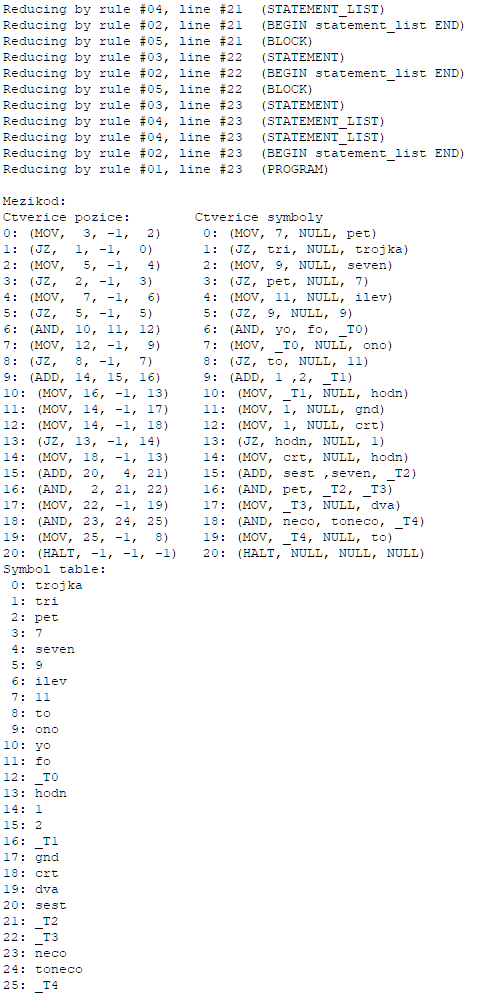
Obrázek - Test 2 základní trasování (vlastní)



Obrázek - Testovací soubor 3 (vlastní)



Obrázek - Úplné trasování testu 3 (vlastní)



Obrázek - Test č. 3 mezikód (vlastní)

# Závěr