

Implementační dokumentace k 2. úloze do IPP 2021/2022

Jméno a příjmení: Tomáš Juhász

Login: xjuhas04

interpret.py

Interpret při spuštění jako první kontroluje spouštěcí parametry. Implementovaný je `--help`, který v krátkosti popíše funkci skriptu a jako ho spouštět, `--source` sloužící na určení souboru se zdrojovým xml kódem a `--input` sloužící na určení souboru se vstupem pro instrukce.

Ak source soubor nebyl zadán parametrem kód se získá ze standardního vstupu. Kód je následně rozložen na jednotlivé elementy pomocí xml knihovny které jsou dále kontrolovány tak aby splňali správný formát.

Následně jsou dané elementy vloženy do třídy, jednotlivé argumenty do třídy `Argument` a potom třídy `Argument` do třídy `Instruction`. Třídy `Instruction` jsou dále vloženy do listu který se zoradí podle čísla kódu `order` instrukcí.

Po vytvoření listu se vytvoří list `labels` obsahující indexy instrukcí v instrukčním listu na který budou neskôr skokové instrukce přecházet. Do listu `labels` se zapisuje přeběhnutím instrukčního listu smyčkou a vyhledáním všech instrukcí s opcode `'LABEL'` nebo `'CALL'`, v případě `call` se indexy ukládají do zásobníka který je vložený v listu `labels`.

Dále se vytvoří třída rámců a inicializují se její počátečné hodnoty (`GF` prázdný list, `LF` prázdný zásobník, `TF` zostava nedefinovaný) a spustí se funkce zodpovědná za exekuci jednotlivých instrukcí.

Tato funkce přímo seznam instrukcí ať s indexem od kterého má smyčka instrukce vyberá, což umožňuje funkcím `jump` spouštět i jednotlivé úseky kódu pod návěštím. Po výběru instrukce z listu smyčkou se spustí funkce zodpovědná za zpracování instrukce s daným kódem opcode.

Aritmetické instrukce (`add`, `sub`, `mul`, `idiv`), porovnávací (`lt`, `gt`, `eq`), booleanové (`and`, `or`, `not`), konverzní funkce (`int2char`, `str2int`) a funkce na práci s řetězci (`concat`, `strlen`, `getchar`, `setchar`) fungují na podobném principu, kde nejprve skontrolují správnost typů, potom zpracují buď konstantu, nebo proměnnou z argumentů a vytvoří novou proměnnou do které výsledek zapisují.

Funkce pracující s rámci (`createframe`, `popframe`, `pushframe`) i kontrolují existenci nutných rámců a převádějí potřebné rámce mezi zásobníkem `LF` a listem `TF`.

Skokové funkce (`jump`, `jumpifeq`, `jumpifneq`) znovu spouštějí hlavní funkci výběru instrukcí s novým startovacím indexem (indexem instrukce od které bude list znovu přecházen), v případě `jumpifeq` a `jumpifneq` se kontroluje na základě podmínek nutnost skoku.

Funkce na práci s vstupem/výstupem (`read`, `write`, `dprint`), `read` si ověřuje existenci input souboru a následně tak jako i funkce `write` upravuje vstupní hodnoty na správný formát, `write` a `dprint` vypisují hodnotu proměnné nebo konstanty na standardní výstup nebo chybový výstup.

Zásobníkové funkce `pops` a `push` využívají tak i jako rámecový zásobník `LF`, nebo zásobník `calls` datový typ `deque()`. Zásobník využívaný funkcemi `pops/pushs` je globální.

Většina práce jednotlivých instrukcí i jako kontrola typů argumentů, získávání hodnoty proměnné z rámců a pod. je implementovaná přes menší pomocné funkce za účelem redukce redundantního kódu.

