

This is an old revision of the document!

Tema 1

- Deadline: 23.11.2018 23:55
- Data publicării: 09.11.2018, 23:55
- Responsabili:
 - Andrei Preda
 - Radu Nicolau

Enunț

Să se implementeze un program în limbaj de asamblare care efectuează evaluarea unei expresii aritmetice în formă postfixată (Reverse Polish notation) și apoi afișează rezultatul. Numerele ce apar în expresie sunt numere întregi, pe 32 de biți, iar operațiile ce se aplica lor sunt: +, -, /, *.

Implementare

Programul va folosi ca input un string în care se afla expresia postfixată. Această expresie se va citi de la tastatura folosind macro-ul GET_STRING.

Găsiți în [arhiva cu resursele temei](#) un fișier schelet de la care puteți începe implementarea.

Exemplu rulare:

```
$.tema1
-200 150 60 + 10 - /
-1
$.tema1
300 400 + 10 *
7000
```

Testare

Tema se poate testa pe platforma vmchecker sau local folosind checker-ul din [arhiva cu resursele temei](#).

Arhiva conține o serie de input-uri în directorul inputs/input*.txt și rezultatele așteptate pentru fiecare test, în directorul outputs. Verificarea acestor teste este făcută automat de către checker. **Nu folosiți ca input fișierele infix*.txt.**

Fișierul README din arhivă conține instrucțiuni despre folosirea checker-ului.

Trimitere și notare

Temele vor trebui încărcate pe platforma [vmchecker](#) (în secțiunea IOCLA) și vor fi testate automat. Arhiva încărcată trebuie să fie o arhivă .zip care să conțină:

- fișierul sursă ce conține implementarea temei, denumit `tema1.asm`
- fișier README ce conține descrierea implementării

Punctajul final acordat pe o temă este compus din:

- punctajul obținut prin testarea automată de pe vmchecker - 80%
- coding style - 10%. Se va ține cont de:
 - claritatea codului
 - indentare coerentă
 - comentarii
 - **nume sugestive** pentru label-uri
- fișier README - 10%

Temele care nu trec de procesul de asamblare (*build*) nu vor fi luate în considerare.

Search

- Anunțuri
- Bune practici
- Calendar
- Catalog
- Feed RSS
- IOCLA Need to Know
- Reguli și notare
- Resurse utile

Cursuri

- Curs 01: Introducere
- Curs 01 - 02: Arhitectura sistemelor de calcul
- Curs 02 - 03: Arhitectura x86
- Curs 04: Reprezentarea datelor în sistemele de calcul
- Curs 05: Reprezentarea datelor în sistemele de calcul - C2
- Curs 06 - 07: Setul de instrucțiuni
- Curs 07: Declararea datelor
- Curs 08 - 09: Moduri de adresare
- Curs 09: Stiva
- Curs 10 - 11: Funcții
- Curs 12: C + asm
- Curs 13: Unelte, utilitare
- Curs 13 - 15: Buffer overflows, securitate
- Curs 16 - 17: Optimizări
- Curs 18 - 19: Virgulă flotantă

Laboratoare

- Laborator 01: Introducere
- Laborator 02: Toolchain
- Laborator 03: First baby steps
- Laborator 04: Rolul registrelor, adresare directă și bazată
- Laborator 05: Structuri, vectori. Operații pe șiruri
- Laborator 06: Lucrul cu stiva
- Laborator 07: Apeluri de funcții
- Laborator 08: Interacțiunea C-assembly
- Laborator 09: Analiza statică și dinamică a



Mașina virtuală folosită pentru testarea temelor de casă pe vmchecker este descrisă în secțiunea [Mașini virtuale](#) din pagina de resurse.

Precizări suplimentare

- [Aici](#) puteți găsi un cheatsheet, recomandări, o serie de bug-uri frecvente, etc.
- Inputul citit de la tastatură este valid (nu se efectuează împărțiri la 0, nu se citesc caractere diferite de [0-9] și "-+/*")
- Operanzii pot avea valori negative, va trebui să folosiți [imul](#) pentru înmulțire și [idiv](#) și [cdq](#) pentru împărțire.
- Răspunsul oricărei operații nu va depăși 31 de biți (nu va face seta flag-ul de Overflow)
- În cazul operației de împărțire, doar câtul va fi luat în considerare. Spre exemplu, pentru input-ul '7 6 / 2 +' programul va trebui să afișeze 3:

```
$.tema1
7 6 / 2 +
3
```

- Dacă întâmpinați probleme în rularea modului de debug în SASM, asigurați-vă că fișierul `~/gdbinit` nu conține următoarea linie `source ~/peda/peda.py`.

Resurse

Arhiva ce conține checker-ul, testele și fișierul de la care puteți începe implementarea este [aici](#).

[programelor. GDB](#)

- [Laborator 10: Gestiunea bufferelor. Buffer overflow](#)
- [Laborator 11: Optimizări](#)
- [Laborator 12: Calcul în virgulă mobilă](#)

Teme

- [Tema 1 - Prefix AST](#)
- [Tema 2 - Stegano](#)
- [Tema 3 - Exploit ELF's, not elves](#)

Table of Contents

- [Tema 1](#)
 - [Enunț](#)
 - [Implementare](#)
 - [Testare](#)
 - [Trimitere și notare](#)
 - [Precizări suplimentare](#)
 - [Resurse](#)

iocla/teme/tema-1.1541931399.txt.gz · Last modified: 2018/11/11 12:16 by radu.nicolau

Old revisions

Media Manager Back to top

BY-SA CHIMERIC DE W3C CSS DOKUWIKI GET FIREFOX RSS XML FEED W3C XHTML 1.0