**UNIVERSITATEA DE MEDICINĂ, FARMACIE, ȘTIINTE ȘI TEHNOLOGIE “GEORGE EMIL PALADE” DIN TÂRGU MUREȘ**

**FACULTATEA DE INGINERIE**

**Program de studiu: INFORMATICĂ**

**LUCRARE DE LICENȚĂ**

**Mediu de Simulare pentru Algoritmi în Limbaj de Asamblare**

**Coordonator științific:** Conf. dr. ing. Haller Piroska

**Asistent:** Asistent drd. Bolboacă Roland

**Absolvent:** Gorea Tudor-Andrei

?

**Cuprins:**

(completez la final cand stiu si paginile)

1. Abstract ....................................................................................
2. Introducere ...............................................................................
3. Aspecte Teoretice .....................................................................
4. Aspecte de Proiectare ...............................................................
5. Aspecte de Implementare .........................................................
6. Functionalități și Caracteristici ................................................
7. Concluzii ..................................................................................
8. Bibliografie ..............................................................................

**ABSTRACT**

Limbajele de Asamblare sunt un instrument foarte puternic în programarea *low-level* a calculatoarelor. Oferind control la nivelul *hardware*, aceste limbajele sunt o opțiune foarte bună în optimizarea performanței aplicațiilor, în ingineria inversă a programelor, în scopul analizei securității acestora, și în înțelegerea arhitecturii calculatoarelor. Cu toate acestea, limbajele de Asamblare sunt foarte dificil de învățat, mai ales pentru începători, aceste limbaje având o sintaxă dificilă, lipsindu-le abstracția limbajelor de nivel mai înalt, fiind necesară implementarea manuală a mai multor sarcini care sunt gestionate automat în alte limbaje (ex: if, while, for, gestionare funcții), precum și îngreunarea depanării cauzată de lipsa constructelor de nivel înalt. În această lucrare vom prezenta un mediu de simulare pentru algoritmi în limbaj de asamblare, folosind limbajul de programare MyASM. Mediul este conceput pentru a simplifica experiența de învățare, fiind ușor de utilizat pentru scrierea și rularea codului, cu caracteristici precum rularea normală, cu posibilitatea de a fi întreruptă, rularea pas cu pas a instrucțiunilor, vizualizarea regiștrilor, a memoriei și a stivei în cadrul rulării pas cu pas și posibilitatea de a oferi date de intrare și de a vizualiza datele de ieșire. Limbajul de programare MyASM, bazat în mare pe limbajul de asamblare x86, a fost creat cu scopul de a ușura și de a scoate în evidență elementele de bază ale programării în limbaj de asamblare. Implementarea aplicației noastre este construită cu ajutorul .NET Framework, folosind limbajul de programare C#, conținând un modul WinForms pentru interfața vizuală, un modul pentru compilator, și un modul de Unit Testing pentru testarea corectitudinii compilatorului. Deși aplicația nu a fost testată de un public mai larg, credem că are potențialul de a fi un instrument valoros pentru învățarea limbajelor de asamblare.

**Cuvinte cheie:** Assembly, MyASM, Compilator, Simulator

**INTRODUCERE**

Limbajele de Asamblare sunt un tip important de limbaje de programare, fiind folosite pentru a scrie instrucțiuni care pot fi executate direct de procesorul unui computer. Aceste limbaje au o diverse aplicații, precum: programarea *low-level*, control la nivel *hardware*, optimizări de performanță, înțelegerea arhitecturii calculatoarelor și a procesoarelor, inginerie inversă pentru a găsi vulnerabilități de securitate, sau mentenanța sistemelor scrise în acest tip de limbaj. Cu toate acestea, procesul de învățare al unui limbaj de asamblare poate fi un proces dificil, mai ales pentru începători în domeniul programării, acestea având o sintaxă dificilă, precum și lipsa abstracției limbajelor de nivel mai înalt, dar și depanarea îngreunată din cauza acestui lips.

În această lucrare vom prezenta un mediu de simulare pentru algoritmi în limbaj de asamblare, folosind limbajul de programare MyASM. Mediul este conceput pentru a simplifica experiența de învățare, fiind ușor de utilizat pentru scrierea și rularea codului sursă MyASM, cu caracteristici precum rularea normală, cu posibilitatea de a fi întreruptă, rularea pas cu pas a instrucțiunilor, vizualizarea regiștrilor, a memoriei și a stivei în cadrul rulării pas cu pas și posibilitatea de a oferi date de intrare și de a vizualiza datele de ieșire. Limbajul de programare MyASM, bazat în mare pe limbajul de asamblare x86, a fost creat cu scopul de a ușura și de a scoate în evidență elementele de bază ale programării în limbaj de Asamblare. Implementarea aplicației noastre este construită cu ajutorul *Framework*-ului .NET, folosind limbajul de programare C#, folosind mediul de dezvoltare Visual Studio. Aplicația este împărțită în trei module: un modul WinForms pentru interfața vizuală, un modul pentru compilator, și un modul de *Unit Testing* pentru testarea corectitudinii compilatorului.

În capitolele următoare vom prezenta etapele dezvoltării și ciclul de viață al limbajului de programare MyASM, alături de mediul de simulare care permite rularea codului sursă MyASM și vizualizarea regiștrilor și a memoriei operative.

**ASPECTE TEORETICE**

(Assembly, Compilatoare, C#)

**ASPECTE DE PROIECTARE**

(Limbajul MyASM)

**ASPECTE DE IMPLEMENTARE**

(MyASM IDE, ciclul compilarii, ciclul rularii)

**FUNCTIONALITĂȚI ȘI CARACTERISTICI**

(user manual)

**CONCLUZII**

(nu am nici cea mai mica idee)

**BIBLIOGRAFIE**

(va contine multe link-uri de youtube)