



UNIVERSIDADE ESTADUAL DO PARANÁ - CAMPUS APUCARANA

NOME DO(S) ALUNO(S)
Gabriel Ricetto Da Rocha

SIMULADOR DE CICLOS DE INSTRUÇÕES



APUCARANA – PR
2024

Gabriel Ricetto Da Rocha

SIMULADOR DE CICLOS DE INSTRUÇÕES

Trabalho apresentado à disciplina de
Arquitetura e Organização de Computadores,
do curso de Bacharelado em Ciência da
Computação.

Professor: Guilherme Nakahata

APUCARANA – PR 2024

SUMÁRIO

INTRODUÇÃO	3
CAPÍTULO 1: OBJETIVO	4
CAPÍTULO 2: MOTIVAÇÕES E RECURSOS UTILIZADOS	
2.1 Estrutura de dados	5
2.2 Linguagens da programação	5
2.3 Bibliotecas	5
CAPÍTULO 3: RESULTADOS	
3.1 Testes na prática	6
CONCLUSÃO	8
REFERÊNCIAS	9

INTRODUÇÃO

O relatório visa descrever a implementação de um programa em C para simular a execução de operações aritméticas e de controle utilizando um conjunto de códigos operacionais (OpCodes). O programa permite a inserção de valores, execução de operações, exibição de instruções e dados de memória, além de opções para limpar os dados e sair do programa. A seguir, apresentamos uma análise detalhada baseada no conteúdo do arquivo PDF fornecido e a implementação do código.

CAPÍTULO 1

OBJETIVO

1. O objetivo principal do programa é fornecer uma interface interativa para executar operações baseadas em OpCodes, permitindo ao usuário inserir instruções e operar sobre dados armazenados em posições de memória. Este tipo de programa é fundamental para a compreensão prática de conceitos teóricos apresentados na disciplina de Linguagens Formais, Autômatos e Computabilidade.

CAPÍTULO 2

MOTIVAÇÕES E RECURSOS UTILIZADOS

2.1 Estrutura de Dados

O programa utiliza uma estrutura de dados simples em C, com um array de estruturas `Valores` para armazenar os OpCodes e operandos. Cada estrutura contém:

- `OpCode`: um string representando o código da operação.
- `Operando1` e `Operando2`: inteiros representando os operandos da operação.

2.2 Linguagens de Programação

A escolha da linguagem C foi motivada pela necessidade de controle de baixo nível sobre os recursos de memória e eficiência na execução.

2.3 Bibliotecas

As bibliotecas padrão do C utilizadas são:

- `stdio.h`: para funções de entrada e saída (e.g., `printf` e `scanf`).
- `stdlib.h`: para alocação dinâmica de memória.
- `string.h`: para manipulação de strings (e.g., `strcpy`, `strcmp`)

CAPÍTULO 3

RESULTADOS

3.1 Testes na Prática

O código permite ao usuário inserir até 50 instruções com OpCodes e operandos. As operações suportadas incluem carregamento e armazenamento em memória, operações aritméticas básicas (adição, subtração, multiplicação, divisão), raiz quadrada, negação do valor armazenado no registrador base (MBR), e instruções de controle de fluxo como saltos condicionais e incondicionais. A tabela de instruções no código fornece uma visão clara das operações suportadas:

```
=====
=      Opcoes      =
=====
=      1-Inserir    =
=      2-Ver instrucao =
=      3-Ver Dados  =
=      4-Executar   =
=      5-Limpar     =
=      6-Sair       =
=====

opcao: 1
Digite o OpCode, Operando1 e Operando2 (opcional): 000010 251 5
Digite o OpCode, Operando1 e Operando2 (opcional): 000010 252 10
Digite o OpCode, Operando1 e Operando2 (opcional): 000010 253 15
Digite o OpCode, Operando1 e Operando2 (opcional): 000001 251 0
Digite o OpCode, Operando1 e Operando2 (opcional): 000011 252 0
Digite o OpCode, Operando1 e Operando2 (opcional): 000011 253 0
Digite o OpCode, Operando1 e Operando2 (opcional): 001111 254 0
Digite o OpCode, Operando1 e Operando2 (opcional): 001100 0 0
Codigo de 'No operation' reconhecido
```

```
=====
=      Opcoes      =
=====
=      1-Inserir   =
=      2-Ver instrucao =
=      3-Ver Dados  =
=      4-Executar   =
=      5-Limpar     =
=      6-Sair       =
=====
opcao: 4
=====
=      Opcoes      =
=====
=      1-Inserir   =
=      2-Ver instrucao =
=      3-Ver Dados  =
=      4-Executar   =
=      5-Limpar     =
=      6-Sair       =
=====
opcao: 3
=====
=      Dados de Memoria      =
=====
Posicao 251: 5
Posicao 252: 10
Posicao 253: 15
Posicao 254: 30
```

Funções Principais do Código

1. `Limpar()`: Inicializa o array de estruturas, zerando os campos.
2. `Execucao()`: Executa as operações baseadas nos OpCodes e operandos inseridos.
3. `InsercaoValores()`: Permite a inserção de valores pelo usuário.
4. `exibirTabela()`: Exibe a tabela de instruções suportadas.
5. `exibirDados()`: Exibe os dados armazenados na memória.
6. `opcoes()`: Exibe o menu de opções para o usuário.
- 7.

CONCLUSÃO

O programa implementado fornece uma plataforma prática para a execução de operações aritméticas e de controle utilizando OpCodes. A estrutura do código é clara e permite fácil expansão para incluir novas operações ou funcionalidades. A utilização de conceitos básicos de autômatos e linguagens formais torna este projeto uma ferramenta educativa valiosa, alinhada com os objetivos da disciplina de Linguagens Formais, Autômatos e Computabilidade.

REFERÊNCIAS

github:<https://github.com/GuilhermeNakahata/UNESPAR-2024/blob/main/Arquitetura%20e%20Organizacao%20de%20Computadores/2%C2%B0%20Bimestre/Trabalhos/Trabalho2Bimestre%20-%20AC.pdf>