

**FUNDAÇÃO EDSON QUEIROZ**

**UNIVERSIDADE DE FORTALEZA – UNIFOR**

**CENTRO DE CIÊNCIAS TECNOLÓGICAS – CCT**

**CURSO ENGENHARIA DE COMPUTAÇÃO**

**Projeto de Pesquisa**

**TCC I**

Proposta de Arquitetura de Assembler Online Único para Diversos Processadores

Judah Holanda Correia Lima 1010500-5

Fevereiro - 2015

Judah Holanda Correia Lima

Proposta de Arquitetura de Assembler Online Único para Diversos Processadores

Projeto de pesquisa a ser apresentado e submetido à avaliação para elaboração de Trabalho de Conclusão de Curso – TCC do Curso de Engenharia de Computação do Centro de Ciências Tecnológicas da Universidade de Fortaleza.

Orientador:

Danilo Reis Vasconcelos

Fortaleza – Ceará 2013

Sumário

Resumo 4

Introdução 5

Tema 7

Delimitação do Tema 7

Objeto 7

Problema 7

Hipótese 7

Objetivos 8

Objetivo Geral 8

Objetivos Específicos 8

Justificativa 8

Discussão Teórica 8

Metodologia 9

Provável Sumário 10

Cronograma 10

Bibliografia 10

# Resumo

Montadores Assembly online têm alta performance e vêm com uma quantidade de ferramentas de auxilio comparáveis os softwares instalados em disco rígido para o mesmo propósito. Embora existam vários Assemblers online, nenhum deles utiliza o mesmo conjunto de instruções para todos os processadores que ele dá suporte. O presente estudo explora o potencial de uma arquitetura de um montador Assembly multiplataforma. E é um estudo de uma arquitetura proposta bem como o produto em si, em sua versão inicial e simplificada, sendo então também um estudo em progresso desta arquitetura e seu desenvolvimento e potencial que o produto resultante deste trabalho possa ter. Para atingir tal objetivo, a ideia proposta foi de utilizar um reduzido conjunto de instruções comum a qualquer processador, mas suficientemente grande para serem feitas todas as operações básicas. Os testes demonstraram que mesmo em estágio inicial a arquitetura é viável mesmo em um servidor rodando o montador em PHP. Então pode-se concluir que este projeto tem uma arquitetura viável e tem potencial de evolução aceitável e desempenho para os parâmetros do mercado.

# Introdução

Atualmente é possível perceber que softwares que eram um dia programas que rodam em ambiente off-line estão se tornando serviços web [1,2]. Isto possibilita o uso de diversas plataformas em qualquer lugar que possua conexão com a internet e facilita trabalhos colaborativos [3,4,5].

Para a criação de um projeto de software/firmware comercial normalmente é necessário a instalação de diversos programas, dentre eles IDE’s, programas de documentação, gerenciamento, controle de versão, entre outros. E cada um deles é direcionado para um sistema operacional específico e algumas vezes é necessário utilizar dois sistemas distintos, pois um ou mais programas não estão disponíveis para a plataforma utilizada no inicio do projeto, bem como o tempo gasto para instalação. Depois de toda esta maratona ainda existe a pesquisa de frameworks e códigos para serem reutilizados neste projeto, pois não existe um local que todos os códigos do mundo estão armazenados e de fácil acesso. Finalmente temos a integração destes códigos com o projeto que isso implica em fazer tanto a integração com a IDE como a plataforma utilizada e isso algumas vezes implica na modificação do código que deseja-se integrar ou encapsula-lo em outro que faca a compatibilidade entre os dois e isso implica em ler um código que geralmente não esta seguindo o mesmo padrão de codificação e documentação isto quando segue algum ou tem documentação.

Uma IDE online além de tornar desnecessário a instalação ela permite o uso em qualquer lugar que internet esteja disponível, independente de plataforma ou sistema operacional. E esse é um dos motivos pelos quais serviços Web tem sido tamanho sucesso e tem aumentado tanto ao longo dos anos, programas que antes precisavam ser instalados na maquina agora se tornam online, sendo necessário apenas um browser. Onde não é necessário se preocupar nem mesmo com atualização de software.

Integrando todas essas ferramentas e códigos em um mesmo serviço ira tornar a criação de projetos muito mais rápida e pratica além de aumentar significativamente o reuso de código.

Contudo, mesmo fazendo tudo isso cada família de processador tem sua própria linguagem de maquina e consequentemente um Assembly próprio o que dificulta muito a migração de uma plataforma para outra[6].

Este estudo propõe uma nova abordagem, criando um Assembly unificado que use apenas um conjunto restringindo de instruções. Assim facilitando a migração de plataformas.

Ainda assim existe um outro problema a ser resolvido, ao se aprender algo novo, normalmente o mais difícil de se assimilar é a terminologia. Então mesmo que a ideia fundamental seja simples, quando é descrita com palavras não familiares isso pode dificultar bastante um assunto [7]. Como o Assembly utiliza mnemônicos isso demanda um estudo e aprendizado de cada instrução o que torna a linguagem bem mais complexa, linguagem de alto nível como C/C++ e Java tem suas funções escrita de forma extensa diferentemente de uma abreviação/sigla como no caso dos mnemônicos, por isso usualmente se aprende primeiro linguagens que alto nível, pois elas estão mais próximas da linguagem humana [6,7]. Então propõe-se que as instruções seja escritas de forma extensa, facilitando a leitura do código e o aprendizado da linguagem.

Como vários em países têm-se um costume de se programar na língua oficial do pais. Cria-se outro problema, impede-se que pessoas que não falem a língua não consigam utilizar o código escrito em tal língua. Então para contornar este problema o sistema terá como regra a codificação em inglês, devido ser uma das línguas mais faladas no mundo, uma das mais fáceis de se aprender e ter se tornado o padrão na maioria dos projetos.

Ainda assim é necessário delimitar um padrão para nomenclatura de variáveis e métodos, pois mesmo tendo uma linguagem padrão isso não garante que os códigos serão de fácil leitura. Então um padrão de nomenclatura também deve ser adotado. Bem como de chaveamento e padrões de projetos.

# Tema

Proposta de uma Arquitetura de Assembler Online.

## Delimitação do Tema

Utilizar um conjunto de instruções ainda mais reduzido que o RISC convencional que contenha todas as operações básicas que qualquer microprocessador possua, permitindo assim um conjunto de instruções multiplataforma.

Criar um conjunto de instruções fáceis de serem lidas.

Desenhar um banco de dados para que se possa ter um banco de códigos com controle de versão básico onde o programador não tenha que ficar se preocupando com programas externos para este fim.

Criar um buscador de códigos e indexador para que seja fácil pesquisar um código e utilizá-lo de forma rápida e simples e assim facilitar o reuso de códigos.

# Objeto

## Problema

É possível utilizar/criar um Assembly Único para vários processadores de arquitetura diferente?

O processamento do montador é viável pra a linguagem PHP?

O Sistema tem uma abordagem eficiente?

## Hipótese

É possível notar que muitos processadores tem instruções semelhantes e algumas vezes até mesmo iguais. Se fosse possível encontrar um conjunto de instruções comum a todos eles e que estas funções, mesmo que reduzidas possam executar todos os comandos básicos será possível criar um Assembly comum. Ainda que a linguagem de máquina resultante ao final do processo só possa ser lida por determinado processador.

A velocidade de processamento em linguagem PHP por um Servidor em nuvem é suficientemente rápida para o Montador proposto.

A arquitetura reduzida RISC proposta é suficientemente rápida para boa parte das aplicações atuais, de tal forma que a perda de performance devido a ausência de outras instruções é justificada pela sua flexibilidade.

# Objetivos

## Objetivo Geral

Potencializar o reuso de códigos com a criação de um Assembly unificado.

## Objetivos Específicos

Criar conjunto de instruções, determinar padrão de nomenclatura a ser seguido pelo seus usuários, utilizar controle de versão e armazenamento no próprio sistema e criar sistema básico (Easembly).

# Justificativa

Mesmo já existindo linguagens multiplataforma em auto nível, não é possível utilizá-la em processadores com um poder de processamento mais modesto.

Mnemônicos dificultam a leitura e entendimento dos códigos.

Conjuntos de instruções diferentes para cada família de processadores dificulta a migração de um processador para outro.

A utilização de uma plataforma web permite o programador codificar em qualquer lugar que possua internet em uma infinidade de dispositivos sem a necessidade de instalar diversas ferramentas.

# Discussão Teórica

A ideia deste conjunto reduzido de instruções vem do sucesso da arquitetura RISC nos processadores modernos. Para tornar este conjunto de instruções multiarquitetura é necessário reduzir o numero de instruções ao máximo para que que se tenha certeza que este conjunto vai estar presente em qualquer processador.

Foi possível encontrar inúmeros trabalhos publicados onde a ideia de uma IDE Online se repete, embora não com o mesmo intuito de um Assembler unificado, mas muitas vezes utilizando uma linguagem independente de arquitetura como Java [8,9,10,11,12,13]. Foram encontrados Assemblers para processadores reconfiguráveis [14], embora não seja o foco deste estudo algumas instruções são mantidas ao reconfigurar o processador e isto será de grande utilidade pra projeta o conjunto de instruções, assim como um gerador Assembler [15].

O armazenamento de códigos, nada mais é do que banco de dados de Strings com metadados para gerenciamento dos mesmos.

# 

# Metodologia

*A metodologia utilizada para a criação desta arquitetura será feita a partir da pesquisa e estudo do estado da arte de projetos, sistemas e artigos que se assemelham ao tema e assim desenhar a arquitetura baseada neste estudo. Ao final fazer um protótipo e a validação da arquitetura será realizada a partir de um estudo de caso sobre o protótipo.*

# Provável Sumário

INTRODUÇÃO

Estrutura da monografia

1. Introdução
2. Arquiteturas Relacionadas
3. Arquitetura Proposta
4. Estudo de Caso
5. Análise dos Resultados
6. Conclusão
7. Anexo

REFERÊNCIAS BIBLIOGRÁFICAS

# Cronograma

Etapa I: revisão da bibliografia.

Etapa II: desenvolvimento do protótipo

Etapa III: análise do material coletado.

Etapa IV: redação do trabalho.

Etapa V: revisão e apresentação do trabalho final.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Jan/14 | Fev/14 | Mar/14 | Abr/14 | Mai/14 |
| Etapa I |  |  |  |  |  |
| Etapa II |  |  |  |  |  |
| Etapa III |  |  |  |  |  |
| Etapa IV |  |  |  |  |  |
| Etapa V |  |  |  |  |  |

# Bibliografia

[*Descrição de todo o material (livros, periódicos, sites, relatórios, etc) utilizado no trabalho.* ***Seguir as normas da ABNT***]

[1] CANFORA, G. Migrating interactive legacy systems to Web services. 2006.

[2] Myeon Kim, Su. E-Commerce and Web Technologies. : Springer Berlin Heidelberg, 2004.

[3] Kai-Wai Chu, Samuel. Using online collaborative tools for groups to co‐construct knowledge. 2011.

[4] Miller, Michael. Web-Based Applications That Change the Way You Work and Collaborate Online.: Que Publishing, 2008.

[5] Bourne John. Online Engineering Education: Learning Anywhere, Anytime. 2005.

[6] Hyde, Randall. The Art of Assembly Language. : No Starch Press, 2003.

[7] R. Smith, Jack. Programming the PIC Microcontroller with MBasic. : Newnes, 2005.

[8] Kuan-Cheng, Lin. An On-line Instruction/Learning Environment for Supporting Individualized Learning in Java Programming. : IEEE, 2007.

[9] More, A. Web Based Programming Assistance Tool for Novices. : IEEE, 2011.

[10] Datta, A. Online compiler as a cloud service. : IEEE, 2014.

[11] Mohammed, S. WIDE an interactive Web integrated development environment to practice C programming in distance education. : IEEE, 2013.

[12] Minzhe, Guo. Back to Results Design of Online Runtime and Testing Environment for Instant Java Programming Assessment. : IEEE, 2010.

[13] Ansari, A.N. Online C/C++ compiler using cloud computing. : IEEE, 2011.

[14] Md Salin, S.I. Back to Results One-pass Assembler Design for a Low-end Reconfigurable RISC processor. : IEEE, 2014.

[15] Arbone, C. Model-Driven Inline Assembler Generator for Retargetable Compilers. : IEEE, 2013.

[16] Hatfield, Bo. A GENERAL-PURPOSE CUSTOM-DESIGN EDASSEMBLER IN C. : IEEE, 2003.

[17] R. Tavernier, Karel. Macro-Based Cross Assemblers. : IEEE, 1980.

[18] Nakano, Koji. Processor, Assembler, and Compiler Design Education using an FPGA. : IEEE, 2008.

[19] Mathiske, Bernd. An assembler and disassembler framework for JavaTM programmers. : ScienceDirect, 2008.

[20] J. Davis, Ian. From Whence It Came: Detecting Source Code Clones by Analyzing Assembler. 2010.

[21] Md Salin, S.I. Two-pass Assembler Design for a Reconfigurable RISC Processor.: IEEE, 2013.

**10. DE ACORDO** (assinaturas dos professores que avaliaram o projeto)

Professor Orientador: Nome:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Ass: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Professor da mesma área: Nome:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Ass: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_