**Nomes:** Josué Nunes Campos – 03465

Lucas Barros Pereira Costa – 03511

Mateus Coelho Santos – 03488

**DOCUMENTAÇÃO DO MONTADOR SIMPLIFICADO**

* **Objetivo**

Este sistema tem como objetivo a implementação de um montador MIPS de maneira simplificada.

O sistema consiste na leitura de um conjunto de instruções em Assembly do MIPS e a partir da leitura, ele cria um outro arquivo com a conversão desse conjunto de instruções.

* **Projeto do Sistema e Principais Decisões**
* Organização do projeto

Inicialmente decidimos construir o montador simplificado utilizando python, por ser uma linguagem mais utilizada nos dias de hoje, porém devido ao costume de se trabalhar com a linguagem C, construímos uma segunda versão (que seria a final) em C, o que nos deu uma base comparativa mais real entre as duas linguagens, o que de certa forma acabou como um adendo para nosso aprendizado no curso.

Observamos algumas dificuldades no momento de iniciar o projeto, por mais que cada linha de instrução tenha o mesmo formato (instrução registrador 1, registrador 2, registrador 3 ou imediato), a separação de registradores e instrução eram diferentes para serem separadas usando um único split().

* Organização do código

Definimos então, como criar o montador simplificado em etapas.

Primeiro fazemos a leitura do arquivo e em seguida separamos por linha cada instrução e cada registrador ou imediato. Primeiramente, decidimos atribuir para cada instrução, seu opcode, shamt e funct em decimal, assim como os registradores, e por fim inserir esses valores numa função de conversão decimal-binário. Porém, para deixar o código mais compacto e simples, nós já atribuímos os valores em binário, tendo que converter somente os valores imediatos nas instruções do tipo-I.

* **Conclusão**

Após o término da implementação do sistema, pudemos observar como se dá o processo de conversão de uma linguagem assembly para o nível de linguagem de máquina.

Por fim, foi possível também adquirir o conhecimento prático acerca do porquê a utilização de assembly, justamente para que a máquina já interprete cada instrução de forma padronizada e tabelada, diferente do que acontece em linguagens de alto nível onde o tamanho das instruções não são sempre iguais e não possuem valores tabelados de suas variáveis e instruções.