

# Manual do Usuário

## Mips: Assembly encoder / decoder

Esse projeto tem como objetivo realizar tanto a codificação (transformar instrução assembly MIPS para hexadecimal) quanto a decodificação (transformar instrução em hexadecimal para assembly MIPS válido).

Nesse escopo, por enquanto suportamos as instruções: **or**, **and**, **sub**, **sltiu**, **lw**, **sw**, **beq** e **j**.

## Como utilizar

O programa é inteiramente desenvolvido em C e utiliza apenas as libs nativas da linguagem. Para executar o projeto (já compilado), basta executar o seguinte comando a partir do diretório raiz do projeto:

```
./converter <encode/decode>
```

O comando acima executará a codificação ou decodificação dos arquivos.

O arquivo de input para `./converter encode` é `entrada.asm`. Deixei um código de exemplo na entrada. Sua saída é no arquivo `saida.txt`, contendo as instruções codificadas para hexadecimal. Para testar outras entradas, basta modificar o conteúdo o arquivo `entrada.asm` com assembly válido e suportado pelo projeto.

Já para testar `./converter decode`, o arquivo de entrada é `entrada.txt`, contendo uma lista de códigos em hexadecimal. A saída do programa se encontra no arquivo `saida.asm`, contendo um programa válido assembly.

## Recompilando o projeto

Caso alguma alteração tenha sido feita, é necessário recompilar o programa. Se você tiver **make** instalado em sua máquina, basta rodar: **make** para recompilar o programa e depois rodar `./converter <encode/decode>` novamente.

Caso não tenha **make**, basta compilar com o gcc: `gcc -o converter io.c decoder.c main.c encoder.c helpers.c` e novamente rodar `./converter <encode/decode>`