

MipsCode

Uma ferramenta para desenvolver código em assembly para o processador MIPS

FEITO PARA VOCÊ

O MipsCode possui uma IDE voltada para assembly mips, ideal para quem precisa de um ambiente de desenvolvimento completo e integrado para a linguagem.

[Faça uma demonstração >](#)

```
1 .text
2 main:
3     addi $2, $0, 5
4     syscall
5 end:
6     addi $2, $0, 5
7     addi $2, $0, 5
8     addi $2, $0, 5 →
9     addi $2, $0, 5
```

SEJA UM EXPLORADOR



Está iniciando agora?

Podemos te ajudar a programar em assembly de forma fácil e intuitiva.



Compartilhe experiência

Uma comunidade para compartilhar conhecimento sobre programação.



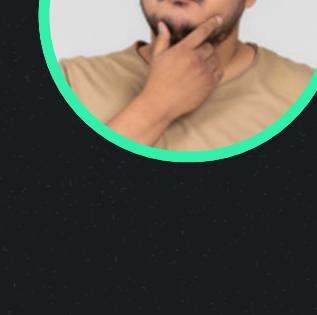
Guarde seus projetos

Você pode salvar seus projetos e acessá-los a qualquer momento.

DEPOIMENTOS

“

O site MipsCode é um ótimo recurso para aprender e praticar mips. O site fornece uma variedade de exemplos e tutoriais para ajudar os usuários a se familiarizarem com a linguagem. Eu recomendaria este site para qualquer pessoa que esteja procurando um bom recurso para aprender assembly mips.



João Pereira

Professor de Arquitetura de Computadores

● ○ ○

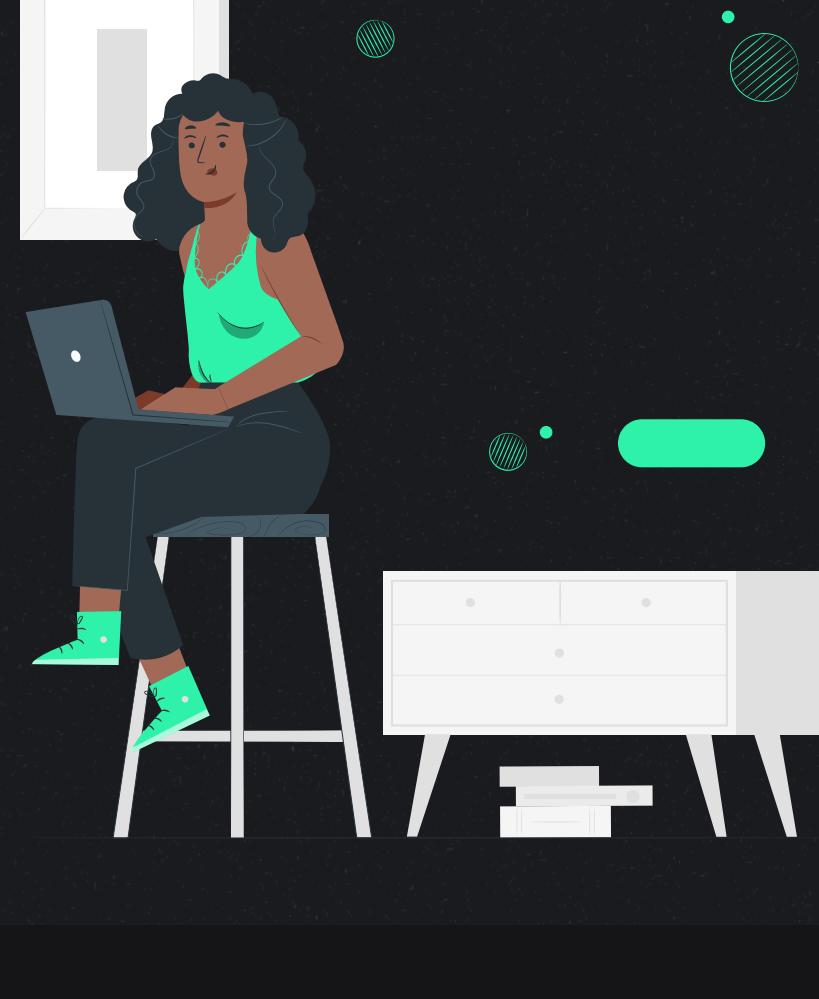
CONTATO

Alguma dúvida? Fale com a gente.

Nome *

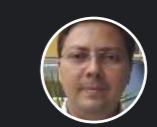
Email *

Mensagem



ENTRE NA SUA CONTA

 Email ou nome de usuário Senha[Crie sua conta](#)[Esqueceu a senha?](#) Continuar com o Google



Tutoriais

Dashboard / Tutoriais

Buscar arquivo

Fácil

Como calcular o mmc de dois valores utilizando o for.

Nível: Fácil

Ver conteúdo

Como calcular o mmc de dois valores utilizando o for.

Nível: Fácil

Ver conteúdo

Como usar o jump em Assembly Mips.

Nível: Difícil

Ver conteúdo

Como usar o jump em Assembly Mips.

Nível: Difícil

Ver conteúdo

Aprenda a usar o ADDI, SUB e MULT.

Nível: Intermédio

Ver conteúdo

Aprenda a usar o ADDI, SUB e MULT.

Nível: Intermédio

Ver conteúdo

< 1 2 3 ... 8 9 10 >

Ver todos os tutoriais

CRIE UMA NOVA CONTA

 Nome de usuário Email Senha Confirme a Senha[Cadastrar-se](#)[Já possui uma conta?](#)[!\[\]\(a8f9309f944226d1420f5fed22e2b6e6_img.jpg\) Cadastrar com o Google](#)

Arquitetura de Conjunto de Instruções MIPS

Introdução – O MIPS

Não podemos confundir Arquitetura do Conjunto de Instruções MIPS com MIPS, Milhões de Instruções por Segundo. O MIPS que tratamos aqui é uma Arquitetura de Conjunto de Instruções (Instruction Set Architecture – ISA), desenvolvida pela empresa MIPS Computer Systems, que hoje é chamada de MIPS Technologies. MIPS significa Microprocessor Without Interlocked Pipeline Stages (Microprocessador Sem Estágios Intertravados de Pipeline).

A empresa foi fundada em 1984 por um grupo de pesquisadores da Universidade de Stanford e o foco era os microprocessadores com Arquitetura RISC (falarei sobre esse assunto na série de Artigos sobre Microprocessadores). John Leroy Hennessy fez parte da fundação da empresa, assim como da História da evolução dos Microprocessadores, o que pode ser comprovado com a leitura de seus livros e artigos científicos. Vários equipamentos utilizaram microprocessadores MIPS como, por exemplo, o Nintendo 64, Sony PlayStation, Roteadores Cisco, etc.

Conjunto de Instruções

Antes de falarmos especificamente sobre o MIPS, vamos discutir um pouco sobre Conjunto de Instruções. Como bem sabemos, todo Sistema Computacional é composto, muito basicamente, por Entrada, Saída, Processamento e Armazenamento. Cada um desses subsistemas pode ser organizado de formas diferentes (organização) no sistema, e cada elemento que faz parte desses subsistemas pode ser projetado também de formas diferentes (arquitetura).

O Conjunto de Instruções é um dos elementos desse grande sistema, e é de extrema importância para a construção de um sistema computacional. Um Processador não é exatamente um dispositivo único, ele é um conjunto de sistemas, cada um responsável por executar determinadas ações. O que temos, na verdade, é uma CPU – Unidade Central de Processamento – composta pela Unidade de Controle, Unidade Lógica Aritmética, entre muitas outras UNIDADES FUNCIONAIS necessárias para realizar o processamento de qualquer tipo de dados que precisamos.

Aí é que está o “X” da questão! Por exemplo, se um microprocessador não é capaz de executar uma soma em ponto flutuante, então este computador não poderá processar determinados tipos de dados, programas, etc. O computador ficará limitado, o que nos dias atuais não é nada interessante. Portanto, quando se PROJETA um novo microprocessador, primeiro é necessário definir que tipo de instruções, dados e programas ele será capaz de executar.

Além disso, precisa-se manter a compatibilidade com microprocessadores anteriores. O novo microprocessador deve ser capaz de continuar executando os seus softwares. Não é uma ideia interessante lançar um microprocessador com muitas inovações se os usuários não puderem mais usar os softwares que estão acostumados. É claro que, um dia, logo mais à frente, a tecnologia vai mudar e, de certa forma, nos veremos na obrigação de evoluir. Os transistores, a tecnologia atual de fabricação de computadores, está em seu limite, e muitos pesquisadores estão buscando novas matérias primas para construção de processadores cada vez mais rápidos. Quando uma nova tecnologia surgir, nos encontraremos exatamente nesse ponto de evolução.

Os nossos computadores, notebooks e celulares atuais são dispositivos de uso geral. Isso significa que eles precisam ter capacidade de processamento para diversos tipos de dados diferentes, que variam desde um texto simples até um vídeo em três dimensões. É diferente de um controle remoto de TV ou um Microondas, que são projetados para um fim específico. Dessa forma, as instruções que o microprocessador é capaz de executar formam o conjunto de instruções.

Algumas instruções que o microprocessador de uso geral pode executar são:

- operações aritméticas;
- operações lógicas;
- operações relacionais;
- operações de ponto flutuante;
- transferência de dados;
- desvios condicionais;
- desvios incondicionais;
- controle;
- entre outras.

Arquitetura de Conjunto de Instruções MIPS

[Primeira Instrução MIPS](#)

[Compilação de Expressões no MIPS](#)

[Convertendo uma instrução com Array no MIPS](#)

[Armazenando um valor em Array no MIPS](#)

[Instruções LW e SW com Array no MIPS](#)

[Instrução IF Simples no MIPS](#)

[Instrução IF Composto no MIPS](#)

[Instrução SLT no MIPS](#)

[Operações Lógicas no MIPS](#)

[Próximo passo >](#)

VEJA O QUE VOCÊ PODE REALIZAR NO MIPS

APRENDA[Explorar Documentação >](#)**CRIE AGORA**[Programe com Mips >](#)**COMPARTILHE** (Discord)[Entre na comunidade >](#)

MEU REPOSITÓRIO

Guarde seus projetos e accesse onde estiver.

[!\[\]\(097cdd6c9c875b64d9b8c9a2409491c4_img.jpg\) Criar novo projeto](#)[!\[\]\(f9f168a9979beed8b01f8750d577d508_img.jpg\) Ver todos os projetos](#)

Projetos recentes

Multiplicação por 2.

3 horas atrás

Projeto que realiza a multiplicação de um valor por 2.

[Abrir projeto](#)

Multiplicação por 2.

3 horas atrás

Projeto que realiza a multiplicação de um valor por 2.

[Abrir projeto](#)

Multiplicação por 2.

3 horas atrás

Projeto que realiza a multiplicação de um valor por 2.

[Abrir projeto](#)

TUTORIAIS

Aprenda de maneira fácil e prática.

Como calcular o mmc de dois valores utilizando o for.

Nível: Fácil

[Ver conteúdo](#)

Como usar o jump em Assembly Mips.

Nível: Difícil

[Ver conteúdo](#)

Aprenda a usar o ADDI, SUB e MULT.

Nível: Intermédio

[Ver conteúdo](#)

Como calcular o mmc de dois valores utilizando o for.

Nível: Fácil

[Ver conteúdo](#)

Como usar o jump em Assembly Mips.

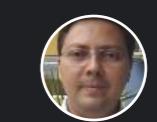
Nível: Difícil

[Ver conteúdo](#)

Aprenda a usar o ADDI, SUB e MULT.

Nível: Intermédio

[Ver conteúdo](#)[!\[\]\(13d2932398567322129c582c81e3ad37_img.jpg\) Ver todos os tutoriais](#)



Meu Repositório

Dashboard / Meu Repositório

[Criar novo arquivo](#)[Importar arquivo](#)[Mais Recentes](#)

Multiplicação por 2.

3 horas atrás

Projeto que realiza a multiplicação de um valor por 2.

[Abrir arquivo](#)

Multiplicação por 2.

3 horas atrás

Projeto que realiza a multiplicação de um valor por 2.

[Abrir arquivo](#)

Multiplicação por 2.

3 horas atrás

Projeto que realiza a multiplicação de um valor por 2.

[Abrir arquivo](#)

Multiplicação por 2.

3 horas atrás

Projeto que realiza a multiplicação de um valor por 2.

[Abrir projeto](#)

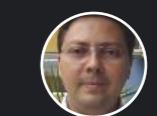
Multiplicação por 2.

3 horas atrás

Projeto que realiza a multiplicação de um valor por 2.

[Abrir arquivo](#)

< 1 2 3 ... 8 9 10 >



Tutoriais

Dashboard / Tutoriais / Como calcular o mmc de dois valores utilizando o for.

Como calcular o mmc de dois valores utilizando o for.

a instrução "syscall" é um dos pilares da execução do assembly, ela atua como uma notificação ao processador de que ele deve executar uma ação.

O \$2 é o registrador encarregado de informar ao processador qual ação deve tomar. Quando o syscall for executado o processador irá olhar para o conteúdo armazenado neste registrador.

Por exemplo o valor inteiro 5 sinaliza que o dispositivo de entrada(teclado) deve receber um valor inteiro do usuário, e armazenar este valor no próprio \$2.

```
reg2 = int(input()) # escreva um valor inteiro
```

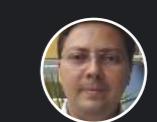
[Veja no Terminal](#)[Copiar código](#)

O valor inteiro 1 sinaliza que o dispositivo de saída(monitor) deva retornar um valor.

Clique duas vezes (ou pressione "Enter") para editar

Mas antes devemos falar sobre o \$4. Este registrador é o responsável por armazenar o valor que será retornado pelo dispositivo de saída.

[Tutorial anterior](#)[Próximo tutorial](#)

 Encontre nas configurações...

Configurações

[Dashboard](#) / [Configurações](#) / [Informações Básicas](#)

Informações Básicas

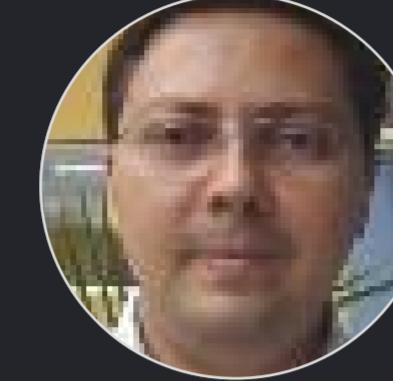
Perfil do usuário

Logon e Segurança

Preferências

Exibição

Perfil do usuário

[Enviar nova foto](#) [Remover foto](#)**Nome**

Gilbert Azevedo

**Biografia**

Sou Gilbert Azevedo, um programador apaixonado pelo que faço. Tudo começou quando eu tinha 12 anos e ganhei meu primeiro computador. A partir daí, não parei mais de aprender. Sempre fui muito curioso e gostava de descobrir como as coisas funcionavam. Isso me ajudou muito na minha carreira, pois me permitiu ter um olhar crítico e analítico sobre o código e o funcionamento dos sistemas. Hoje, trabalho como programador freelance e também sou professor de programação. Amo ensinar aos outros o que sei e ajudar a formar novos profissionais. Acredito que a programação é uma das áreas mais dinâmicas e interessantes da tecnologia, e por isso mesmo, é um constante desafio. E eu adoro um bom desafio!

140 / 140 caracteres

[Salvar](#)

Logon e Segurança

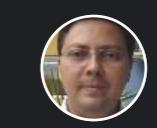
Email

gilbertazevedo@gmail.com

Confirmação de senha

[Cancelar](#)[Atualizar](#)**Senha**

[Solicitar exclusão de conta](#)



Q Encontre nas configurações...

Configurações

Dashboard / Configurações / Preferências

Informações Básicas

Notificações

Logon e Segurança

Preferências

Exibição

Exibição

Tema

Escuro



Tamanho da fonte do terminal

Médio



Idioma

Português do Brasil





```
01 .text
02
03 main:
04     addi $2, $0, 5
05     syscall
06     addi $3, $0, 5
07     syscall
08 end:
09     add $4, $2, $3
10     addi $2, $0, 1
11     syscall
```

Console

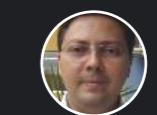
1:10 PM -- Crogram is finished running (dropped off bottom) --
1:15 PM Assemble: operation completed successfully.
1:26 PM Digite uma entrada:
1:48 PM Existe um erro no seu código

```
01      .text
02
03      main:
04          addi $2, $0, 5
05          syscall
06          addi $3, $0, 5
07          syscall
08      end:
09          add $4, $2, $3
10          addi $2, $0, 1
11          syscall
```



Console

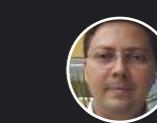
Registradores



```
01 .text
02
03 main:
04     addi $2, $0, 5
05     syscall
06     addi $3, $0, 5
07     syscall
08 end:
09     add $4, $2, $3
10     addi $2, $0, 1
11     syscall
```

Registradores

Identificação	Nome	Valor
0	\$zero	0
1	\$at	0
2	\$v0	1
3	\$v1	5
4	\$a0	10
5	\$a2	0
6	\$a3	0
7	\$t0	0
8	\$t1	0
9	\$t2	0
10	\$t3	0
11	\$t4	0
12	\$t5	0



Tutoriais

[Dashboard](#) / [Tutoriais](#)

[Criar novo tutorial](#)

Como calcular o mmc de dois valores utilizando o for.

[Ver conteúdo](#)

Como usar o jump em Assembly Mips.

[Ver conteúdo](#)

Aprenda a usar o ADDI, SUB e MULT.

[Ver conteúdo](#)

Como calcular o mmc de dois valores utilizando o for.

[Ver conteúdo](#)

Como usar o jump em Assembly Mips.

[Ver conteúdo](#)

Aprenda a usar o ADDI, SUB e MULT.

[Ver conteúdo](#)

< [1](#) [2](#) [3](#) ... [8](#) [9](#) [10](#) >