

Documentação TP01 Montador Risc-V

Larissa Ribeiro Brum
23.2.8023

Ithan de Paula Amaral
23.2.8065

Junho 2025

1 Instalação

1.1 Instalação do Python no Linux

Para instalar o Python e o gerenciador de pacotes 'pip' em distribuições Linux, abra o terminal e execute:

```
sudo apt update  
sudo apt install python3 python3-pip -y
```

1.2 Verificação da instalação

Após a instalação, verifique se o Python foi instalado corretamente com:

```
python3 --version
```

É esperado que você veja algo como Python 3.x.x.

2 Execução do Programa

2.1 Clonando o repositório

No terminal, execute o seguinte comando para clonar o repositório:

```
git clone https://github.com/ithanamaral/Trabalho_Pratico_FOAC.git
```

2.2 Adicionando seu arquivo .asm

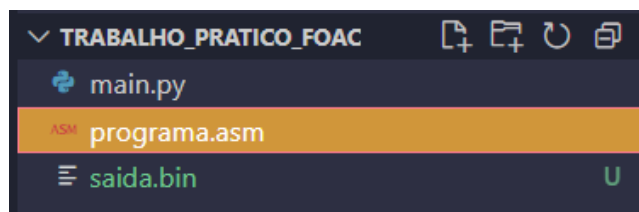
Após a clonagem:

1. Navegue até a pasta clonada:
`cd Trabalho_Pratico_FOAC`
2. Adicione dentro dessa pasta o seu arquivo '.asm' com o código Assembly que deseja montar.

Nome	Data de modificação	Tipo	Tamanho
.git	19/06/2025 16:35	Pasta de arquivos	
main.py	19/06/2025 16:35	Python.File	11 KB
programa.asm	19/06/2025 16:35	Arquivo ASM	1 KB
saida.bin	10/06/2025 19:06	Arquivo BIN	1 KB

O repositório do projeto já tem um arquivo default para facilitar os testes e a execução pelo professor.

OBS: Importante que o arquivo '.asm' deve estar na mesma pasta do script 'main.py'.



Não mova o arquivo para subpastas — o programa não conseguirá localizá-lo corretamente.

3 Executando o montador

Abra a pasta clonada com sua IDE preferida (como VS Code, PyCharm, etc.) Ou execute diretamente pelo terminal com:

```
python3 main.py
```

MONTADOR ASSEMBLY

Digite o nome do arquivo sem extensão: programa

Conteúdo do arquivo:

```
lb x1, 0(x0)           # x1 = MEM[0] (suponha que seja 5, por exemplo)
sub x2, x1, x1          # x2 = x1 - x1 = 0
and x3, x1, x1          # x3 = x1 & x1 = x1 (ainda 5)
ori x4, x3, 15          # x4 = x3 | 0x0F => 5 | 15 = 15
srl x5, x4, x2           # x5 = x4 >> 0 => 15 >> 0 = 15
beq x5, x2, igual       # 15 == 0? Não, então não salta
sb x5, 0(x0)            # MEM[0] = 15
```

igual:

não faz nada

Você quer:

- 1 - Arquivo em Binário
- 2 - Binário no terminal
- 3 - Ambos

Digite sua decisão: 2

```
00000000000000000000000000000001
01000000000100001000000100110011
00000000000100001111000110110011
00000000111100011110001000010011
00000000001000100101001010110011
00000000001000101010000010110011
0000000001010000000000000100011
```

Binário acima

Siga as instruções do terminal, informe o nome do arquivo `‘.asm’` (sem a extensão), e o programa gerará o arquivo `‘saida.bin’` com a versão binária do seu código Assembly, ou apenas irá demonstrá-lo no próprio terminal.