

Arquitetura e Organização de Computadores I

João Marcelo Uchôa de Alencar

13 de março de 2023

Introdução

Objetivo

Importância de Aprender *Assembly*

Metodologia

Material Didático

Ambiente de Programação

Avaliação

Conclusão

Objetivo

Aprender a programar na linguagem de montagem de processadores Intel.

- ▶ Vamos usar *assembly* real para entender o funcionamento dos processadores da arquitetura x86.
- ▶ Não vamos nos aprofundar em todos os detalhes.
- ▶ Sempre que possível, vamos fornecer código em C para comparação.

Importância de Aprender *Assembly*

Apesar da demanda de programadores para linguagem de montagem ter diminuído, compreender *assembly* ainda é importante:

- ▶ Apenas ler sobre *assembly* não é suficiente. É preciso **programar** para entender de fato.
- ▶ A maioria das linguagens de alto nível permite acesso a níveis mais próximos ao *hardware*, mas em alguns casos específicos *assembly* é mais **eficiente**.
- ▶ É uma didática de aprender sobre **organização** e arquitetura de computadores.
- ▶ Ajuda a contextualizar a implementação de linguagens de alto nível em **compiladores e interpretadores**.

Entretanto, se o estudante tem dificuldade em programação, é uma abordagem complicada.

Importância de Aprender *Assembly*

Em 2022.2, ministrei Arquitetura de Computadores:

- ▶ Para tentar inovar, resolvi seguir o livro de organização do Patterson.
- ▶ O foco desse livro é apresentar os conceitos de arquitetura usando MIPS.

Opinião de um aluno na Avaliação Institucional:

"Parar de querer implementar o modelo de ensinar tudo em volta da linguagem assembly. Pois além de ser um conteúdo insignificante, não cobre outros assuntos que poderiam ser abordados na disciplina de Arquitetura de Computadores."

O que vocês acham?

Importância de Aprender *Assembly*

- ▶ Assim como as linguagens naturais, quando você aprende *assembly* algo do *pensamento* da máquina é capturado.
- ▶ Agora é necessário um esforço individual de concentração e dedicação.
- ▶ Se o aluno não deseja aprender, considera algo insignificante, não há didática e professor que produza resultado.
- ▶ Enquanto há alternativas para a cadeira inicial de Arquitetura, nesta aqui o foco é mesmo x86 através da linguagem montagem.

Metodologia

- ▶ Em todas minhas disciplinas, procuro adotar um livro texto principal e tentar seguir o máximo possível do conteúdo.
- ▶ Vantagens:
 - ▶ Um esquema geral definido desde o começo da disciplina.
 - ▶ Metodologia testada, muitas vezes no mundo inteiro, com várias turmas.
 - ▶ Exercícios com resolução possível comprovada, sem delírios da cabeça de professor.
 - ▶ Se o aluno faltar uma aula, já sabe onde procurar o conteúdo.
 - ▶ Também tento trazer conteúdo da minha própria experiência.
- ▶ Desvantagens:
 - ▶ É difícil acompanhar se você não consegue ler um livro.
 - ▶ O que é perfeitamente normal, livros não são a única forma de conhecimento.
 - ▶ Infelizmente, seja qual for o curso ou área, não é razoável fazer um curso superior sem leitura.

Material Didático

- ▶ Existe uma grande quantidade de material, entretanto não encontrei texto em português que ao mesmo tempo fosse completo e acessível.
- ▶ Nas outras versões da disciplina, o professor usou um livro texto em inglês.
- ▶ Então vamos fazer o esforço de também aprimorar nosso inglês durante a disciplina.

Eu sei que é complicado, mas o inglês é essencial no mercado de TI. É muito melhor e mais estimulante você tentar aprender lendo textos de informática e tecnologia do que fazer cursos para aprender a perguntar onde é o banheiro e qual a cor do céu.

Material Didático

Em geral classifico os livros texto que pesquisei em duas categorias:

1. Ensino na graduação:

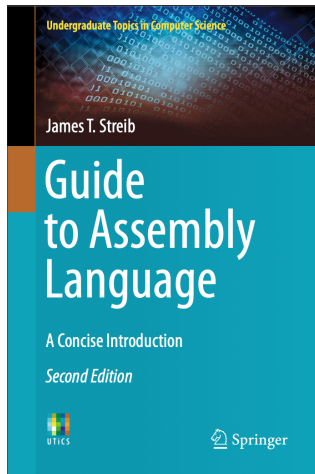
- ▶ Público alvo de iniciantes.
- ▶ Tratam de um subconjunto das instruções x86.
- ▶ Atualização frequente do conteúdo.
- ▶ Exercícios próprios.
- ▶ Ambiente Windows com Visual Studio.

2. Formação Profissional:

- ▶ Público alvo de profissionais.
- ▶ Tratam de aspectos mais complexos do conjunto de instruções.
- ▶ Alguns estão defasados.
- ▶ Poucos exercícios.
- ▶ Ambiente Linux na linha de comando.

Material Didático

- ▶ O ideal seria a opção 1, mas há um problema.
- ▶ O Microsoft Visual Studio Community Edition é gratuito.
 - ▶ Executa no Windows, sistema operacional que nem todos estão com acesso.
 - ▶ Exige o *download* de alguns gigabytes de arquivos.
 - ▶ Precisa de no mínimo 4 gigabytes de memória RAM para executar.
- ▶ Mas pela simplicidade, vamos começar por essa opção.
- ▶ Irei providenciar uma alternativa à instalação do Microsoft Visual Studio de forma nativa.
 - ▶ Vamos usar uma plataforma de nuvem.
 - ▶ Adiciona um pouco de complexidade, mas é mais um conhecimento importante para vocês.



- ▶ Como programar em *assembly* em um processador Intel.
- ▶ Usa o Microsoft Assembler (MASM).
- ▶ Tem como meta usar o mínimo tempo possível de estudo:
 - ▶ Simplificar o uso de registradores.
 - ▶ Usar E/S de forma semelhante à linguagem C.
 - ▶ Simular o uso de estrutura de controles de alto nível.
- ▶ O único pré-requisito é o aluno ter feito uma cadeira de programação. Até os conceitos de arquitetura ele trata de revisar.

Ambiente de Programação

- ▶ O Microsoft Visual Studio Community Edition é gratuito.
- ▶ Exige uma máquina Windows com no mínimo 4 GB de memória RAM.
- ▶ Trata-se de um *download* de 9 GB.

Pessoalmente acho um bom ambiente de programação, que vale a pena aprender a usar. Mas sei que os requisitos acima são complicados.

Ambiente de Programação

- ▶ Vamos utilizar uma máquina virtual na nuvem para programar.
- ▶ Não é o melhor dos cenários, mas é uma alternativa interessante.
- ▶ Vou criar um ambiente com as ferramentas já instaladas e disponibilizar para a turma.
- ▶ Também estou fazendo um material para caso alguém queira se aprofundar, mas não é obrigatório.

Acredito que esta alternativa irá funcionar, mas é preciso *feedback* dos alunos. Vocês precisam entrar em contato comigo se tiverem algum problema na nuvem. Entretanto, qualquer tentativa de ajuda nas últimas horas antes das entregas será ignorada.

Quem deixar para fazer em cima da hora está assumindo a responsabilidade por não ter configurado o ambiente antes.

Avaliação

- ▶ Três notas, média final é a média das três notas.
- ▶ Trabalhos a serem entregues em plataforma digital (a definir):
 - ▶ Questões discursivas.
 - ▶ Trabalhos de implementação.
 - ▶ Sem mistério, são questões de exercício do livro.
 - ▶ Só irei acrescentar instruções do formato da pastas nas questões de código fonte, para que facilite a minha correção, pois faço questão de compilar e executar o código.
- ▶ Irei divulgar o prazo limite para entrega de cada nota, mas só corrijo totalmente depois do prazo. Infelizmente, preciso evitar o caso que corrijo de um aluno e outro entrega o trabalho corrigido como se tivesse feito.

Avaliação

- ▶ Para tentar diminuir a cópia, irei experimentar receber trabalhos de **duplas**, no máximo.
- ▶ Não é permitido um dupla copiar o trabalho de outra dupla. Irei zerar a nota dos quatros envolvidos caso isso aconteça.
- ▶ Se uma parte da dupla desistir, a outra irá continuar só, ou no máximo fazer com outra pessoa cujo o par também desistiu.
- ▶ Vocês tem até a entrega da primeira nota para definir a dupla.

Conclusão

Dúvidas?