

Capacitação Mainframe

Formação de Programadores Assembly para Mainframe

Professor: Carlos Augusto Machado (Cazé)

Carga Horária: 60 horas

Modalidade: Online ao Vivo

Dias e Horário: 2ª e 4ª das 19 às 22h30

Sobre o ASSEMBLY

A linguagem Assembly é uma linguagem de programação de baixo nível projetada para um tipo de processador específico. Podemos produzir o Assembler compilando o código-fonte a partir de uma linguagem de programação de alto nível (COBOL). Também podemos escrever programas já na linguagem Assembly e por sua vez converter o código do Assembly em código de máquina usando um Assembler. (Assembler é o montador).

Autor: Wim Hoogenraad Editor: ITpedia

Objetivo

Capacitar os alunos a desenvolverem rotinas em Assembly.

Conhecimentos Adquiridos

Ao final do curso o aluno estará apto a desenvolver rotinas em Assembly para efetuar cálculos, ler e atualizar arquivos e criar Macros.

Programa do Curso

1. A linguagem ASSEMBLY (Assembler)
2. Sistemas de Numeração
 - 2.1. Decimal
 - 2.2. Binário
 - 2.3. Hexadecimal
3. Tabela EBCDIC
4. Endereços de Memória
5. Arquitetura do Mainframe IBM
 - 5.1. Visão Geral
 - 5.2. Registradores
 - 5.3. PSW
 - 5.4. Endereçamento
 - 5.5. Base + Deslocamento
 - 5.6. Indexador + Base + Deslocamento
 - 5.7. Representação de Dados
 - 5.8. Representação de Instruções
6. Montador ASSEMBLER e Especificação de Elementos
 - 6.1. O Montador Assembler
 - 6.2. O Statement Assembler
 - 6.3. Símbolos
 - 6.4. Código de Instrução
 - 6.5. Instruções de Máquina
 - 6.6. Macro Instruções

- 6.7. Operandos e Comentários
- 6.8. Operandos e Registrador
- 6.9. Operandos e Memória
- 6.10. Termos e Expressões
- 6.11. Expressões para referência de Operandos
- 6.12. Continuação de Statements
- 7. LINKAGE CONVENTION
- 8. Instruções
 - 8.1. Constantes e Storage
 - 8.2. Literais
 - 8.3. Instruções ao Montador
 - 8.4. Formato das Instruções
 - 8.5. Dicas
 - 8.6. Instruções
 - 8.6.1.1. LOAD ADDRESS
 - 8.6.1.2. LOAD
 - 8.6.1.3. LOAD REGISTER
 - 8.6.1.4. LOAD HALFWORD
 - 8.6.1.5. LOAD MULTIPLE
 - 8.6.1.6. STORE
 - 8.6.1.7. STORE HALFWORD
 - 8.6.1.8. ADD PACKED
 - 8.6.1.9. ADD FULLWORD
 - 8.6.1.10. ADD HALFWORD
 - 8.6.1.11. ADD REGISTER
 - 8.6.1.12. SUBTRACT PACKED
 - 8.6.1.13. SUBTRACT FULLWORD
 - 8.6.1.14. SUBTRACT HALFWORD
 - 8.6.1.15. SUBTRACT REGISTER
 - 8.6.1.16. MULTIPLY PACKED
 - 8.6.1.17. MULTIPLY FULLWORD
 - 8.6.1.18. MULTIPLY HALFWORD
 - 8.6.1.19. MULTIPLY REGISTER
 - 8.6.1.20. DIVIDE
 - 8.6.1.21. DIVIDE REGISTER
 - 8.6.1.22. CC (CONDITION CODE)
 - 8.6.1.23. BRANCH ON CONDITION
 - 8.6.1.24. BRANCH ON CONDITION
 - REGISTER 8.6.1.25. BRANCH AND LINK
 - 8.6.1.26. BRANCH AND SAVE
 - 8.6.1.27. BRANCH AND LINK REGISTER
 - 8.6.1.28. BRANCH AND SAVE
 - REGISTER 8.6.1.29. MNEMÔNICOS
 - 8.6.1.30. Instruções Lógicas
 - 8.6.1.31. ADD LOGICAL FULLWORD
 - 8.6.1.32. ADD LOGICAL REGISTER
 - 8.6.1.33. SUBTRACT LOGICAL FULLWORD
 - 8.6.1.34. SUBTRACT LOGICAL REGISTER

8.6.1.35. Instruções de Comparação
8.6.1.36. COMPARE LOGICAL FULLWORD
8.6.1.37. COMPARE LOGICAL REGISTER
8.6.1.38. COMPARE LOGICAL CHARACTER
8.6.1.39. COMPARE LOGICAL IMMEDIATE
8.6.1.40. COMPARE LOGICAL CHARACTER UNDER
MASK 8.6.1.41. INSERT CHARACTER
8.6.1.42. INSERT CHARACTER UNDER MASK
8.6.1.43. STORE CHARACTER
8.6.1.44. STORE CHARACTER UNDER MASK
8.6.1.45. MOVE CHARACTER
8.6.1.46. Álgebra Booleana
8.6.1.47. TEST UNDER MASK
8.6.1.48. Conversão de Formatos
8.6.1.49. PACK
8.6.1.50. UNPK
8.6.1.51. CONVERT TO DECIMAL
8.6.1.52. CONVERT TO BINARY
8.6.1.53. ZERO AND ADD PACKED
8.6.1.54. COMPARE PACKED
8.6.1.55. EDIT
8.6.1.56. BRUNCH ON COUNT
9. Macro Instruções
9.1. O que são Macro Instruções
9.2. SAVE
9.3. RETURN
9.4. GETMAIN
9.5. FREEMAIN
9.6. STORAGE
9.7. WTO
9.8. SNAP
9.9. ABEND
9.10. DATA CONTROL BLOCK
9.11. OPEN
9.12. CLOSE
9.13. GET
9.14. PUT
9.15. CALL
9.16. LINK
9.17. LOAD
9.18. ATTACH
9.19. Passagem de Parâmetro
10. Exemplos de Programas

Público-alvo

Profissionais da área de Mainframe.

a

Pré-Requisitos

Conhecimento de COBOL Básico.

Inglês Básico.

Hardware para Sistema Operacional Windows.

Metodologia de Ensino-Aprendizagem e Avaliativa

A construção da aprendizagem inclui encontros semanais para as aulas, acesso online ao ambiente mainframe, além do acesso ao Ambiente Virtual de Aprendizagem do INEFÉ (AVA). As aulas são transmitidas em tempo real ao vivo, ficando as mesmas gravadas e disponibilizadas na plataforma AVA em até 48hs após a aula, tendo o aluno acesso aos vídeos durante todo o período de aulas e mais 30 (trinta) dias corridos após o encerramento do curso.

O conteúdo a ser desenvolvido organizar-se-á por aulas teóricas e práticas, atividades específicas para fortalecer o melhor aprendizado do conteúdo ministrado.

As avaliações serão compostas por atividades e provas conforme estabelecido pelo professor responsável. A Média Final mínima é de 5,0 para obtenção do Certificado de Conclusão do Curso com Aproveitamento.

Conheça um pouco do Professor Carlos Augusto Machado (Cazé)



Mainframe IT Specialist

* Profissional aposentado do Banco do Brasil-DITEC, onde atuou por mais de 20 anos, sendo altamente reconhecido no mercado da alta plataforma com mais de 38 anos de experiência no Mainframe.

* Larga vivência e competência comprovada em COBOL, CICS, VSAM, ASSEMBLY e outras.

* Atualmente continua atuando junto ao BB-DITEC pela empresa CAST Informática; * Fez parte da equipe de trabalho em CICS e COBOL do Serpro e outras grandes instituições; * Bacharel em Análise e Desenvolvimento de Sistemas;

* Professor do Curso de Pós-Graduação em Alta Plataforma Mainframe IBM e de Cursos de Capacitação e Treinamentos em diversas áreas do Mainframe IBM..

* Sua paixão pela tecnologia e busca constante pela excelência o destacam como uma figura respeitada e admirada no campo da tecnologia da informação.

.....