



Plano de Atividades Domiciliares ADE

Unidade Curricular: Laboratório de Sistemas Computacionais: Compiladores

Professor(es):
Luiz Eduardo Galvão Martins

Contato: legmartins@unifesp.br

Ano Letivo: 2021

Semestre: 1º

Carga horária total: 58(P) + 14(T) = 72h

Turmas: Integral

Plataforma de acesso ao curso: Moodle UNIFESP.

Objetivos (remoto): Esta unidade curricular faz parte das unidades curriculares integradas definidas no Projeto Pedagógico do Curso, as quais são utilizadas para que o aluno possa, de fato, desenvolver um sistema computacional completo durante o seu processo de aprendizagem, envolvendo a integração entre *hardware* e *software*. O sistema completo compreende o desenvolvimento da arquitetura do processador, a definição de uma linguagem de programação, o projeto de um compilador, a definição de um sistema operacional e um processo de comunicação em rede entre dois ou mais sistemas. Dentro deste contexto, ao término desta unidade curricular, o aluno deverá ter implementado um compilador completo para o sistema computacional especificado. O objetivo geral dessa disciplina é capacitar o aluno a construir um compilador completo, envolvendo o processo de análise e síntese do compilador.

Conteúdo Programático e Cronograma (especificar planejamento de atividades síncronas e assíncronas e CH equivalente, conforme estratégia didática/pedagógica em ensino remoto escolhida pelo docente ou grupo de docentes. Importante explicitar se o conteúdo será síncrono e assíncrono em cada ação):

- Semana 1

Atividades síncronas (1 hora): Apresentação da disciplina. Revisão sobre implementação do código intermediário.

Atividades assíncronas (3 horas): Revisão dos módulos de análise léxica, sintática e semântica apresentados na disciplina Compiladores. Modelagem dos módulos de análise do Compilador: diagramas de atividades e de blocos (SysML).



- Semana 2

Atividades síncronas (1 hora): Orientações sobre o desenvolvimento do projeto. Acompanhamento da modelagem e implementação do compilador. Discussão sobre dúvidas.

Atividades Assíncronas (3 horas): Revisão da arquitetura e conjunto de instruções do processador implementado (FPGA). Estudo e análise de projetos de semestres anteriores (disponibilizados no Moodle).

- Semana 3

Atividades síncronas (1 hora): Orientações sobre o desenvolvimento do projeto. Acompanhamento da modelagem e implementação do compilador. Discussão sobre dúvidas.

Atividades Assíncronas (3 horas): Definição dos tipos de quádruplas do código intermediário. Modelagem do módulo de geração do código intermediário: diagrama de atividades e de blocos

- Semana 4

Atividades síncronas (1 hora): Orientações sobre o desenvolvimento do projeto. Acompanhamento da modelagem e implementação do compilador. Discussão sobre dúvidas.

Atividades Assíncronas (3 horas): Implementação do módulo de geração de código intermediário.

- Semana 5

Atividades síncronas (1 hora): Orientações sobre o desenvolvimento do projeto. Acompanhamento da modelagem e implementação do compilador. Discussão sobre dúvidas.

Atividades Assíncronas (3 horas): Implementação do módulo de geração de código intermediário.

- Semana 6

Atividades síncronas (1 hora): Orientações sobre o desenvolvimento do projeto. Acompanhamento da modelagem e implementação do compilador. Discussão sobre dúvidas.

Atividades Assíncronas (3 horas): Implementação do módulo de geração de código intermediário.

- Semana 7

Atividades síncronas (1 hora): Orientações sobre o desenvolvimento do projeto. Acompanhamento da modelagem e implementação do compilador. Discussão sobre dúvidas.

Atividades Assíncronas (3 horas): Implementação e testes do módulo de geração de código intermediário. Submissão do primeiro entregável.



- Semana 8

Atividades síncronas (1 hora): Orientações sobre o desenvolvimento do projeto. Acompanhamento da modelagem e implementação do compilador. Discussão sobre dúvidas.

Atividades Assíncronas (3 horas): Implementação dos módulos de síntese (gerador de código *assembly* e gerador de código binário).

- Semana 9

Atividades síncronas (1 hora): Orientações sobre o desenvolvimento do projeto. Acompanhamento da modelagem e implementação do compilador. Discussão sobre dúvidas.

Atividades Assíncronas (3 horas): Implementação e testes dos módulos de síntese (gerador de código *assembly* e gerador de código binário).

- Semana 10

Atividades síncronas (1 hora): Orientações sobre o desenvolvimento do projeto. Acompanhamento da modelagem e implementação do compilador. Discussão sobre dúvidas.

Atividades Assíncronas (3 horas): Implementação e testes dos módulos de síntese (gerador de código *assembly* e gerador de código binário).

- Semana 11

Atividades síncronas (1 hora): Orientações sobre o desenvolvimento do projeto. Acompanhamento da modelagem e implementação do compilador. Discussão sobre dúvidas.

Atividades Assíncronas (3 horas): Implementação e testes dos módulos de síntese (gerador de código *assembly* e gerador de código binário).

- Semana 12

Atividades síncronas (1 hora): Orientações sobre o desenvolvimento do projeto. Acompanhamento da modelagem e implementação do compilador. Discussão sobre dúvidas.

Atividades Assíncronas (3 horas): Implementação e testes dos módulos de síntese (gerador de código *assembly* e gerador de código binário).

- Semana 13

Atividades síncronas (1 hora): Orientações sobre o desenvolvimento do projeto. Acompanhamento da modelagem e implementação do compilador. Discussão sobre dúvidas.

Atividades Assíncronas (3 horas): Implementação e testes dos módulos de síntese (gerador de código *assembly* e gerador de código binário).



- Semana 14

Atividades síncronas (1 hora): Orientações sobre o desenvolvimento do projeto. Acompanhamento da modelagem e implementação do compilador. Discussão sobre dúvidas.

Atividades Assíncronas (3 horas): Implementação e testes dos módulos de síntese (gerador de código *assembly* e gerador de código binário).

- Semana 15

Atividades síncronas (1 hora): Orientações sobre o desenvolvimento do projeto. Acompanhamento da modelagem e implementação do compilador. Discussão sobre dúvidas.

Atividades Assíncronas (3 horas): Implementação e testes dos módulos de síntese (gerador de código *assembly* e gerador de código binário). Preparação do relatório final. Submissão no Moodle

- Semana 16

Atividades síncronas (1 hora): Orientações sobre o desenvolvimento do projeto. Acompanhamento da modelagem e implementação do compilador. Discussão sobre dúvidas.

Atividades Assíncronas (3 horas): Implementação e testes dos módulos de síntese (gerador de código *assembly* e gerador de código binário). Preparação do relatório final.

- Semana 17

Atividades síncronas (1 hora): Orientações finais sobre o desenvolvimento do projeto. Acompanhamento da modelagem e implementação do compilador. Discussão sobre dúvidas.

Atividades Assíncronas (3 horas): Implementação e testes dos módulos de síntese (gerador de código *assembly* e gerador de código binário). Preparação do relatório final. Submissão no Moodle. Entrega do projeto final (segundo entregável).

- Semana 18

Atividades síncronas (1 hora): Finalização da disciplina. Orientações sobre o desenvolvimento do projeto. Acompanhamento da modelagem e implementação do compilador. Discussão sobre dúvidas.

Atividades Assíncronas (3 horas): Atividades válidas como exame. Implementação e testes dos módulos de síntese (gerador de código *assembly* e gerador de código binário). Preparação do relatório final. Submissão no Moodle.

Metodologia de Ensino Utilizada:

A metodologia de ensino consistirá em:

- Disponibilização de material de apoio para a realização do projeto de compiladores, por meio da plataforma Moodle;
- Atividades assíncronas para a realização do projeto (modelagem e programação do compilador);



- Encontros remotos semanais para acompanhamento dos projetos e discussões sobre dúvidas.

Metodologia de Avaliação (estratégias para atingir conceitos “cumprido” ou “não cumprido”):

O conceito final será computado com base na avaliação de dois entregáveis ao longo do semestre.

O primeiro entregável será composto por:

- Módulos de análise léxica, sintática e semântica do Compilador (código fonte);
- Módulo de geração de código intermediário do Compilador (código fonte);
- Modelos SysML correspondentes (diagramas de atividades e de blocos);
- Exemplos de geração de código intermediário, a partir de programas C-.
- Vídeo explicativo do entregável.

O segundo entregável será composto por:

- Relatório final (conforme *template* a ser disponibilizado);
- Código fonte completo do Compilador.
- Vídeo explicativo do entregável.

Bibliografia básica e complementar para uso remoto

Básica:

LOUDEN, Kenneth C; SILVA, Flávio S.C. Compiladores: princípios e práticas. São Paulo: Thomson, 2004. 569 p. ISBN 978-85-221-0422-2.

Aho, Alfred V et al. Compiladores: princípios, técnicas e ferramentas. 2 ed. São Paulo: Person Addison Wesley, 2007. 634 p. ISBN 978-85-88639-24-9. tradução de "Compilers: principles, techniques, and tools.

APPEL, Andrew W; PALSBERG, Jens. Modern compiler implementation in Java. 2 ed. New York: Cambridge at the University Press, 2002. 501 p p. ISBN 978-0-521-82060-8.;[10];RICARTE, I. Introdução à Compilação. Editora Elsevier/Campus, 2008.

Complementar:

SCOTT, Michael L. Programming language pragmatics. New York: Morgan Kaufmann, c2009. 910 p. ISBN 978-0-12-374514-9.

HOPCROFT, John E; MOTWANI, Rajeev; ULLMAN, Jeffrey D. Introdução à teoria de autômatos, linguagens e computação. [Introduction to automata theory, languages, and computation.]. Rio de Janeiro: Campus, 2002. 560 p. ISBN 978-85-352-1072-9.

ULLMAN, Jeffrey D; MOTWANI, Rajeev; HOPCROFT, John E. Introduction to automata



Ministério da Educação
Universidade Federal de São Paulo
Instituto de Ciência e Tecnologia



theory, languages, and computation. 3.ed. Boston (USA): Pearson, 2006. 535 p. ISBN 978-0-321-45536-9.

Ricarte, Ivan. Introdução à compilação. Rio de Janeiro: Elsevier, 2008. 264 p. ISBN 9788535230673.

PRICE, Ana Maria de Alencar; TOSCANI, Simão Sirineo. Implementação de linguagens de programação: compiladores. 3.ed. Porto Alegre: Bookman, 2008. 9. 195 p. ISBN 978-85-7780-348-4.

Como Fazer Apresentações em Eventos Acadêmicos e Empresariais – Linguagem Verbal, Comunicação Corporal e Recursos Audíveis. Maria Helena da Nobrega. Editora Atlas. ISBN: 8522456380, 2010.

Metodologia de Pesquisa para Ciência da Computação. Raul Sidnei Wazlawick. ISBN: 9788535235227, 2009.