Ferramenta educacional para auxiliar no ensino de Fork/Join

Lucas da Silva Marcos[1], **Rodolfo Adamshuk Silva**[1]

[1]Coordenação do Curso de Engenharia de Software
Universidade Tecnológica Federal do Paraná (UTFPR) – Dois Vizinhos, PR – Brazil

lucasmarcos@alunos.utfpr.edu.br, rodolfoa@utfpr.edu.br

Abstract.

Resumo.

1. Introdução

1.1. Problema

Programação concorrente. Ensino de programação concorrente.

Ensino de programação no ensino superior. Unidade curricular de Programação Concorrente e Distribuída.

Modelo fork-join. Anos 60. Algumas implementações.

Decomposição. Grafo de tarefas. Métricas.

Linguagem de alto nível. Compiladores. Analise léxica. Analise sintática.

1.2. Motivação

Conceitos complexos de programação concorrente.

Escassez de material e ferramentas de apoio e suporte a educação.

Dificuldades de aprendizado.

1.3. Objetivo

Construção de uma ferramenta de apoio ao ensino por meio de um compilador para uma linguagem de alto nível que represente o modelo fork-join.

Desenvolver uma ferramenta educacional para auxiliar no ensino do modelo Fork/Join de concorrência no contexto de uma unidade curricular para cursos de Ensino Superior em Engenharia de Software com foco nas atividades desenvolvidas na disciplina de Programação Concorrente Distribuída.

2. Trabalhos relacionados

Fork/Join [Conway 1963] e [Dennis and Van Horn 1966].

Sistemas Distribuídos [Coulouris et al. 2001] e [Van Steen and Tanenbaum 2017].

Compiladores [Louden 2004].

Ferramentas educacionais.

Materia de apoio para aprendizagem de Programação Concorrente.

Compiladores em sala de aula.

Compiladores de Pascal usados para o ensino de matemática.

3. Proposta

O desenvolvimento de um compilador e plataforma com o foco no uso em sala de aula.

Será especificado uma gramática formal da linguagem de alto nível utilizada. A linguagem possuirá os primitivos Fork e Join como modelo de concorrência.

Será construído um analisador léxico, reportando aos alunos erros nos programas.

Será construído um analisador sintático, que modelara a estrutura do programa e reportará erros de construção.

Como saída esperada teremos, além das mensagens de erro ou de sucesso, um grafo de tarefas e dependências, além de métricas do programa especificado, como o caminho crítico e o grau máximo de concorrência.

Também será construído uma plataforma web, como um editor de textos integrado e renderização do grafo e métricas. Exemplo Compiler Explorer https://godbolt.org.

4. Metodologia

Pesquisa aplicada e exploratória.

Aplicaremos o protótipo em sala de aula, coletaremos dados e feedback dos alunos. Feedback pesquisa qualitativa.

Cronograma.

5. Resultados esperados

Espera-se que a ferramenta proposta contribua significativamente para o aprimoramento do ensino de programação concorrente, fornecendo um recurso para o aprendizado dos alunos e auxiliando os professores no processo de ensino.

Referências

Conway, M. E. (1963). A multiprocessor system design. In *Proceedings of the November 12-14, 1963, fall joint computer conference*, pages 139–146.

Coulouris, G., Dollimore, J., and Kindberg, T. (2001). *Sistemas distribuidos*, volume 6. Addison Wesley Madrid.

Dennis, J. B. and Van Horn, E. C. (1966). Programming semantics for multiprogrammed computations. *Communications of the ACM*, 9(3):143–155.

Louden, K. (2004). *Compiladores - Princípios e Práticas*. Pioneira Thomson Learning.

Van Steen, M. and Tanenbaum, A. S. (2017). *Distributed systems*. Maarten van Steen Leiden, The Netherlands.