

Ferramenta educacional para auxiliar no ensino do modelo fork-join

Lucas da Silva Marcos¹, Rodolfo Adamshuk Silva¹

¹Universidade Tecnológica Federal do Paraná (UTFPR)
Dois Vizinhos, Paraná

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

Abstract.

Resumo. *A importância da programação concorrente no campo da ciência da computação é inegável. No entanto, a complexidade inerente ao ensino desta disciplina em ambientes acadêmicos é um desafio significativo. Este artigo propõe uma ferramenta educacional destinada a facilitar o ensino do modelo de programação fork-join, um paradigma utilizado na programação concorrente.*

1. Introdução

Rascunho.

- seis parágrafos, dois problema, dois motivação, dois objetivo
- concorrência
- fork-join
- linguagem de alto nível
- ferramenta de ensino
- compilador
- grafo e métricas

O ensino de programação concorrente enfrenta um desafio significativo devido à natureza complexa e abstrata dos conceitos envolvidos. A concorrência, que envolve a execução simultânea de múltiplas tarefas, introduz uma nova camada de complexidade que muitos estudantes acham difícil de compreender. A compreensão dos conceitos fundamentais, como sincronização e comunicação entre processos concorrentes, é essencial para o desenvolvimento de sistemas robustos e escaláveis, mas a falta de experiência prática e ferramentas adequadas pode dificultar o aprendizado eficaz desses conceitos.

O modelo fork-join, amplamente utilizado na programação concorrente, apresenta desafios adicionais para os estudantes, especialmente em relação à decomposição adequada de tarefas e à gestão eficiente de recursos compartilhados. A natureza dinâmica da criação e sincronização de threads pode resultar em problemas de concorrência, como condições de corrida e deadlocks, que são difíceis de identificar e corrigir sem um entendimento profundo do modelo fork-join. Como resultado, muitos alunos enfrentam dificuldades ao desenvolver e depurar programas baseados neste paradigma.

A motivação para o desenvolvimento de uma ferramenta educacional que suporte o ensino do modelo fork-join em uma linguagem de alto nível surge da necessidade de tornar os conceitos de concorrência mais acessíveis e tangíveis para os estudantes. Ao

fornecer uma linguagem de programação abstrata e familiar, enriquecida com operações específicas do modelo fork-join, espera-se reduzir a curva de aprendizado e permitir que os alunos se concentrem nos conceitos fundamentais, em vez de se preocuparem com detalhes de implementação.

O objetivo principal deste trabalho é desenvolver um compilador que permita aos alunos expressar o modelo fork-join em uma linguagem de alto nível de forma eficiente. O compilador fornecerá recursos de análise léxica e sintática para identificar erros no código.

Além disso, a ferramenta educacional incluirá recursos para visualização da decomposição de tarefas em forma de grafo, juntamente com métricas associadas para avaliar o desempenho e a eficiência dos programas desenvolvidos.

1.1. Ideias soltas

A programação concorrente é uma área essencial da ciência da computação, permitindo que programas executem tarefas simultaneamente e otimizem o uso dos recursos do sistema. No entanto, ensinar esse conceito em ambientes acadêmicos pode ser desafiador devido à sua complexidade.

O problema central é como facilitar o ensino eficaz do modelo de programação fork-join, um paradigma amplamente utilizado na programação concorrente.

Uma porcentagem crescente da atividade de computação será realizada por sistemas de computadores multiprogramados. Tais sistemas são caracterizados pela aplicação de recursos de computação (capacidade de processamento, memória principal, armazenamento de arquivos, equipamentos periféricos) a muitas computações separadas, mas operando simultaneamente.

1.2. Problema

O ensino eficaz de programação concorrente enfrenta diversos desafios, incluindo a compreensão dos conceitos subjacentes, a implementação prática de soluções e a avaliação do desempenho dos programas desenvolvidos. Além disso, a natureza abstrata e complexa dos modelos de programação concorrente pode dificultar a visualização e a análise das interações entre as tarefas em execução. Os estudantes muitas vezes têm dificuldade em compreender e aplicar corretamente os conceitos de concorrência, resultando em um aprendizado menos eficaz e em soluções menos otimizadas.

1.3. Motivação

A motivação para o desenvolvimento desta ferramenta educacional surge da necessidade de superar os desafios enfrentados no ensino da programação concorrente. Ao oferecer uma ferramenta que permite expressar o modelo fork-join em uma linguagem de alto nível, acompanhada de recursos de análise léxica e sintática, espera-se proporcionar aos alunos uma experiência de aprendizado mais interativa e eficaz. Além disso, a visualização da decomposição de tarefas em forma de grafo, juntamente com a análise de métricas de desempenho, visa promover uma compreensão mais profunda e intuitiva dos conceitos de concorrência.

A motivação para abordar esse problema reside na necessidade de fornecer aos alunos ferramentas educacionais que simplifiquem o aprendizado da programação concorrente. A complexidade inerente ao modelo fork-join requer abordagens pedagógicas eficazes para garantir que os estudantes compreendam os conceitos e possam aplicá-los de forma prática.

1.4. Objetivo

A ferramenta em questão é um software de apoio pedagógico que permite a expressão do modelo fork-join em uma linguagem de alto nível, enriquecida com as operações FORK e JOIN. A ferramenta será equipada com um analisador léxico e um analisador sintático, ambos projetados para identificar e apontar erros no uso da linguagem pelos alunos, proporcionando assim um feedback imediato e contribuindo para um processo de aprendizagem mais eficiente.

A ferramenta é integrada em um portal web de fácil utilização, tornando-a adequada para aplicação em sala de aula. Este artigo também apresenta uma pesquisa de resultados obtidos em sala de aula, fornecendo evidências empíricas da eficácia da ferramenta no ensino do modelo Fork/Join.

O objetivo deste trabalho é desenvolver uma ferramenta educacional que facilite o ensino do modelo fork-join, tornando-o mais acessível e compreensível para os estudantes de programação concorrente. Especificamente, almeja-se:

Permitir que os alunos expressem o modelo fork-join em uma linguagem de alto nível, enriquecida com as operações FORK e JOIN, de forma intuitiva e eficiente. Oferecer recursos de análise léxica e sintática para identificar e corrigir erros no código fornecido pelos alunos, proporcionando feedback imediato durante o processo de aprendizagem. Visualizar a decomposição de tarefas em forma de grafo, auxiliando os alunos na compreensão das interações entre as tarefas concorrentes e suas dependências. Facilitar a análise de métricas de desempenho associadas ao modelo fork-join, ajudando os alunos a avaliar a eficácia e a eficiência de suas soluções. Integrar a ferramenta em um portal web de fácil utilização, tornando-a acessível para aplicação em sala de aula e promovendo a interação entre alunos e instrutores no processo de ensino-aprendizagem.

O objetivo deste trabalho é desenvolver uma ferramenta educacional específica para o modelo fork-join. Essa ferramenta permitirá que os alunos expressem o modelo em uma linguagem de alto nível, enriquecida com as operações FORK e JOIN. Além disso, a ferramenta incluirá analisadores léxicos e sintáticos para identificar erros no uso da linguagem pelos alunos, proporcionando feedback imediato. Além disso, a ferramenta visualizará a decomposição de tarefas em um grafo de tarefas, auxiliando os alunos a compreenderem as implicações de desempenho e eficiência de seus programas. A integração dessa ferramenta em um portal web acessível tornará seu uso viável em sala de aula, e resultados empíricos comprovam sua eficácia no ensino do modelo Fork/Join.

1.5. Fragmentos

- Ensino de programação concorrente.
- Ensino de programação no ensino superior.
- Unidade curricular de Programação Concorrente e Distribuída.
- Algumas implementações.

- Decomposição. Grafo de tarefas. Métricas.
- Linguagem de alto nível.
- Conceitos complexos de programação concorrente.
- Escassez de material e ferramentas de apoio e suporte a educação.
- Dificuldades de aprendizado.

A programação concorrente é de extrema importância.

O ensino dela em universidade pode ser uma dificuldade.

Software de apoio ajudam na aprendizagem.

O modelo fork-join pode ser expresso em um linguagem de alto nível incrementada com as operações FORK e JOIN.

Uma analisador léxico e um analisador sintático apontam erros dos alunos no uso dessa linguagem.

Decomposição de tarefas em grafo de tarefas e métricas associadas.

Uma plataforma web de fácil aplicação em sala de aula.

Uma pesquisa de resultados em sala de aula.

1.6. Problema

Programação concorrente ou programação simultânea é um paradigma de programação para a construção de programas de computador que fazem uso da execução simultânea de várias tarefas computacionais interativas, que podem ser implementadas como programas separados ou como um conjunto de threads criadas por um único programa. Essas tarefas podem ser executadas por um único processador, vários processadores em um único equipamento ou processadores distribuídos por uma rede. Programação concorrente é relacionada com programação paralela, mas foca mais na interação entre as tarefas. A interação e a comunicação correta entre as diferentes tarefas, além da coordenação do acesso simultâneo aos recursos computacionais são as principais questões discutidas durante o desenvolvimento de sistemas simultâneos.

Vantagens do paradigma incluem o aumento de desempenho, pois aumenta-se a quantidade de tarefas sendo executadas em determinado período de tempo, e a possibilidade de uma melhor modelagem de programas, pois determinados problemas computacionais são simultâneos por natureza.

Na computação paralela, o modelo fork-join (bifurcar-juntar) é uma forma de configurar e executar programas em paralelo de tal forma que a execução das tarefas concorrentes "juntam-se"(join) em um ponto subsequente, onde a execução passa a ser sequencial. As seções paralelas podem se bifurcar recursivamente até que uma determinada granularidade da tarefa seja atingida. Fork-join pode ser considerado como um padrão de projeto paralelo. Foi formulado já em 1963.

Na ciência da computação, análise léxica, lexing ou tokenização é o processo de converter uma sequência de caracteres (como em um programa de computador ou página da web) em uma sequência de tokens (strings com um significado atribuído e, portanto, identificado). Um programa que realiza análise lexical pode ser denominado lexer, tokenizer, ou scanner, embora scanner também seja um termo para o primeiro estágio de

um lexer. Um lexer geralmente é combinado com um analisador, que juntos analisam a sintaxe de linguagens de programação, páginas da Web e assim por diante.

A análise sintática é um processo de um compilador de uma linguagem de programação, é a segunda fase da compilação onde se analisa uma sequência que foi dada entrada para verificar sua estrutura gramatical segundo uma determinada gramática formal. Este processo trabalha em conjunto com a análise lexical, a primeira etapa, onde se verifica de acordo com determinado alfabeto, e análise semântica, a terceira etapa, onde verificam-se os erros semânticos.

A análise sintática transforma um texto na entrada em uma estrutura de dados, em geral uma árvore, o que é conveniente para processamento posterior e captura a hierarquia implícita desta entrada. Através da análise lexical é obtido um grupo de tokens, para que o analisador sintático use um conjunto de regras para construir uma árvore sintática da estrutura. A vasta maioria dos analisadores sintáticos implementados em compiladores aceitam alguma linguagem livre de contexto para fazer a análise. Estes analisadores podem ser de vários tipos, como o LL, LR e SLR.

1.7. Motivação

A semântica é definida para várias meta-instruções que realizam operações essenciais para a escrita de programas em sistemas de computadores multiprogramados. Essas meta-instruções estão relacionadas ao processamento paralelo, proteção de cálculos separados, depuração de programas e o compartilhamento entre usuários de segmentos de memória e outros objetos de computação, cujos nomes são estruturados hierarquicamente. A sofisticação da linguagem contemplada está no meio do caminho entre uma linguagem de montagem e uma linguagem algébrica avançada.

Sabe-se que o modelo Fork/Join foi criado nos anos 60, sendo esse a base de quase todos os modelos de concorrência usados atualmente. Todavia, ainda é um conteúdo que apresenta pouca referência na literatura científica, uma vez que é usado de modo muito limitado, apresentando-se quase que exclusivamente apenas nas disciplinas de Programação Concorrente Distribuída.

Dessa forma, objetivo desta pesquisa é 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.

Considerando o fato de que o assunto é pouco descrito em pesquisas científicas, como já citado, esta pesquisa traz como diferencial o objetivo de desenvolver uma ferramenta educacional para auxiliar professores e alunos durante o processo de ensino-aprendizagem do modelo Fork/Join ao longo das aulas de Programação Concorrente Distribuída dos cursos de Engenharia de Software.

Assim, a necessidade dessa ferramenta evidencia-se ao percebermos que os conceitos de programação concorrente têm como principal limitação a pouca variedade de ferramentas de suporte. Fato esse que dificulta no uso do modelo Fork/Join como também desenvolver outras atividades relacionadas à área.

Nesse sentido, pretende-se desenvolver a ferramenta por meio de um compilador para uma linguagem de alto nível que represente o modelo Fork/Join e que auxilie na

correção de sintaxe, criação do grafo de paralelismo e exiba alguns tipos de métricas (das métricas, essas só serão possíveis termos conhecimento de quais tipos serão alcançados após o desenvolvimento da ferramenta).

Portanto, o estudo para o desenvolvimento desta pesquisa e, também, dessa possível ferramenta serão baseados em artigos sobre Fork/Join e compiladores para desenvolver um artigo apresentando um compilador de apoio ao ensino.

1.8. 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.

O objetivo desta pesquisa é 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 e Distribuída.

Considerando o fato de que o assunto é pouco descrito em pesquisas científicas, como já citado, esta pesquisa traz como diferencial o objetivo de desenvolver uma ferramenta educacional para auxiliar professores e alunos durante o processo de ensino-aprendizagem do modelo Fork/Join ao longo das aulas de Programação Concorrente e Distribuída dos cursos de Engenharia de Software.

Assim, a necessidade dessa ferramenta evidencia-se ao percebermos que os conceitos de programação concorrente têm como principal limitação a pouca variedade de ferramentas de suporte.

Fato esse que dificulta no uso do modelo Fork/Join como também desenvolver outras atividades relacionadas à área.

Nesse sentido, pretende-se desenvolver a ferramenta por meio de um compilador para uma linguagem de alto nível que represente o modelo Fork/Join e que auxilie na correção de sintaxe, criação do grafo de paralelismo e exiba alguns tipos de métricas (das métricas, essas só serão possíveis termos conhecimento de quais tipos serão alcançados após o desenvolvimento da ferramenta).

Portanto, o estudo para o desenvolvimento desta pesquisa e, também, dessa possível ferramenta serão baseados em artigos sobre Fork/Join e compiladores para desenvolver um artigo apresentando um compilador de apoio ao ensino.

2. Trabalhos relacionados

Artigos sobre o modelo fork-join [Conway 1963] e [Dennis and Van Horn 1966].

Livro sobre sistemas distribuídos [Coulouris et al. 2001] e [Van Steen and Tanenbaum 2017].

Livro sobre compiladores [Louden 2004].

Ferramentas educacionais.

Material 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. Um exemplo de plataforma sendo o Compiler Explorer <https://godbolt.org>.

4. Metodologia

Será um pesquisa aplicada e exploratória.

Aplicaremos o protótipo em sala de aula, coletaremos dados e feedback dos alunos. O feedback será uma pesquisa qualitativa com as opiniões dos alunos sobre o uso do sistema em um formulário online.

O cronograma das atividades proposta está descrito abaixo.

4.1. Atividades

- **Atividade 1:** atividade 1;
- **Atividade 2:** atividade 2.

	Junho	Julho	Agosto	Setembro	Outubro	Novembro	Dezembro
Atividade 1							
Atividade 2							

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.