

UNIVERSIDADE FEDERAL DE SANTA CATARINA

CAMPUS JOINVILLE

PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA DE SISTEMAS ELETRÔNICOS

Jonas Oscar Foyth

Kelvin Cesar de Andrade

Joinville

2023

LISTA DE FIGURAS

[Figura 1 - Modelo Proposto 17](#_Toc144758634)

[Figura 2 - Menu para criação da Rede de Petri 17](#_Toc144758635)

[Figura 3 - Modelo desenvolvido através do Software TAPAAL 18](#_Toc144758636)

[Figura 4 - Modo de simulação 19](#_Toc144758637)

[Figura 5 - Verificações realizadas 19](#_Toc144758638)

SUMÁRIO

[1 INTRODUÇÃO 16](#_Toc144757977)

[1.1 SOBRE O TRABALHO Proposto 16](#_Toc144757978)

[2 DESENVOLVIMENTO 17](#_Toc144757979)

[2.1 Simulação 18](#_Toc144757980)

[2.2 Verificação formal 19](#_Toc144757981)

[3 CONCLUSÃO 20](#_Toc144757982)

[REFERÊNCIAS 21](#_Toc144757983)

# INTRODUÇÃO

A verificação formal de testes é uma abordagem fundamental na engenharia de software que se concentra em garantir a correção e a confiabilidade de sistemas de software complexos. Ela difere dos métodos tradicionais de teste, que frequentemente envolvem a execução de casos de teste em um sistema e a verificação do seu comportamento observado. Em vez disso, a verificação formal busca a validação matemática [1] da correção de um sistema de software por meio da análise rigorosa de sua especificação e da modelagem precisa de seu comportamento.

Uma ferramenta importante no domínio da verificação formal de testes é o software TAPAAL (*Teaching, Analysis, and Performance Analysis of Algorithms*) [2]. TAPAAL é uma ferramenta de código aberto que se concentra na análise de sistemas concorrentes e distribuídos, como sistemas embarcados, protocolos de comunicação e sistemas de tempo real. Desenvolvida principalmente para fins acadêmicos, a ferramenta TAPAAL oferece uma plataforma flexível para modelar sistemas usando redes de Petri temporizadas coloridas.

As redes de Petri temporizadas coloridas são uma representação poderosa para modelar sistemas concorrentes e distribuídos que envolvem restrições de tempo. Com o TAPAAL, os engenheiros de software podem criar modelos precisos desses sistemas, definir propriedades de verificação, como ausência de deadlocks, e realizar análises automatizadas para verificar a correção do sistema em relação a essas propriedades.

Em resumo, a verificação formal de testes, apoiada pelo software TAPAAL, desempenha um papel crucial na garantia da qualidade e na correção de sistemas de software complexos, ajudando a evitar problemas críticos, como falhas de segurança, falhas de desempenho e comportamento inesperado. Ela oferece uma abordagem matematicamente sólida para garantir que os sistemas de software funcionem conforme o esperado, proporcionando maior confiabilidade e segurança em aplicações críticas.

## SOBRE O TRABALHO Proposto

O trabalho proposto consiste em realizar a verificação formal do modelo ilustrado através da Figura 1 de um sistema operacional, contendo as operações de alocação de memória, gerenciamento de tarefas e disco. Dessa forma, o trabalho especifica a elaboração das propriedades (*queries*) de deadlock e *liveness*.

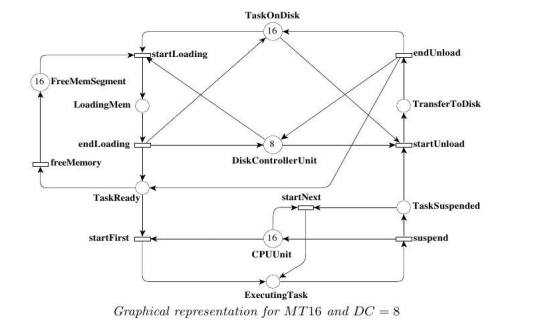


Figura - Modelo Proposto

# DESENVOLVIMENTO

O primeiro passo foi realizar a representação da Rede de Petri através do software TAPAAL. Essa representação, pode ser feita através do menu ilustrado através da Figura 2 no qual contém os campos de transições e estados.



Figura - Menu para criação da Rede de Petri

O resultado final pode ser visto na Figura 3, onde a representação completa do modelo proposto foi elaborada no software, tornando assim o modelo pronto para a realização de simulações.

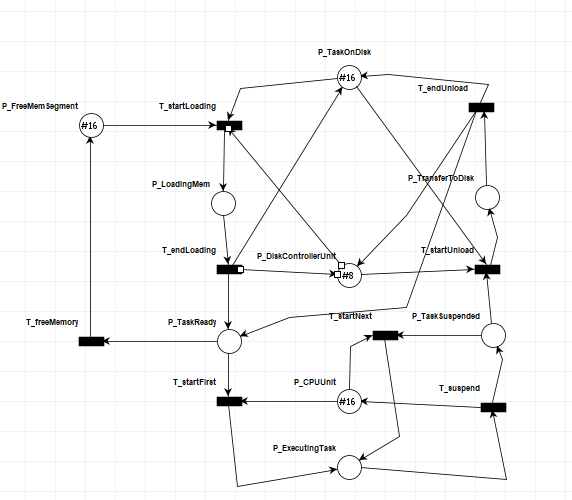


Figura - Modelo desenvolvido através do Software TAPAAL

## Simulação

A simulação pode ser realizada através do campo *Toggle Simulation Mode (**)* no qual mudará a tela para o modo de simulação, ilustrado através da Figura 4. Esse campo permite alterar configurações como velocidade de transição, habilitar/parar simulação, indicação de estado, número de estados percorridos, entre outros.

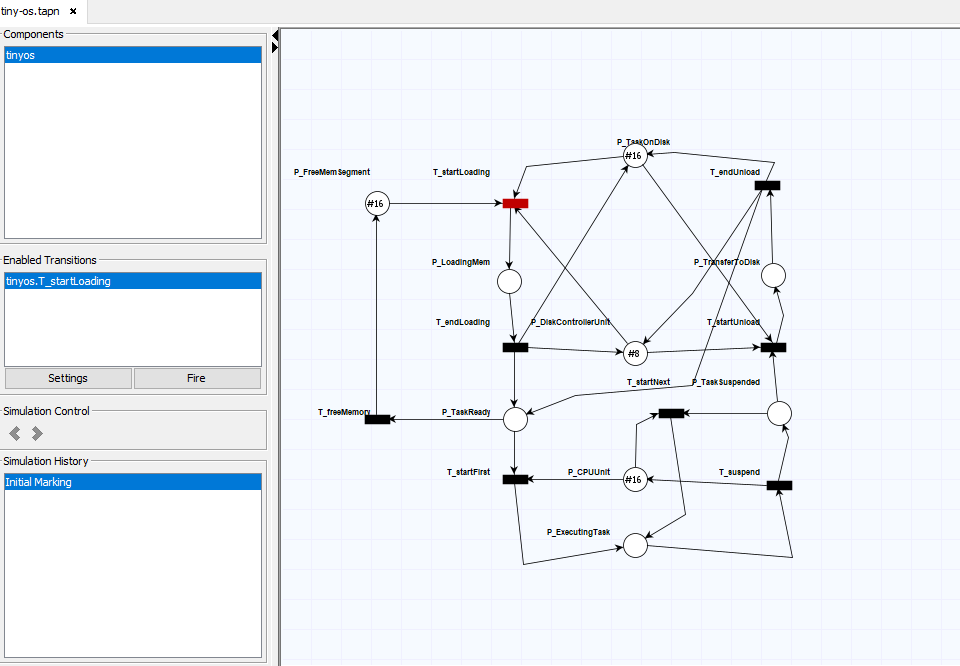


Figura - Modo de simulação

## Verificação formal

A verificação através do software TAPAAL é realizada através da utilização de *Queries* (a descrição bem como a elaboração das queries que foram utilizadas no projeto para verificar as propriedades de deadlock e *liveness* podem ser vistas através do arquivo README contida no repositório do projeto*),* no qual são estruturas formais que são utilizadas para elaborar a rotina de verificação. Sua configuração pode ser feita através do campo destacado através da Figura 5.

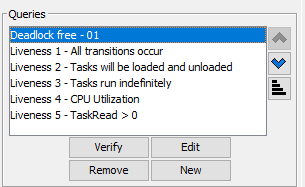


Figura - Verificações realizadas

# CONCLUSÃO

A verificação formal de um sistema operacional é um desafio complexo, mas fundamental, para garantir sua correção, confiabilidade e segurança. Envolve a aplicação de técnicas rigorosas de modelagem e verificação para garantir que o sistema operacional funcione de acordo com suas especificações e que não apresente falhas críticas que possam comprometer a estabilidade do sistema. Com isso, atendendo o objetivo do trabalho, foi possível realizar a verificação de 5 condições:

* Deadlock
* Todas as transições ocorreram (*liveness* 1)
* As tarefas são carregadas e descarregadas da memória (liveness 2)
* As tarefas rodam conforme necessidade (*liveness* 3)
* Utilização correta da CPU (*liveness* 4)
* Que pelo menos uma tarefa esteja sendo executada (*liveness* 5)

Portanto, a combinação da verificação formal com o uso do software TAPAAL pode ser uma abordagem poderosa para garantir a correção e a confiabilidade de sistemas operacionais. Isso é especialmente importante em ambientes críticos, como sistemas embarcados, sistemas de tempo real e servidores, onde a estabilidade do sistema é essencial. A verificação formal desempenha um papel crucial na redução de riscos e na melhoria da segurança e da qualidade dos sistemas operacionais.

REFERÊNCIAS

[1] OTTI, Fernando L.; FOSS, Luciana; RIBEIRO, Leila; SANTOS, Osmar M. Especificação e Verificação Formal de Sistemas Distribuídos. Sociedade Brasileira de Computação, 2003.

[2] TAPAAL Team. TAPAAL: Teaching, Analysis, and Performance Analysis of Algorithms. Versão 3.5. Disponível em: <https://www.tapaal.net/>. Ano de lançamento: 200