

APLICANDO GRAFOS DE CONTROLE DE FLUXO HIERÁRQUICOS PARA GERENCIAR A MODULARIDADE NO SIMULADOR PAMPASIM

PEDRO PORTO SOUZA¹; IGOR GOMES DUTRA²; RAFAEL BURLAMAQUI AMARAL³

¹Universidade Federal de Pelotas — ppsouza@inf.ufpel.edu.br

²Universidade Federal de Pelotas — igdutra@inf.ufpel.edu.br

³Universidade Federal de Pelotas — rafael.amaral@inf.ufpel.edu.br

1. INTRODUÇÃO

O PampaSim¹ é um ambiente para a simulação de Sistemas Operacionais projetado para uso didático (SOARES et al., 2023). Concebido no LUPS (Laboratório de Sistemas Ubíquos e Paralelos) da Universidade Federal de Pelotas (UFPEL), o PampaSim busca preencher a lacuna entre os simuladores de baixa profundidade técnica e de grande acessibilidade, e os simuladores de grande profundidade e baixa acessibilidade, podendo assim ser utilizado para enriquecer a didática das disciplinas de Sistemas Operacionais dos cursos de computação da UFPEL.

As simulações que o PampaSim suporta atualmente são determinísticas, discretas e orientadas à eventos, onde os eventos simbolizam a comunicação entre diferentes entidades, que por sua parte são entendidas como uma representação simplificada das partes de um Sistema Operacional (SO). Durante o desenvolvimento do ambiente, alguns problemas surgiram em virtude do tipo de simulação escolhido, e alguns desses problemas são de particular interesse: como o PampaSim busca permitir a introdução de complexidade às suas simulações de maneira gradual, acompanhando o passo das aulas, foi especificado o uso de conjuntos de entidades opcionais (denominados módulos) que podem ou não fazer parte de um experimento. Essa modularidade traz consigo complicações na determinação da ordem de processamento dos eventos (módulos interdependentes), quais eventos são tratados por quais entidades, e quais eventos devem ser “traduzidos” por serem destinados à entidades de um módulo que não fora incluso num determinado experimento simulado (eventos órfãos). Por consequência, é necessário um modelo interno de fluxo de eventos mais bem-definido e estruturado para solucionar satisfatoriamente esses problemas.

Este trabalho apresenta o desenvolvimento e a implementação de um modelo de Grafos de Controle de Fluxo Hierárquicos (GCFH) no PampaSim para solucionar esses desafios. Nos mesmos moldes que as abordagens de FRITZ; SARGENT (1995) e SCHRIBER; BRUNNER (1997), nosso modelo organiza a simulação em uma estrutura mestre/sub-simulação, garantindo o roteamento correto de eventos e a consistência temporal dos experimentos, independentemente dos módulos ativos.

¹<https://institucional.ufpel.edu.br/projetos/id/u3777>

2. METODOLOGIA

A solução foi desenvolvida após um processo de refatoração incremental do PampaSim, guiado por princípios de modularidade, usabilidade e reprodutibilidade dos experimentos. Esta nova arquitetura adota um modelo hierárquico onde a simulação principal (escalonamento de processos) atua como um módulo “mestre”. Outras funcionalidades, como gerenciamento de memória, são encapsuladas como “sub-simulações” opcionais. A comunicação entre o mestre e as sub-simulações ocorre por meio de um conjunto predefinido de “eventos compartilhados” de entrada e saída. Para lidar com a ausência de um módulo, cada sub-simulação define “regras de tradução” para seus eventos de entrada. Essas regras determinam o que acontece com um evento se o módulo não estiver presente. Por exemplo, um evento pode ser convertido em outro (como $e5 \rightarrow e7$) ou simplesmente descartado ($e6 \rightarrow$ nenhum), garantindo que a ausência do módulo não interrompa a lógica essencial da simulação. A Figura 1 ilustra essa interação.

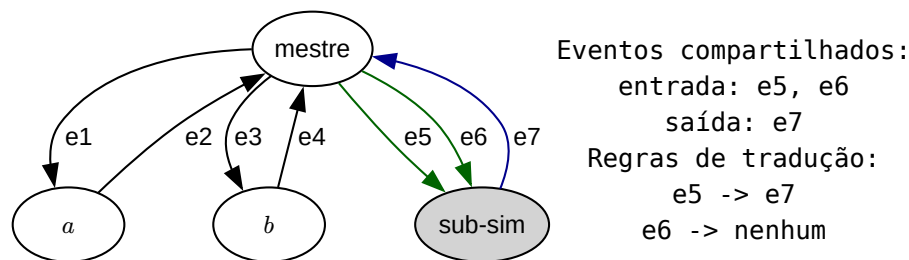


Figura 1: Mecanismo de sub-simulações como módulos opcionais.

Este design representa uma implementação prática dos conceitos de Grafos de Controle de Fluxo Hierárquicos, onde cada sub-simulação atua como um componente encapsulado com uma interface de eventos bem definida.

Relacionando com o simulador, é o caso dos eventos de operações em memória, que podem ser emitidos por um processo escalonado — seu processamento depende da presença do módulo de memória, mas caso o módulo não esteja incluso, algumas operações podem ser ignoradas, como a leitura/escrita em posições de memória.

Contudo, alguns módulos exigem uma alteração no sequenciamento de eventos do fluxo principal de execução, como o evento de despacho de processos `Dispatch` grifado em vermelho na Figura 2; ele faz parte da sequência de eventos que carrega um processo na memória e o prepara para a execução no processador. Nestes casos, as traduções se fazem necessárias para evitar o curto-circuito dos encadeamentos, ou seja, as traduções dão continuidade ao fluxo mesmo quando o módulo que lidaria com o evento não está presente.

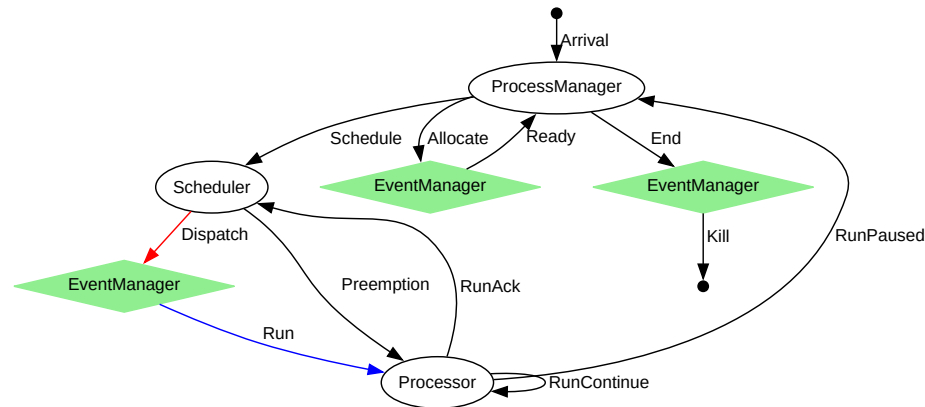


Figura 2: Fluxograma de eventos simplificado do PampaSim

Essa abstração se encaixa bem numa topologia de árvore: a ordem que a árvore de simulação impõe pode ser utilizada para tornar mais intuitiva a lógica de roteamento de eventos. O roteamento é realizado por EventManagers associados às simulações, eles consultam o mapeamento expandindo pelas traduções para determinar a entidade/sub-simulação destino dos eventos. Eventos entre entidades do mesmo nível passam pelo EventManager da simulação mais profunda que os contém, e eventos entre sub-simulações, o conjunto de EventManagers encontrados no caminho mais curto entre fonte e destino.

Em testes realizados ao longo da implementação, após a inclusão de um módulo opcional o simulador apresentou resultados equivocados. Os tempos de início e fim (medidos em ciclos) dos processos eram diferentes da simulação simples, o que não deveria acontecer. Essa inconformidade foi atribuída à falta de distinção entre tempo de sincronia de simulação e tempo simulado, que até então eram tratados como linearmente dependentes. A introdução de *clocks* (variáveis contadoras de tempo) separados para cada tempo e a anotação de eventos chave com um marcador que indica o “consumo de tempo real” resolveu esse problema específico, a cada iteração da simulação que processa ao menos um evento marcado, o *clock* de tempo real é ajustado. Os modelos GCFH formais propõem uma solução mais elegante com componentes atômicos (ACs) e *clocks* locais, uma abstração que planejamos incorporar futuramente para generalizar nosso mecanismo de múltiplos *clocks*.

3. RESULTADOS E DISCUSSÃO

Embora o modelo tenha sido desenvolvido de forma independente, sua arquitetura final demonstrou fortes semelhanças e encontrou respaldo teórico nos Grafos de Controle de Fluxo Hierárquicos descritos por (FRITZ; SARGENT, 1995). As poucas discrepâncias existentes são, em sua maioria, terminológicas ou relativas a conceitos da teoria original, como as “portas de atraso”, que não possuem aplicação direta no PampaSim.

A implementação do modelo hierárquico produziu os seguintes resultados-chave:

- Desacoplamento Modular Efetivo: O problema de alteração no tempo de simulação foi resolvido. Experimentos com e sem o módulo de memória agora produzem

resultados de escalonamento de processos idênticos e consistentes, validando a separação entre tempo de simulação e tempo simulado.

- Roteamento de Eventos Robusto: O problema de roteamento foi solucionado pela hierarquia de instâncias da classe EventManager (o despachante de eventos do PampaSim). Cada sub-simulação gerencia seus eventos internos, e o módulo mestre gerencia a comunicação entre sub-simulações, de forma análoga ao roteamento de pacotes em redes IP, onde cada nó decide o próximo salto.
- Tratamento de Eventos Órfãos: O problema dos eventos de módulos opcionais foi resolvido com o mecanismo de tradução. Os eventos que cruzam a fronteira de um módulo são explicitamente declarados como “compartilhados”. Para cada um, a regra de tradução define seu comportamento na ausência do módulo, seja o descarte (para eventos não essenciais) ou a conversão para um evento genérico de sucesso/falha, garantindo que a simulação prossiga sem erros.

4. CONCLUSÕES

Este trabalho apresenta a concepção e implementação de um modelo de controle de fluxo hierárquico no simulador PampaSim, que soluciona os desafios de complexidade, roteamento e consistência temporal impostos por sua arquitetura modular.

A principal inovação foi a aplicação prática dos princípios de Grafos de Controle de Fluxo Hierárquicos para criar uma estrutura mestre/sub-simulação com regras de tradução de eventos. Essa abordagem não apenas corrigiu inconsistências na simulação, mas também aumentou a robustez e a extensibilidade do ambiente, permitindo que novos módulos de SO sejam desenvolvidos e integrados de forma mais simples e segura. Do ponto de vista pedagógico, a solução reforça o objetivo do PampaSim de permitir um aprendizado incremental.

Como trabalhos futuros, pretende-se evoluir a atual solução de múltiplos clocks explícitos para uma implementação completa do modelo de componentes atômicos com clocks locais, conforme a teoria de GCFH, aprimorando ainda mais a fidelidade e a escalabilidade da simulação.

5. REFERÊNCIAS BIBLIOGRÁFICAS

- FRITZ, D. G.; SARGENT, R. G. **An overview of hierarchical control flow graph models**. Proceedings of the 27th Conference on Winter Simulation. **Anais...**Virginia, USA: IEEE Computer Society, 1995. Disponível em: <<https://doi.org/10.1145/224401.224819>>
- SCHRIBER, T. J.; BRUNNER, D. T. **Inside discrete-event simulation software: How it works and why it matters**. Proceedings of the 29th conference on Winter simulation. **Anais...**1997.
- SOARES, J. A. N. et al. **PAMPAOS: Um Ambiente para a Simulação de Sistemas Operacionais**. XXXII CIC - Congresso de Iniciação Científica, da 9ª Semana Integrada de Inovação, Ensino, Pesquisa e Extensão. **Anais...**Universidade Federal de Pelotas, 2023.