



PAMPAOS: APLICAÇÃO DO PADRÃO DE PROJETO COMMAND

GABRIEL LEITE BESSA¹; HECTOR HUDSON DINIZ FERNANDES¹; JOÃO ANTÔNIO NEVES SOARES¹; LUAN MARK DA SILVA BORELA¹; VINÍCIUS GARCIA PERUZZI¹; RAFAEL BURLAMAQUI AMARAL¹

¹Universidade Federal de Pelotas – {gabriel.lb, hhdfernandes, jansoares, lmdsborela, vgperuzzi, rafael.amaral}@inf.ufpel.edu.br

1. INTRODUÇÃO

O domínio a respeito de Sistemas Operacionais é importante em todos os cursos de informática e computação, pois é nessa disciplina que os estudantes adquirem o entendimento fundamental sobre a interação entre o software e o hardware em níveis mais baixos do sistema. As diretrizes curriculares nacionais para os cursos de graduação em computação (MEC, 2012), prevê que o estudo acerca deste tema é fundamental no currículo dos cursos de graduação em computação. Esse conhecimento desempenha um papel essencial na capacitação dos alunos para o desenvolvimento de aplicações capazes de explorar plenamente os recursos computacionais disponíveis.

Nesse contexto, a utilização de simuladores como ferramentas complementares para consolidar os conhecimentos adquiridos em sala de aula assume grande relevância para o aprimoramento da didática. Para atingir esse propósito, a literatura apresenta algumas propostas de aplicações direcionadas para a simulação do esperado de um Sistema Operacional. Destacam-se, portanto, os simuladores YASS (MUSTAFA, 2013) e SOsim (MAIA, 2015). Entretanto, eles apresentam algumas limitações, como alta complexidade e interfaces desatualizadas.

Dessa forma, o projeto “Desenvolvimento de um ambiente para a simulação de Sistemas Operacionais com o propósito de uso educacional” (AMARAL, 2023) propõe a especificação de um ambiente de simulação de sistemas operacionais, nomeado PampaOS. O principal objetivo é proporcionar uma ferramenta modular e de fácil utilização, permitindo aos usuários configurá-lo de acordo com suas necessidades específicas, tanto dentro quanto fora da sala de aula, visando uma maior absorção dos conteúdos por parte dos alunos.

No âmbito deste trabalho, em particular, a proposta consiste em abordar a implementação da camada de comunicação intermediária, que está sendo desenvolvida para esse simulador, fazendo uso do padrão comportamental, denominado Command (GAMMA et al, 1995). Com este oferecendo o gerenciamento flexível de operações e comandos, os transformando em objetos independentes e permitindo melhor controle sobre sua execução.

Esse padrão nos permite obter características desejáveis sobre os demais estudados que permita, como dito, um maior desacoplamento. Isso é desejável, porque é relevante permitir que cada componente possa ser atualizado e mantido de forma individual e sem necessidade de conhecer os demais elementos do simulador. Os componentes que tiram proveito dessa situação, que podem ser citados, são o escalonador, a interface gráfica, o gerenciador de memória, o gerenciador de discos, gerenciador de dispositivos de entrada e saída, bem como a própria execução na máquina virtual que serve como *Application Programming Interface (API)*.

2. METODOLOGIA



A metodologia adotada para selecionar o padrão em questão envolveu uma pesquisa bibliográfica com o intuito de identificar ferramentas de desenvolvimento capazes de aprimorar o projeto. O objetivo era criar um simulador com características de extensibilidade, facilidade de modificação, manutenção eficiente e alto grau de desacoplamento. Durante a investigação, deparamos com trabalhos que abordaram o Padrão de Projeto Command como uma solução viável para o desacoplamento e aprimoramento da manutenção do código (CARDOSO et al, 2008). Vale destacar que o padrão Command também oferece a capacidade de implementar funcionalidades como "fazer" e "desfazer" (AKIN et al), as quais representam uma abordagem fundamental em um ambiente de simulação.

Esse padrão foi especialmente selecionado, pois em alguns dos casos estudados, ficou aparente que algumas soluções poderiam ser modificadas ou até melhoradas via atualização das funções dos componentes, ou atualização de ferramentas utilizadas para a implementação do projeto, porém, executar essa tarefa é muito difícil, uma vez que não se tenha acesso aos projetos originais e que todo o código tenha componentes que estejam dependentes uns dos outros.

3. RESULTADOS E DISCUSSÃO

Primeiramente, com o objetivo de ter maior independência entre os objetos, foi implementado o padrão de projeto Command, classificado como um padrão comportamental (CARDOSO et al, 2008), capaz de transformar um pedido em um objeto independente, contendo toda informação associada à mensagem. Desta forma, é possível implementar um padrão de troca de mensagens, tais como: classe de origem e destino, ação desejada e qualquer outro argumento extra que a ação possa necessitar. Esta abordagem permite que o tratamento das mensagens seja realizado de forma totalmente independente da classe que as originou.

Na Figura 1, é demonstrado como são as interações entre as classes implementadas no simulador PampaOS. Na qual toda a interação que ocorre entre os componentes principais do sistema passa pelo Command, que é composto pelo Invoker (o invocador que as outras classes chamam) e os Handlers que são os responsáveis pela execução do comando passado ao Invoker utilizando o padrão de mensagens descrito acima. Após isso, os Handlers acionam a classe específica que precisa realizar aquela determinada ação.

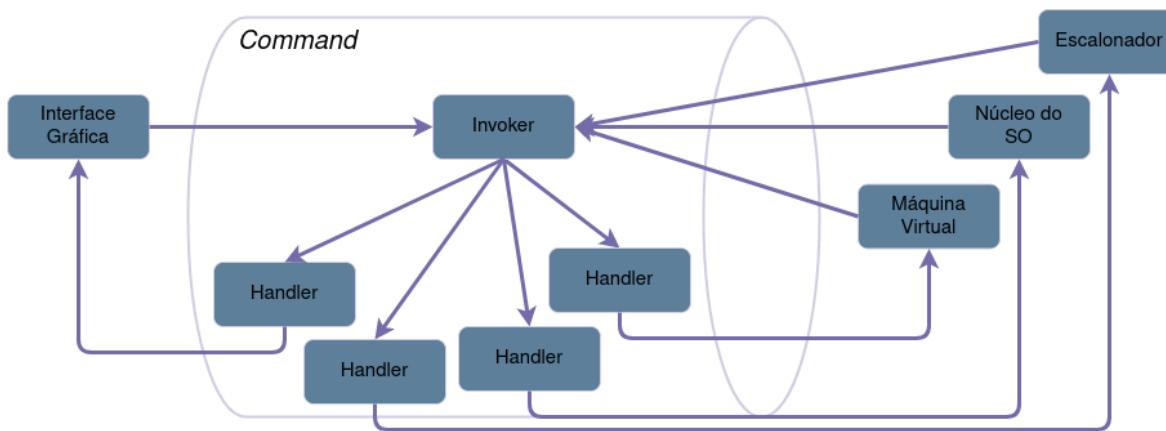


Figura 1 - Demonstração de Funcionamento do Command no PampaOS.
Fonte: Os Autores.

4. CONCLUSÕES

Este trabalho apresentou os avanços da implementação do padrão de projeto Command para construção de um canal de comunicação no contexto do simulador de Sistemas Operacionais PampaOS. Diante disso, podemos concluir que a aplicação do padrão Command demonstrou de forma abrangente seus benefícios, resultando em um notável desacoplamento entre os diversos componentes do sistema. Esse desacoplamento, por sua vez, trouxe consideráveis vantagens em termos de facilitação da manutenção contínua e da expansibilidade do software.

Futuramente pretende-se realizar melhorias na implementação do canal de comunicação proposto para o simulador, como: aplicar a ação de “fazer” e “desfazer”, que o padrão possibilita, e gerar *logs*, aproveitando-se do fato de que grande parte das informações mais importantes circulam por meio das classes do Command.

5. REFERÊNCIAS BIBLIOGRÁFICAS

AMARAL, R. Desenvolvimento de um ambiente para a simulação de Sistemas Operacionais com o propósito de uso educacional. UFPEL. 2023. Acessado em 9 set. 2023. Disponível em: <https://institucional.ufpel.edu.br/projetos/id/u6453>.

AKIN, M.; ERDODAN, N. Using Object Oriented Design Patterns to Develop an Interative Command System for a CAD Software with Undo and Redo Support.

CARDOSO, P.; BARRETO, J.; CARDOSO, L.; HOFFMAN, L. Using Design Patterns, Components and Metadata to Design the Command and Monitoring Frameworks of the INPE’s Satellite Control System. SpaceOps 2008 Conference, 2008.

GAMMA, E.; HELM, R.; JOHNSON, R.; VLISSIDES J. Design Patterns: Elements of Reusable Object Oriented Software, Addison Wesley, 1995.

MEC. Diretrizes Curriculares Nacionais para os cursos de graduação em Computação. 2012. Acessado em 9 set. 2023. Online. Disponível em:



<http://portal.mec.gov.br/escola-de-gestores-da-educacao-basica/323-secretarias-112877938/orgaos-vinculados-82187207/12991-diretrizes-curriculares-cursos-de-graduacao>

MUSTAFA, Besim. YASS: A System Simulator for Operating System and Computer Architecture Teaching and Learning. European Journal of Science and Mathematics Education, v. 1, n. 1, p. 34-42, 2013.

MAIA, L. P. SOsim: Simulador para o Ensino de Sistemas Operacionais. 2001. Dissertação - Informática, Universidade Federal do Rio de Janeiro.

REIS, F. P., J., P. A. P., C., H. A. X., & Integradas—Outsourcing, S. B. S. 2009. TBC-SO/WEB: Um software educacional para o ensino de políticas de escalonamento de processos e de alocação de memória em sistemas operacionais. Simpósio Brasileiro de Informática na Educação-SBIE.