|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Título** | **Resenha** | **Metodologia** | **Pontos Fortes** | **Pontos Fracos** |
| IX: A Protected Dataplane Operating System for High Throughput and Low Latency | Fala sobre um sistema operacional desenvolvido com a intenção de melhorar a performance de aplicações web, como Banco de Dados e servidores, separando o Processamento de Rede do restante do kernel.  Tinham alguns objetivos: um tempo de resposta na faixa de microssegundos e alta taxa de transferência de pacotes, mas sem perder segurança e abstração dada pelo kernel.  Foram obtidos resultados satisfatórios, com performance superior ao Linux e outros sistemas operacionais atuais. | - Baseado em um kernel Linux alterado para obter as características desejadas.  - Foram realizados testes em um cenário parecido com a realidade. | - Há comentários sobre características de gerenciamentos de recursos, como memória.  - Tem uma base conceitual muito forte. | - É muito complexo e exige muito conhecimento técnico de redes para ser compreendido em sua totalidade. |
| ONOS: Towards an Open, Distributed SDN OS | Explica a arquitetura de um sistema distribuído, altamente escalável, focado em performance de rede, e altamente tolerante a falhas. Busca garantir 99.99% de uptime e resposta a 1 milhão de requisições por segundo.  Ao final foi obtido um sistema operacional quase perfeito, e que atende os requisitos propostos, porém possui não foi trabalhado isolamento das rotas e fluxo de dados, o que é considerado por eles uma falha grave. | - Desenvolveram dois protótipos: o primeiro onde se tinha visão de rede, tolerante a falhas e escalável, e o segundo onde foi melhorada a performance.  - Foram realizados testes em servidores geograficamente espalhados pelos Estados Unidos | - Detalham profundamente o sistema operacional desenvolvido  - Os testes realizados mostram resultados promissores, e a metodologia utilizada é detalhada | - Falhou em explicar conceitos e os objetivos propostos  - Complexidade elevada. |
| TrackOS: a Security-Aware Real-Time  Operating System | Mostra o desenvolvimento de um Sistema Operacional em Tempo Real altamente seguro e estável, que busca proteger veículos de acessos não autorizados.  Seu foco é em algoritmos de controle de acesso de memória e integridade dos dados, para evitar que haja alterações no código do Sistema Operacional, que é executado em loop.  Os resultados obtidos foram satisfatórios, porém o equipamento utilizado não permitia isolamento da memória entre os processos, e nem utilizar memória virtual. | - Desenvolvimento de um sistema operacional que é baseado duas funções: fastLoop (que é onde são executadas as funções essenciais) e slowLoop (que são executadas funções secundárias).  - Foi testado utilizando um aeromodelo com piloto automático. | - Possui muita informação acerca de proteção de memória e segurança em sistemas de tempo real  - Explica razoavelmente os conceitos utilizados | - Superficial  - Muita explicação de algoritmos, porém sem mostrar o contexto.  - Complexo. |
| TinyOS: An Operating System for Sensor Networks | Apresenta um Sistema Operacional em tempo real, de pequeno tamanho, criado para ser instalado em nós de uma rede de sensores.  Possui funções síncronas e assíncronas, e possui detecção de condição de corrida. É todo construído sobre uma variante da linguagem C chamada NesC, criada especificamente para este sistema operacional, que quase é totalmente orientada a eventos e específica para sistemas de redes.  Por não ser um sistema monolítico, o resultado obtido é considerado satisfatório, porém possui um caminho longo até ser considerado finalizado. | - Desenvolvimento de uma linguagem de Programação própria para este fim  - Desenvolvimento de um sistema operacional dedicado a gerenciamento de redes, que permita programação concorrente, e livre de condições de corrida  - Não foram realizados testes em cenários reais, apenas foram desenvolvidas aplicações (como Banco de Dados) que se adaptem ao SO. | - Mostra a administração de recursos, técnicas para se evitar erros, condições de corrida e outras falhas comuns em sistemas operacionais. | - Superficial  - Não mostra testes do SO  - Extremamente complexo, e específico. |