



STELLA AZEVEDO MARQUES

**USO DO CHATGPT COMO TUTOR PARA A
CONFIGURAÇÃO DE AMBIENTES DE SIMULAÇÃO DE
REDES DE COMPUTADORES**

LAVRAS – MG

2023

STELLA AZEVEDO MARQUES

**USO DO CHATGPT COMO TUTOR PARA A CONFIGURAÇÃO DE AMBIENTES DE
SIMULAÇÃO DE REDES DE COMPUTADORES**

Dissertação apresentada à Universidade Federal de Lavras, como parte das exigências do Programa de Pós-Graduação em Ciência da Computação, área de concentração em Redes de Computadores, para a obtenção do título de Mestre.

Prof. Dra. RENATA LOPES ROSA
Orientadora

Prof. Dr. Demóstenes Zegarra Rodrigues
Coorientador

**LAVRAS – MG
2023**

Ficha catalográfica elaborada pelo Sistema de Geração de Ficha Catalográfica da Biblioteca
Universitária da UFLA, com dados informados pelo(a) próprio(a) autor(a).

Marques, Stella Azevedo.

Uso do ChatGPT como tutor para a configuração de ambientes
de simulação de redes de computadores / Stella Azevedo Marques. -
2023.

88 p. : il.

Orientador(a): Renata Lopes Rosa.

Coorientador(a): Demóstenes Zegarra Rodriguez.

Dissertação (mestrado acadêmico) - Universidade Federal de
Lavras, 2023.

Bibliografia.

1. Redes de computadores. 2. ChatGPT. 3. Cenários de Redes
Simuladas. I. Rosa, Renata Lopes. II. Rodriguez, Demóstenes
Zegarra. III. Título.

STELLA AZEVEDO MARQUES

**USO DO CHATGPT COMO TUTOR PARA A CONFIGURAÇÃO DE AMBIENTES DE
SIMULAÇÃO DE REDES DE COMPUTADORES**

**UTILIZATION OF CHATGPT AS A TUTOR FOR CONFIGURING COMPUTER NETWORK
SIMULATION ENVIRONMENTS**

Dissertação apresentada à Universidade Federal de Lavras, como parte das exigências do Programa de Pós-Graduação em Ciência da Computação, área de concentração em Redes de Computadores, para a obtenção do título de Mestre.

APROVADA em 22 de novembro de 2023 .

Prof. Dr. Dante Coaquira Begazo Universidade de São Paulo - USP
Prof. Dr. Bruno de Abreu Silva Universidade Federal de Lavras - UFLA

Prof. Dra. Renata Lopes Rosa
Orientadora

Prof. Dr. Demóstenes Zegarra Rodriguez
Co-Orientador

**LAVRAS – MG
2023**

Dedico este trabalho primeiramente a Deus e a meus pais Amarildo e Elenice por estarem sempre ao meu lado. A minha irmã Sther por sempre me apoiar em todas as minhas escolhas. A minha namorada Jessica por ter passado por mais essa etapa junto comigo. A minha família por serem minha base e principalmente às minhas avós Neném e Mariinha, aos meus avós em memória Vô Tião pelas longas horas na calçada contando histórias e Vô Filinho pelo suco com bolo depois das aulas na escola.

AGRADECIMENTOS

Agradeço a Deus por me dar força durante esse processo e por ter me dado mais uma oportunidade de vida. Agradeço a minha mãe por cuidar de mim durante a minha recuperação e me incentivar a não desistir do mestrado. A professora Renata minha orientadora por me dar suporte tanto acadêmico quanto pessoal quando tive que me afastar do mestrado para me recuperar do acidente. Agradeço a psicóloga Carol por me acompanhar na recuperação mental pós acidente. Agradeço a Jessica pelo apoio, carinho e cuidados. Agradeço a minha prima Elem por estar comigo durante minha recuperação. E a minha amiga Nágila pelas caminhadas. E agradeço imensamente a minha tia Glaucia por ser responsável pela minha fisioterapia e reabilitação física, sem essa reabilitação com certeza não seria possível chegar até aqui.

Não desista, eu confio em você! (Elenice de Fátima Azevedo Marques)

RESUMO

Este trabalho aborda a configuração de ambientes de desenvolvimento destinados à análise de redes de computadores, abrangendo a preparação de máquinas virtuais, instalação de softwares e levantamento de material para estudo de simulação de redes de computadores, por meio da utilização do algoritmo *Generative Pre-Trained Transformer* (GPT) na versão 3.5 utilizado como base da ferramenta *ChatGPT* e analisa como o *ChatGPT* é capaz de auxiliar consideravelmente a configuração de ambientes e o estudo de simulações de redes de computadores.

No contexto acadêmico muitos alunos apresentam algum transtorno de aprendizagem, como dislexia, discalculia, autismo ou desordem de atenção (deficit ou hiperatividade). Principalmente no período de pandemia onde os estudantes tiveram que se adaptar a modalidade de aprendizagem remota esses alunos tiveram que encontrar formas de estudar sozinhos, nesse momento a falta de um tutor presente gera grandes dificuldades em explorar novos assuntos e áreas de conhecimento.

É então que surge a necessidade de explorar novas formas de buscar conhecimentos, e a ideia de avaliar o potencial do *ChatGPT* como assistente inteligente. Particularmente seu uso por estudantes iniciantes nos estudos de redes de computadores. Este trabalho busca avaliar a ferramenta *ChatGPT* quanto a respostas fornecendo orientações minuciosas sobre os procedimentos para configuração de ambientes de simulação de redes, explicação sobre nomenclaturas e sugestões de conteúdo para estudo. Utilizando duas formas distintas de solicitação de ajuda, uma onde um único *chat* seria aberto e outra onde para cada tema diferente um novo *chat* é aberto, definimos como tema diferente: a instalação da máquina virtual, a configuração da máquina virtual, a instalação do *Core Emulator*, a instalação do *Wireshark*, definições de termos técnicos e análise dos *logs* gerados pelo *Wireshark*. Como são temas distintos e com configurações extensas ficou claro que solicitar ajuda ao *ChatGPT* abrindo novos *chats* a cada tema apresenta maior assertividade em relação a tentativa de pedir ajuda diretamente em um único *chat*.

É crucial lembrar que a ferramenta possui limitações, tais como a dependência de conectividade à internet e a possibilidade de fornecer respostas imprecisas. Desta maneira, embora a inteligência artificial representada pelo *ChatGPT* seja um recurso valioso, a ferramenta retornou respostas por vezes inconclusivas ou incompletas, mostrando-se assim não é capaz de substituir integralmente a expertise humana de um tutor experiente e permanece como uma ferramenta complementar, muito útil no contexto de estudo de simulação de redes de computadores, também na configuração de ambientes e estudo de protocolos de roteamento.

Palavras-chave: Cenários de Redes Simuladas. ChatGPT. Redes de Computadores. GPT. *Generative Pre-Trained Transformer*

ABSTRACT

This work addresses the configuration of development environments designed for the analysis of computer networks, encompassing the preparation of virtual machines, software installation, and gathering of materials for the study of computer network simulation using the Generative Pre-Trained Transformer (GPT) algorithm in version 3.5, utilized as the foundation of the ChatGPT tool. The study analyzes how ChatGPT significantly aids in the configuration of environments and the study of computer network simulations.

In the academic context, many students face learning disorders such as dyslexia, dyscalculia, autism, or attention disorders (deficit or hyperactivity). Especially during the pandemic period, when students had to adapt to remote learning, these students had to find ways to study independently. At this moment, the absence of a present tutor generates significant difficulties in exploring new subjects and areas of knowledge.

Hence, there arises the need to explore new ways of acquiring knowledge, and the idea of evaluating the potential of ChatGPT as an intelligent assistant. Particularly, its use by students initiating their studies in computer networks. This work seeks to assess the ChatGPT tool in terms of responses, providing detailed guidance on procedures for configuring network simulation environments, explanations of terminology, and content suggestions for study. Using two distinct forms of requesting help, one where a single chat would be opened, and another where a new chat is opened for each different topic, we defined different topics as follows: virtual machine installation, virtual machine configuration, Core Emulator installation, Wireshark installation, definitions of technical terms, and analysis of logs generated by Wireshark. As these are distinct topics with extensive configurations, it became clear that seeking assistance from ChatGPT by opening new chats for each topic demonstrates greater effectiveness compared to attempting to seek help directly within a single chat.

It is crucial to remember that the tool has limitations, such as dependency on internet connectivity and the possibility of providing imprecise answers. Thus, while the artificial intelligence represented by ChatGPT is a valuable resource, the tool sometimes returned inconclusive or incomplete answers, showing that it is not capable of entirely replacing the human expertise of an experienced tutor. It remains a complementary tool, highly useful in the context of studying computer network simulation, as well as in the configuration of environments and the study of routing protocols.

Keywords: Simulated Network Scenarios. ChatGPT. Computer Networks. GPT. Generative Pre-Trained Transformer

LISTA DE FIGURAS

| | |
|---|----|
| Figura 1.1 – Camada de Adaptação de QoS | 14 |
| Figura 1.2 – Pilha de Protocolo TCP/IP | 15 |
| Figura 2.1 – Topologias de rede ponto-a-ponto | 17 |
| Figura 2.2 – Comparação entre o modelo de referência <i>OSI</i> e o modelo de referência TCP/IP | 18 |
| Figura 2.3 – Modelo de <i>firewall</i> aplicado na camada 7 | 18 |
| Figura 2.4 – Direção dos fluxos de dados | 19 |
| Figura 2.5 – Simulação de rede utilizando o <i>software Core Emulator</i> | 19 |
| Figura 2.6 – <i>Wireshark</i> analisador de redes | 20 |
| Figura 4.1 – Passos para rodar a simulação | 30 |
| Figura 4.2 – <i>Script</i> de instalação do <i>Core Emulator</i> gerado pelo ChatGPT | 33 |
| Figura 4.3 – <i>Script</i> de instalação do <i>Wireshark</i> gerado pelo <i>ChatGPT</i> | 33 |

LISTA DE TABELAS

| | |
|--|----|
| Tabela 4.1 – Configurações das máquinas virtuais | 32 |
| Tabela 5.1 – Comparação pelo número de etapas e passos | 35 |

SUMÁRIO

| | | |
|------------|---|-----------|
| 1 | INTRODUÇÃO | 13 |
| 1.1 | Objetivo Principal | 15 |
| 1.2 | Organização do trabalho | 15 |
| 2 | REVISÃO DA LITERATURA | 17 |
| 2.1 | Modelagem de redes de computadores | 17 |
| 2.2 | Escalabilidade de redes de computadores | 20 |
| 2.3 | Algoritmos e protocolos de roteamento | 21 |
| 2.4 | Protocolo OLSR - <i>Optimized Link State Routing</i> | 22 |
| 2.5 | Camada de Rede - A terceira camada do modelo <i>OSI (Open Systems Interconnection)</i> | 22 |
| 2.6 | Definição acerca do que são dados e como podem ser utilizados nos cenários de emulação de redes | 23 |
| 2.7 | Como analisar e validar um cenário simulado | 24 |
| 2.8 | Conceitos como manutenção, atualização, documentação e estudo sobre resultados de análises de tráfego de redes simuladas | 25 |
| 3 | TRABALHOS RELACIONADOS | 27 |
| 3.1 | <i>Generative Pré-trained Transformer - (GPT)</i> | 27 |
| 3.2 | Suporte ao desenvolvedor de software | 27 |
| 3.3 | Suporte técnico e atendimento ao cliente | 27 |
| 3.4 | Documentação técnica | 28 |
| 3.5 | Geração de código | 28 |
| 3.6 | Auxílio em tarefas de engenharia | 29 |
| 3.7 | Artigos que abordam o uso do <i>ChatGPT</i> no estudo de Redes de computadores | 29 |
| 4 | METODOLOGIA DE PESQUISA | 30 |
| 4.1 | Uso do ChatGPT para o estudo de redes de computadores | 31 |
| 4.2 | Maquina virtual | 31 |
| 4.3 | Core Emulator e Wireshark | 32 |
| 4.4 | A instalação das ferramentas | 32 |
| 4.5 | Configuração do ambiente de emulação em um único chat | 32 |
| 4.6 | Configuração do ambiente usando chats específicos | 34 |
| 5 | APRESENTAÇÃO DOS RESULTADOS | 35 |
| 6 | CONCLUSÃO | 39 |
| | REFERÊNCIAS | 40 |

**APENDICE A – O CRITÉRIO USADO PARA ESCOLHER AS PERGUNTAS PARA
CADA ABORDAGEM 43**

A criação de simulações de cenários de redes de computadores é desafiadora devido a diversas complexidades como: a modelagem ter a necessidade de ser precisa e realista em relação aos componentes de rede, como roteadores, *switches* e periféricos, por exemplo como os *smartphones*, computadores e *notebooks*. Além dos inúmeros protocolos e métricas utilizadas para análise de tráfego da rede, é importante sempre se preocupar com escalabilidade dessa rede, que precisa lidar com diferentes tamanhos e complexidades, podendo sobrecarregar os recursos computacionais disponíveis e impactar na análise apresentada.

A obtenção de dados de entrada reais ou a geração de dados sintéticos que representem de forma precisa o ambiente da rede também é uma tarefa crítica. Além disso, a validação dos resultados e a interpretação das conclusões obtidas nas simulações são desafios enfrentados principalmente por iniciantes da área de redes de computadores. Devido ao fato, que por vezes não exista uma resposta "correta" para comparação direta. Portanto, construir simulações de redes de computadores eficazes requer um equilíbrio delicado entre desempenho, escalabilidade, confiabilidade e representações realísticas dos componentes que formam a rede a ser simulada.

Uma métrica comumente analisada é o desempenho, onde uma rede eficaz deve ter largura de banda suficiente para suportar o tráfego dos dados, sem perder sua escalabilidade, ou seja, ser projetada para suportar o crescimento do número de dispositivos conectados a rede, sem perder sua confiabilidade, tendo em si implementado mecanismos de redundância para garantir a continuidade do serviço em caso de falhas de componentes da rede e que seja possível realizar *backups* regularmente para garantir a recuperação de dados em caso dessas falhas.

Levando em consideração toda essas complexidades pode ser uma tarefa árdua para estudantes iniciantes e até pra profissionais com maior expertise, sendo assim um tutor imparcial e disponível se mostra bastante interessante principalmente do ponto de vista acadêmico.

E observando como o *ChatGPT* é conhecido por sua capacidade de desenvolver softwares como visto em Ahmad et al. (2023) e corrigir automaticamente erros em códigos como relatado em Sobania et al. (2023). Além disso, o uso do *ChatGPT* na categorização de falhas foi relatada por Borji (2023). Nos artigos de pesquisa de Borji (2023), Sobiana (2023) e White (2023) foi estudado o desempenho do *ChatGPT* na codificação e programação de um ponto de vista geral de engenharia de software. Vendo todo esse potencial, surgiu a ideia de porquê não utilizar a ferramenta *ChatGPT* no auxílio do estudo de redes de computadores?

Outro problema encontrado durante esta busca, foi a falta de documentação simplificada especialmente para quem está começando os estudos na área de redes de computadores, muitas vezes detalhes são omitidos nos tutoriais encontrados, diversos termos específicos são utilizados e nem sempre são exemplificados, por diversas vezes faltam documentações que expliquem detalhadamente a implementação e o uso dos protocolos disponíveis, da configuração dos ambientes, dos parâmetros e das métricas utilizados para a modelagem da rede de computadores analisada.

Como pode ser observado na dissertação de Rauen (2003), onde a autora afirma que sua escolha em focar no ensino de redes de computadores em seu estudo se justifica pela crescente demanda por profissionais capacitados, impulsionada pela ampla utilização dessas redes em diversas áreas. Essa demanda resulta em uma busca intensificada por cursos mais eficazes, tanto por parte dos profissionais atuantes na área quanto por usuários, alunos e professores. E que apesar do grande interesse no tema, a abordagem dos conteúdos em obras existentes ainda não atende de maneira satisfatória à formação de recursos humanos competentes para o mercado. Depoimentos informais coletados pela autora evidenciam as dificuldades de aprendizagem enfrentadas, com muitos alunos expressando insatisfação por não atingirem um nível adequado de conhecimento, mesmo após cursarem múltiplas disciplinas na área, ministradas por diferentes professores.

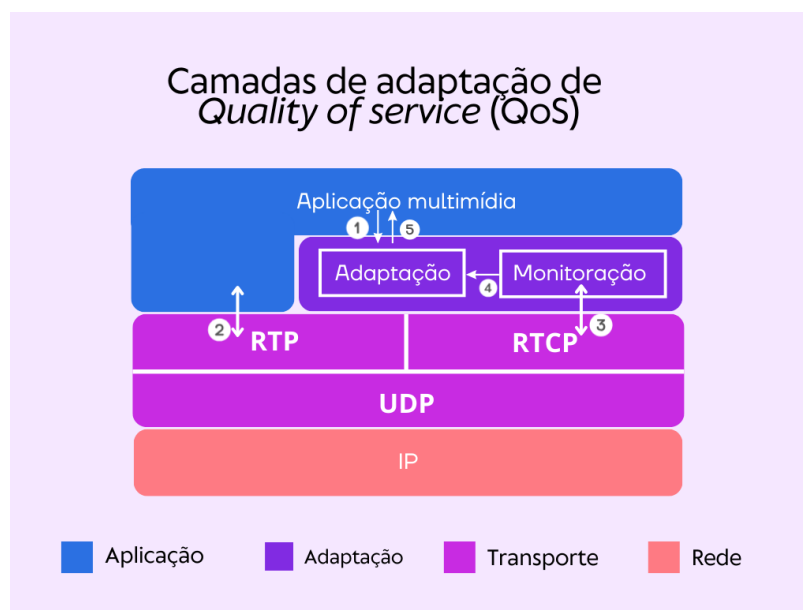
Foi então que levando em consideração a dificuldade encontrada por alunos em iniciar seus estudos na área de redes, em especial da autora em trabalhar com ambientes de simulação de redes de computadores e com a complexidade de se configurar tais ambientes sem a instrução de um tutor qualificado, essa pesquisa deu início no cenário da pandemia da Covid 19 onde o contato humano era

evitado e tornava desafiador o estudo de simulações de redes de computadores sem o auxílio de um tutor humano, o *ChatGPT* se mostrou uma alternativa interessante, porém até o momento sem nenhuma comprovação de que teria utilidade especificamente no estudo de redes de computadores.

Para exemplificação de como o *ChatGPT* pode contribuir na configuração de cenários de simulação de redes de computadores foi escolhido o protocolo OLSR (*Optimized Link State Routing*) por ser um protocolo altamente utilizado para roteamento de redes sem fio, no Capítulo 2 seu funcionamento é abordado com mais detalhamento.

Para fins introdutórios vale descrever que segundo Jacquet et al. (2001) o OLSR é um protocolo de roteamento proativo, que opera na camada de rede, que é a camada 3 do modelo OSI (*Open Systems Interconnection*). Para melhor entendimento Forouzan (2007) define o modelo OSI em 7 camadas, Aplicação que possibilita acesso aos recursos de rede; Apresentação que é responsável por traduzir, criptografar e comprimir dados; Sessão é a camada que estabelece, gerencia e encerra sessões; Transporte deve prover a entrega confiável de mensagens processo a processo e recuperação de erros; Camada de Rede transfere pacotes da origem ao destino e fornece ligação entre redes; Camada de Enlace de dados organiza bits em *frames* e fornece entrega nó a nó; por fim a camada Física responsável por transmitir *bits* através de um meio físico e prover especificações mecânicas e elétricas. Vale ressaltar que o OLSR é um protocolo de roteamento *Ad-Hoc* projetado para redes sem fio móveis, onde os nós podem se comunicar diretamente uns com os outros sem depender de uma infraestrutura de rede fixa. E é responsável por determinar as rotas de comunicação entre os nós na rede, tomando decisões de roteamento com base em informações sobre a topologia da rede e as métricas dos *links*, portanto, ele atua na camada de rede para encaminhar pacotes de dados entre os nós da rede. Segundo Lunardi e Dotti (2001) as setas da figura 1.1 representam:

Figura 1.1 – Camada de Adaptação de QoS



Fonte: Adaptado de Lunardi & Dotti (2001)

- a) Seta 1: Especificação de valores para os parâmetros de QoS *Quality of Service* da aplicação;
- b) Seta 2: fluxo de pacotes de dados RTP (transmissão e recepção);
- c) Seta 3: fluxo de pacotes de controle RTCP (*Sender Reports e/ou Receiver Reports*);
- d) Seta 4: notificação sobre estado da rede, mais parâmetros de QoS (nível de rede);
- e) Seta 5: gatilho para execução das políticas adaptativas.

Como pode ser visualizado na figura 1.1 quando analisamos o funcionamento do OLSR trabalhamos na camada de rede, diretamente na camada de IP conforme pode ser visto na figura 1.2.

Figura 1.2 – Pilha de Protocolo TCP/IP



Fonte: Adaptado de Junior

1.1 Objetivo Principal

Este trabalho tem como objetivo principal demonstrar o uso do *ChatGPT* como tutor para o estudo de redes de computadores, exemplificando seu uso para configuração de ambientes de simulação de redes de computadores, com o foco na replicabilidade desses ambientes.

1.2 Organização do trabalho

O presente trabalho está dividido em 6 capítulos, sendo o capítulo 1 a Introdução e dos demais listados abaixo:

a) Capítulo 2 - apresenta todo o referencial teórico onde são abordados:

- Conceitos sobre modelagem;
- Conceito de escalabilidade de redes de computadores;
- Conceitos sobre algoritmos e protocolos de roteamento;
- Protocolo *OLSR - Optimized Link State Routing*;
- Camada de Rede - a terceira camada do modelo OSI (*Open Systems Interconnection*);
- Definição do conceito de dados e quando são considerados uma entrada realista para um cenário emulado;
- Como analisar e validar um cenário emulado;
- Problemas técnicos como *Overhead* de simulação, utilização de tempo e recursos computacionais, interferência da rede real dentro de uma simulação;
- Conceitos como manutenção, atualização, documentação e estudo sobre resultados de análises de tráfego de rede.

b) Capítulo 3 - são apresentados alguns trabalhos relacionados a este projeto.

- c) Capítulo 4 - aborda a metodologia aplicada para configuração dos ambientes, descrevendo métodos e ferramentas utilizadas para elaboração dos cenários com o uso da ferramenta *ChatGPT*.
- d) Capítulo 5 - apresentação dos resultados exemplificando os resultados obtidos.
- e) Capítulo 6 - finaliza com a conclusão e os próximos passos.

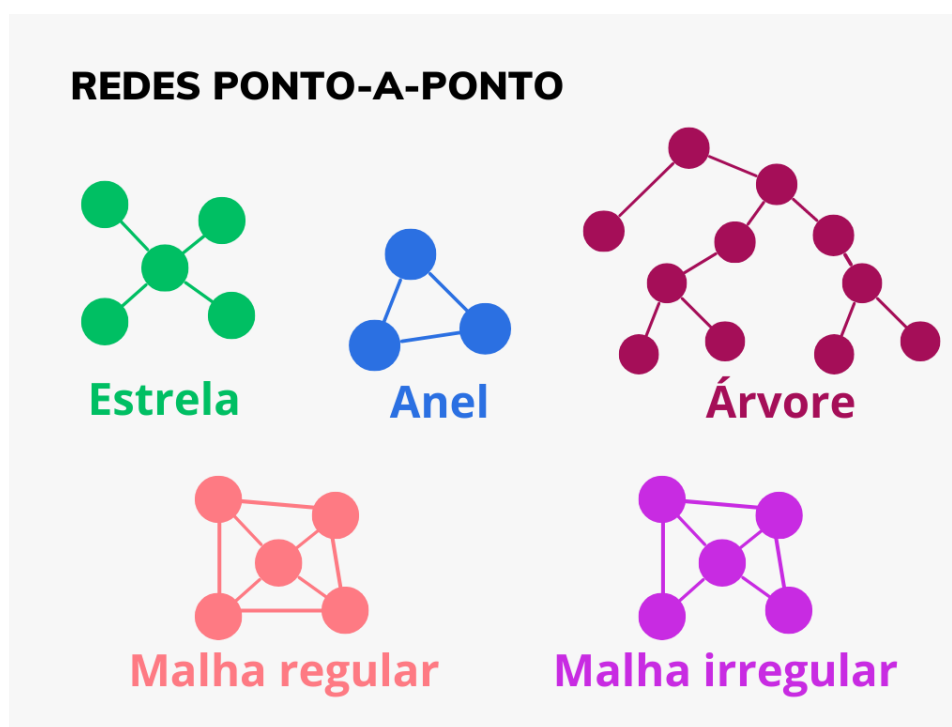
Neste capítulo são apresentados as tecnologias e os conceitos abordados neste trabalho.

2.1 Modelagem de redes de computadores

Modelagem de redes de computadores é o processo de representar uma rede de computadores física ou lógica de maneira abstrata e visual. Ela envolve a criação de um modelo que descreve os componentes da rede, suas interconexões, características de desempenho e comportamento. Esses modelos podem ser usados para uma variedade de fins, incluindo planejamento, projeto, simulação, análise de desempenho e resolução de problemas em redes de computadores.

Aqui estão alguns conceitos-chave relacionados à modelagem de redes de computadores: Modelo de Topologia: Isso descreve como os dispositivos estão fisicamente conectados na rede, incluindo o *layout* da infraestrutura de cabos e dispositivos de rede, como roteadores, *switches* e computadores. Na figura 2.1 alguns exemplos de topologias de rede ponto-a-ponto.

Figura 2.1 – Topologias de rede ponto-a-ponto



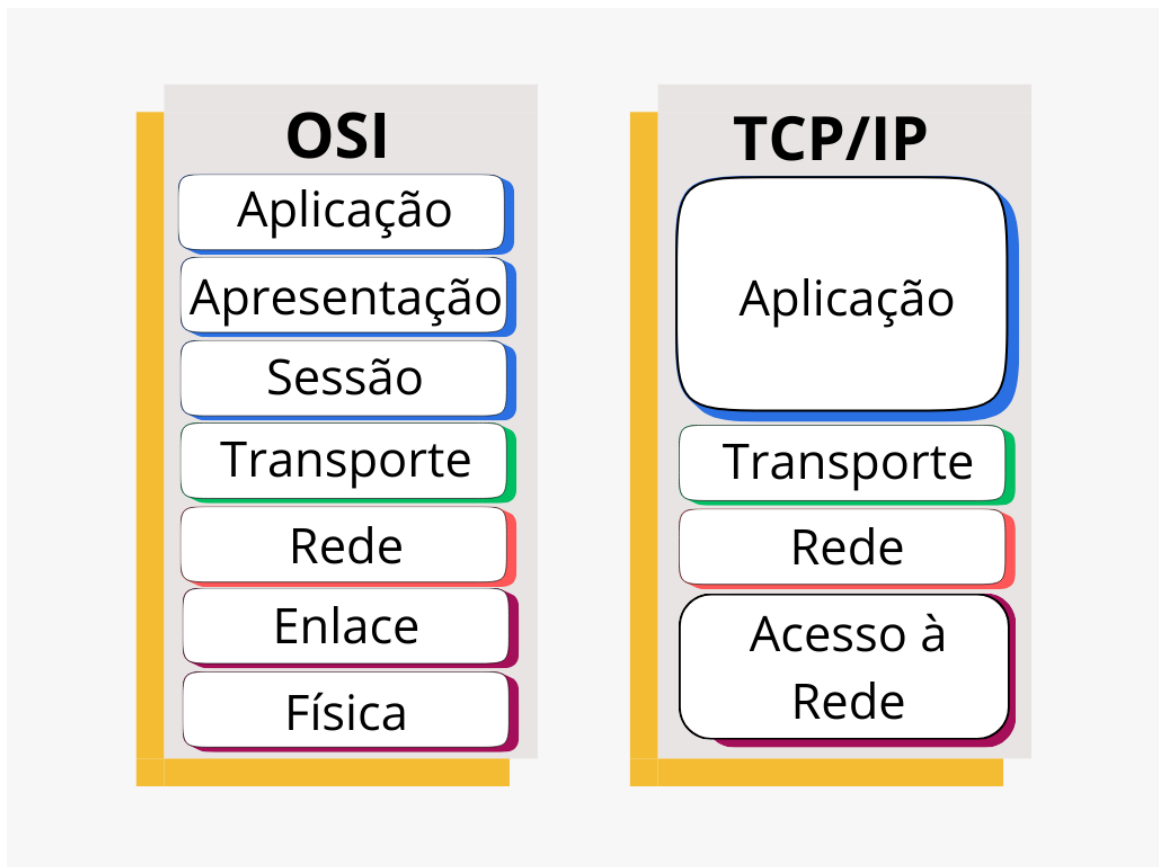
Fonte: Adaptado de Kurose & Ross (2006)

Modelo de Arquitetura: Este é um modelo de alto nível que descreve como os componentes da rede se comunicam e interagem. Pode incluir informações sobre os protocolos de rede utilizados, camadas de comunicação e padrões de comunicação. A figura 2.2 exemplifica duas arquiteturas de redes diferentes o modelo *OSI* e o modelo *TCP/IP*.

Modelo de Desempenho: Este tipo de modelo se concentra nas métricas de desempenho da rede, como largura de banda, latência, taxa de erro e capacidade de processamento. Ele é usado para avaliar e otimizar o desempenho da rede.

Modelo de Segurança: Descreve as políticas e medidas de segurança implementadas na rede, incluindo autenticação, criptografia e controle de acesso. Um exemplo de modelo de segurança bastante difundido é o exemplificado na figura 2.3.

Modelo de Fluxo de Dados: Esse modelo descreve como os dados fluem na rede, desde a origem até o destino, identificando os caminhos e os dispositivos intermediários pelos quais os dados passam.

Figura 2.2 – Comparação entre o modelo de referência *OSI* e o modelo de referência TCP/IP

Fonte: Adaptado de Mendes (2020)

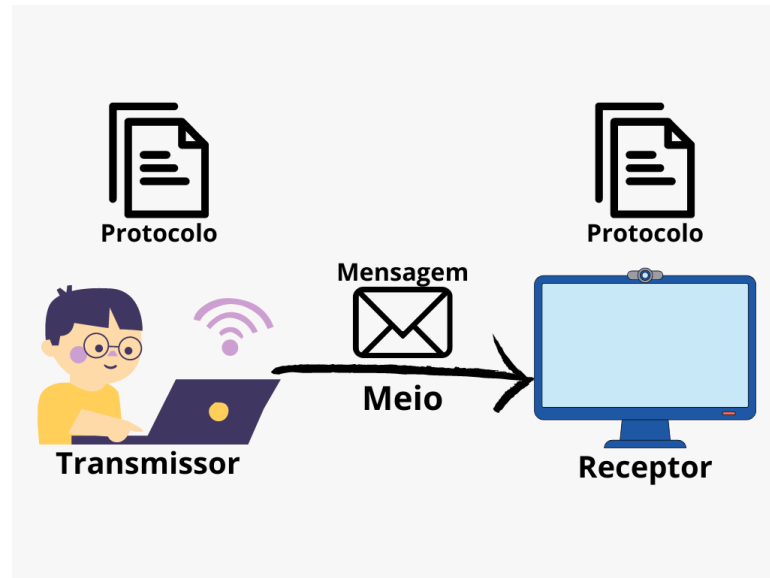
Figura 2.3 – Modelo de *firewall* aplicado na camada 7

Fonte: Adaptado de Xavier (2006)

Na figura 2.4 é exemplificado um modelo unidirecional de fluxo de dados onde o transmissor envia uma mensagem para o receptor.

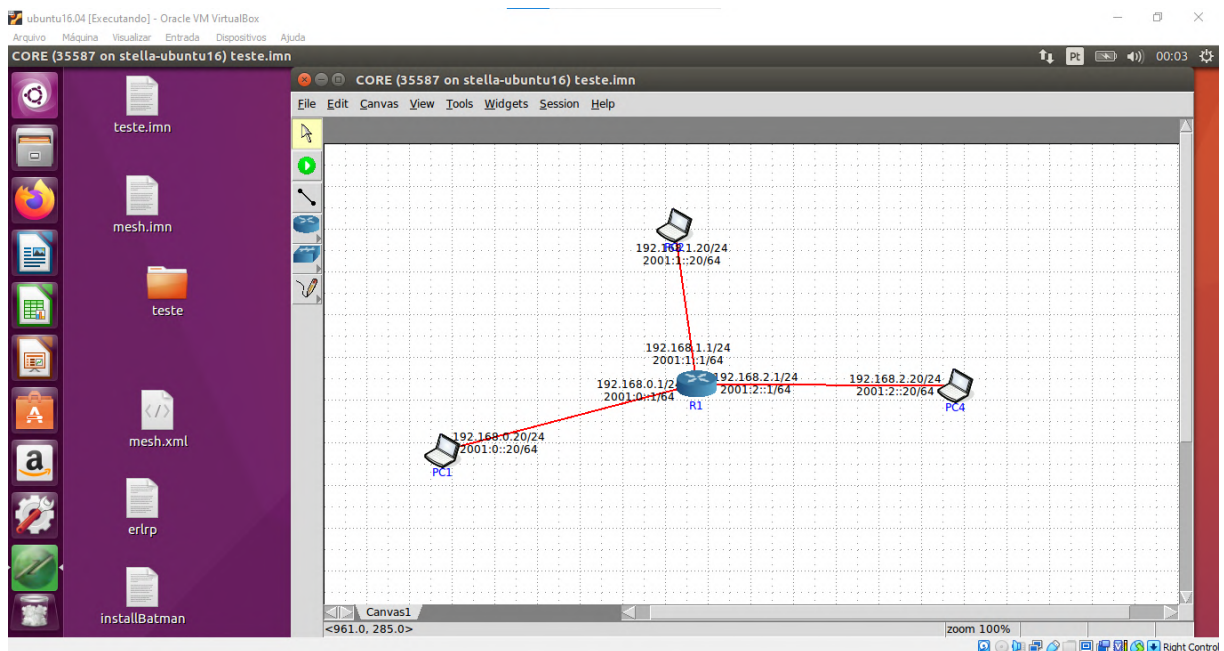
Modelo de Simulação: Modelos de simulação são usados para representar o comportamento da rede em diferentes cenários. Eles permitem testar o desempenho da rede e a eficácia das configurações

Figura 2.4 – Direção dos fluxos de dados



Fonte: Adaptado de Souza (2023)

antes da implementação real. Na figura 2.5 pode ser visualizado a ferramenta *Core Emulator*¹ rodando em uma máquina virtual Linux.

Figura 2.5 – Simulação de rede utilizando o *software Core Emulator*

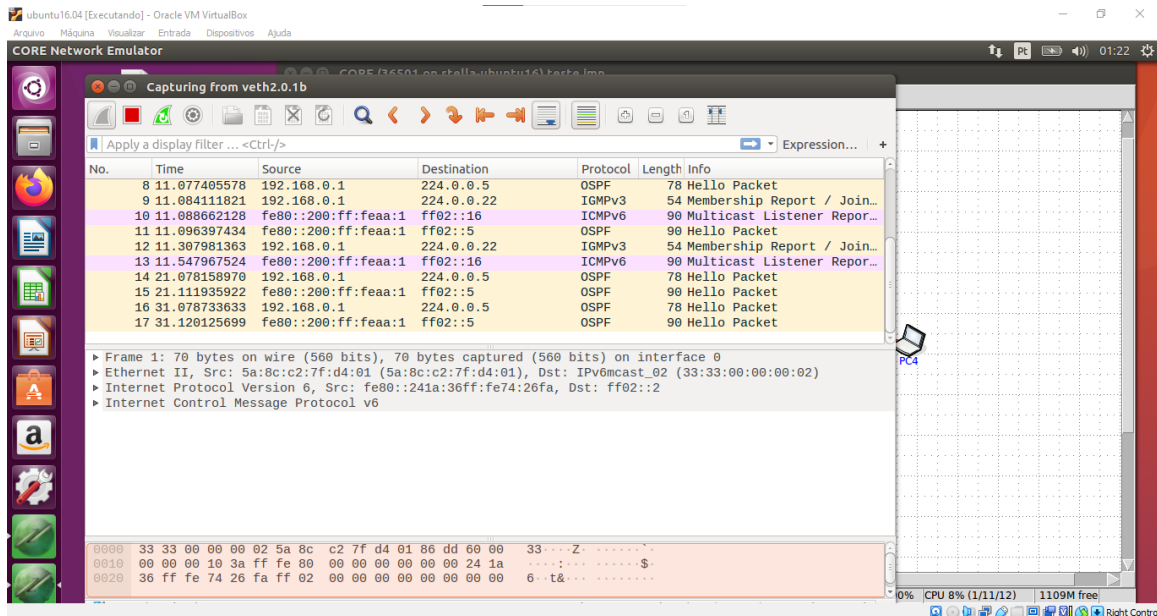
Fonte: A autora (2023)

Modelo de Análise de Tráfego: Esse tipo de modelo analisa os padrões de tráfego na rede, identificando quais aplicativos e protocolos estão em uso e como eles impactam o tráfego de rede. Na

¹ *Core Emulator* - ferramenta para construção de redes virtuais. A documentação completa pode ser acessada em: <https://coreemu.github.io/core/index.html>

figura 2.6 pode ser visualizada uma imagem da ferramenta *Wireshark*² rodando em uma máquina virtual Linux.

Figura 2.6 – *Wireshark* analisador de redes



Fonte: A autora (2023)

Modelo de Gerenciamento de Redes: Descreve como a rede é gerenciada, incluindo o monitoramento de dispositivos, diagnóstico de problemas, configuração e gerenciamento de recursos.

Modelo de Escalonamento: Esse modelo ajuda a determinar como a rede deve ser dimensionada para atender às necessidades atuais e futuras, levando em consideração o crescimento do tráfego e dos dispositivos.

A modelagem de redes de computadores é uma ferramenta valiosa para projetar, operar e otimizar redes de forma eficaz. Ela permite que os engenheiros de rede compreendam melhor o funcionamento de suas redes, tomem decisões informadas e resolvam problemas de maneira mais eficiente. Além disso, a modelagem é uma etapa importante no processo de planejamento e gerenciamento de redes de computadores.

2.2 Escalabilidade de redes de computadores

A escalabilidade em redes de computadores refere-se à capacidade de uma rede crescer e se expandir de forma eficiente para atender às demandas crescentes de tráfego, dispositivos e serviços, ou seja, a escalabilidade pode ser definida como “a facilidade com que um sistema ou componente pode ser modificado para atender a área de um problema” (HENDERSON, 2006). É um princípio fundamental no projeto e gerenciamento de redes, especialmente em um mundo onde as redes estão constantemente evoluindo e se tornando mais complexas. Aqui estão alguns aspectos-chave do conceito de escalabilidade em redes de computadores:

- Capacidade de Crescimento:** Uma rede escalável é projetada para acomodar o aumento no número de dispositivos e usuários sem degradação significativa no desempenho. Isso significa que ela deve ser capaz de lidar com o tráfego adicional sem sobrecarregar seus recursos;

² *Wireshark* - É um programa que analisa o tráfego de rede, e o organiza por protocolos. A documentação completa pode ser acessada em: <https://www.wireshark.org/>

- b) Flexibilidade: Uma rede escalável deve ser flexível o suficiente para se adaptar às mudanças nas necessidades da organização. Isso pode incluir a adição de novos dispositivos, a expansão geográfica da rede ou a introdução de novos serviços;
- c) Desempenho Sustentável: A escalabilidade não se trata apenas de acomodar mais dispositivos, mas também de manter um desempenho aceitável à medida que a rede cresce. Isso envolve a otimização de recursos, como largura de banda e capacidade de processamento;
- d) Redundância e Tolerância a Falhas: Uma rede escalável muitas vezes inclui medidas de redundância e tolerância a falhas para garantir a disponibilidade contínua dos serviços, mesmo em caso de falhas de hardware ou conexão;
- e) Gerenciamento Eficiente: À medida que uma rede cresce, o gerenciamento se torna mais desafiador. A escalabilidade também envolve o uso de ferramentas e práticas de gerenciamento eficientes para facilitar a administração da rede;
- f) Segurança: O crescimento de uma rede também pode aumentar os riscos de segurança. Uma rede escalável deve incorporar medidas de segurança adequadas para proteger os dados e os recursos contra ameaças;
- g) Protocolos e Tecnologias Escaláveis: A escolha dos protocolos e tecnologias certos desempenha um papel crucial na escalabilidade. Protocolos como o IP (*Internet Protocol*) e tecnologias como a virtualização desempenham um papel importante em redes escaláveis;
- h) Dimensionamento Vertical e Horizontal: A escalabilidade pode ser alcançada por meio do dimensionamento vertical (adicionando mais recursos a um único dispositivo) ou do dimensionamento horizontal (adicionando mais dispositivos). O equilíbrio entre essas abordagens depende das necessidades específicas da rede;

Em resumo, a escalabilidade em redes de computadores é fundamental para garantir que as redes possam crescer e se adaptar às mudanças nas necessidades da organização e na demanda de tráfego. É uma consideração crítica no projeto e na manutenção de redes de todos os tamanhos e complexidades.

2.3 Algoritmos e protocolos de roteamento

Segundo Bonaventure et al. (2011), podemos definir protocolos de roteamento como sendo conjuntos de regras e procedimentos que os dispositivos de rede seguem para trocar informações e tomar decisões sobre como encaminhar pacotes de dados de um local para outro em uma rede. Esses protocolos permitem que os dispositivos de rede comuniquem-se uns com os outros para descobrir e manter informações sobre a topologia da rede, calcular rotas eficientes e tomar decisões de roteamento. Exemplos de protocolos de roteamento incluem o OSPF (*Open Shortest Path First*), RIP (*Routing Information Protocol*), BGP (*Border Gateway Protocol*) e EIGRP (*Enhanced Interior Gateway Routing Protocol*).

Ainda segundo Bonaventure et al. (2011), algoritmos de roteamento podem ser definidos como conjuntos de procedimentos e lógicas matemáticas que determinam como os dispositivos de rede calculam as rotas ótimas ou adequadas para encaminhar pacotes de dados. Esses algoritmos usam métricas, como distância, largura de banda, atraso ou custo, para avaliar as rotas possíveis e escolher a melhor rota com base em critérios específicos. Além disso, os algoritmos de roteamento podem ser classificados em duas categorias principais: roteamento por vetor de distância (*Distance-Vector Routing*) e roteamento por estado de enlace (*Link-State Routing*).

Em resumo, os protocolos de roteamento definem como os dispositivos de rede trocam informações e colaboram para tomar decisões de roteamento, enquanto os algoritmos de roteamento representam a lógica matemática usada pelos dispositivos para calcular as rotas ideais com base em métricas específicas. Ambos são componentes fundamentais na operação e no gerenciamento eficiente de redes de computadores.

2.4 Protocolo OLSR - *Optimized Link State Routing*

O *OLSR* é um protocolo de roteamento de malha otimizado para redes sem fio móveis e *Ad-Hoc*. Ele é projetado para eficientemente descobrir e manter as informações de roteamento em redes onde os nós se movem frequentemente e a topologia da rede pode mudar rapidamente. Aqui está um resumo do funcionamento do protocolo OLSR:

- a) Seleção de MPRs (*Multipoint Relays*): O OLSR utiliza o conceito de MPRs para reduzir a sobrecarga de controle na rede. Os MPRs são nós selecionados de forma inteligente que retransmitem pacotes de controle, reduzindo assim a quantidade de tráfego de controle necessário. Os nós MPR são escolhidos para cobrir eficazmente a rede e garantir a conectividade;
- b) Manutenção de Tabelas de Vizinhos: Os nós mantêm informações sobre os vizinhos diretos na rede sem fio, incluindo informações sobre a qualidade do link, como a taxa de perda de pacotes. Essas informações são usadas para calcular as métricas dos links;
- c) Troca de Mensagens de Controle (*HELLO e TC*): Os nós na rede trocam mensagens periódicas de controle, como mensagens *HELLO* para detectar vizinhos diretos e mensagens *TC (Topology Control)* para divulgar informações sobre o estado da topologia da rede. As mensagens *TC* contêm informações sobre os MPRs;
- d) Cálculo de Rotas: Com base nas informações coletadas, os nós *OLSR* calculam rotas eficientes para alcançar outros nós na rede. Isso é feito usando um algoritmo de roteamento que leva em consideração a topologia da rede, as métricas dos links e as informações dos MPRs;
- e) Manutenção de Tabelas de Roteamento: Os nós mantêm tabelas de roteamento que armazenam informações sobre as rotas disponíveis para alcançar outros destinos na rede. Essas tabelas são atualizadas dinamicamente à medida que a topologia da rede muda;
- f) Reativação de Rotas: O *OLSR* é projetado para ser proativo na atualização das informações de roteamento, mas também suporta a reativação de rotas quando necessário. Isso significa que, se uma rota se tornar inválida devido a mudanças na topologia da rede, o *OLSR* pode detectar essa condição e tentar encontrar uma nova rota automaticamente;

Em resumo, o protocolo *OLSR* utiliza uma combinação de seleção de MPRs, troca de mensagens de controle e cálculos de roteamento para facilitar a descoberta e manutenção eficiente de rotas em redes sem fio móveis e *Ad-hoc*. Isso ajuda a minimizar a sobrecarga de controle e a adaptar-se às mudanças na topologia da rede.

2.5 Camada de Rede - A terceira camada do modelo OSI (*Open Systems Interconnection*)

A camada 3 é a terceira camada do modelo *OSI (Open Systems Interconnection)*, que é um modelo de referência amplamente usado para entender e padronizar as funções de uma rede de computadores. A camada 3 é conhecida como "Camada de Rede" (*Network Layer*) e desempenha um papel fundamental no roteamento de pacotes de dados através de redes.

Segundo Nascimento (2018), o modelo *OSI* é composto por sete camadas, foi desenvolvido com base em princípios específicos. Cada camada é criada para atender a diferentes níveis de abstração, executar funções bem definidas e seguir protocolos internacionalmente padronizados. Os limites entre as camadas são estabelecidos para minimizar o fluxo de informações entre interfaces. O número de camadas é equilibrado, sendo grande o suficiente para evitar a colocação desnecessária de funções distintas na mesma camada, mas pequeno o bastante para manter a arquitetura controlável. As principais funções da Camada de Rede (camada 3) incluem:

- a) Roteamento: Esta camada é responsável por determinar as rotas que os pacotes de dados devem seguir de um nó de origem para um nó de destino através da rede. Ela faz isso com base em tabelas de roteamento e informações sobre a topologia da rede.
- b) Endereçamento: A camada de rede usa endereços lógicos (como endereços IP) para identificar dispositivos e redes na rede. Ela fornece um sistema de endereçamento que permite que os pacotes sejam roteados com precisão;
- c) Encaminhamento: O encaminhamento é o processo de selecionar o próximo salto (nó intermediário) para encaminhar um pacote em direção ao seu destino. Isso é feito com base nas informações de roteamento disponíveis na camada de rede;
- d) Fragmentação e Remontagem: A camada de rede pode fragmentar pacotes de dados em partes menores, se necessário, para a transmissão através de *links* de menor Unidade Máxima de Transmissão (MTU - *Maximum Transmission Unit*) e, em seguida, remontá-los no destino;
- e) Controle de Fluxo: Em alguns casos, a camada de rede pode ser responsável por controlar o fluxo de tráfego para evitar congestionamentos na rede;

Outros exemplos de protocolos que operam na camada de rede incluem o protocolo IP (*Internet Protocol*) e seus derivados, como o IPv4 e o IPv6. Esses protocolos desempenham um papel fundamental na comunicação global na Internet e em muitas outras redes de computadores.

2.6 Definição acerca do que são dados e como podem ser utilizados nos cenários de emulação de redes

Dados são informações ou fatos brutos que podem ser armazenados e processados em computadores. Esses dados podem ser estruturados ou não estruturados e podem representar uma ampla variedade de informações, como números, texto, imagens, áudio, vídeo e muito mais. Segundo os autores Semeler e Pinto "dados de pesquisa precisam ser identificáveis, citáveis, visíveis, recuperáveis, interpretáveis, contextualizáveis, interoperáveis e reutilizáveis onde quesitos de consistência e procedência são considerados"(SEMELER; PINTO, 2019). No contexto de simulação de redes, os dados podem ser utilizados de várias maneiras.

Quando pensamos na modelagem da topologia de rede, os dados podem ser usados para representar a topologia da rede, incluindo informações sobre os dispositivos de rede (como roteadores, *switches* e computadores), suas conexões físicas e lógicas, endereços IP, máscaras de sub-rede e muito mais. Esses dados são fundamentais para a criação de modelos precisos de redes em simulações.

Ao analisarmos os dados de tráfego, podemos visualiza-los como padrões de comunicação entre dispositivos, volumes de dados transmitidos, tipos de protocolos usados e informações de qualidade de serviço (QoS), são essenciais para simular como os dados reais são transmitidos e processados em uma rede. Já os dados de desempenho, como latência, largura de banda, taxa de perda de pacotes e outros parâmetros de qualidade de rede, são utilizados para avaliar o desempenho da rede simulada. Esses dados ajudam a determinar se a rede está funcionando de maneira eficiente.

Existem também os dados de configuração de dispositivos, que podem incluir informações de configuração de dispositivos de rede, como configurações de roteadores, políticas de firewall, tabelas de roteamento e outras configurações específicas de dispositivo. Isso permite que a simulação represente com precisão o comportamento real dos dispositivos.

Os dados de segurança, principalmente os relacionados a tráfego malicioso, em cenários de simulação de segurança de rede, dados sobre ameaças, ataques cibernéticos, padrões de tráfego malicioso e vulnerabilidades podem ser usados para testar a resiliência da rede e a eficácia das medidas de segurança.

Um tipo de dado importante quando falamos de simulações são os de monitoramento e de análise, durante a simulação, os dados de monitoramento podem ser coletados para avaliar o comportamento da

rede em tempo real. Isso inclui dados de monitoramento de tráfego, *logs* de eventos, estatísticas de desempenho e muito mais.

Após as simulações, os dados de relatórios e análise pós-simulação, são aqueles dados coletados durante a execução e podem ser analisados para avaliar o desempenho da rede, identificar gargalos, propor melhorias e tomar decisões sobre a infraestrutura de rede.

Em resumo, os dados desempenham um papel fundamental em cenários de simulação de redes, fornecendo informações críticas para modelar, analisar e avaliar o comportamento de redes de computadores em diversos contextos, desde testes de desempenho até simulações de segurança e treinamento de protocolos.

2.7 Como analisar e validar um cenário simulado

A análise e validação de um cenário emulado em simulações de redes de computadores são etapas cruciais para garantir que os resultados da simulação sejam confiáveis e representem de forma precisa o comportamento esperado da rede. Aqui estão algumas diretrizes para analisar e validar um cenário emulado:

- a) **Defina Objetivos Claros:** Antes de iniciar a simulação, estabeleça objetivos claros. Saiba o que você está tentando alcançar com a simulação e quais métricas ou resultados são relevantes para sua análise;
- b) **Configure Cenário de Teste Realista:** Certifique-se de que o cenário emulado reflète com precisão a topologia da rede, a carga de tráfego, as configurações de dispositivos e outras características da rede real que você está tentando simular;
- c) **Colete Dados de Simulação:** Durante a simulação, colete dados relevantes, como estatísticas de desempenho, latência, largura de banda, taxa de perda de pacotes e outros parâmetros de interesse. Use ferramentas de monitoramento e captura de pacotes, se necessário;
- d) **Compare com o Comportamento Real:** Compare os resultados da simulação com dados reais, quando disponíveis. Isso pode envolver a coleta de dados da rede real ou a comparação com resultados anteriores de testes reais, se disponíveis;
- e) **Avalie a Precisão:** Avalie a precisão da simulação em relação aos resultados reais ou expectativas. Verifique se as métricas simuladas estão dentro de margens aceitáveis de erro em relação aos valores reais;
- f) **Identifique Anomalias e Problemas:** Procure por anomalias, problemas de desempenho ou comportamento inesperado na simulação. Isso pode incluir congestionamento, tempos de resposta excessivos, erros de roteamento e assim por diante;
- g) **Ajuste Parâmetros:** Se a simulação não corresponder aos resultados reais, faça ajustes nos parâmetros do cenário, como configurações de dispositivos, carga de tráfego, taxa de transmissão, etc., e execute a simulação novamente;
- h) **Execute Testes de Estresse e Limites:** Execute testes de estresse e limites para verificar como o cenário emulado se comporta sob condições extremas. Isso ajuda a identificar possíveis pontos de falha na rede;
- i) **Documente Resultados e Conclusões:** Documente todos os resultados da simulação, incluindo métricas, configurações e qualquer observação relevante. Isso é importante para referência futura e relatórios;
- j) **Valide com Especialistas:** Se possível, obtenha a validação e *insights* de especialistas em redes ou colegas que estejam familiarizados com a rede real ou o cenário que está sendo simulado;

- k) Itere conforme Necessário: Esteja preparado para iterar e ajustar seu cenário de simulação várias vezes até alcançar resultados confiáveis e precisos;
- l) Replicabilidade: Certifique-se de que outros possam replicar seus experimentos e validar seus resultados. Isso inclui a documentação completa do cenário, configurações e procedimentos de simulação;
- m) Teste de Sensibilidade: Realize testes de sensibilidade, onde você varia um ou mais parâmetros de entrada para ver como eles afetam os resultados da simulação;

Lembre-se de que a validação de cenários emulados em simulações de redes é um processo detalhado e muitas vezes iterativo. Quanto mais próximo o cenário emulado estiver da realidade, mais confiáveis serão os resultados da simulação. A documentação adequada e a validação rigorosa são fundamentais para garantir que as simulações forneçam informações úteis e confiáveis.

Este trabalho é focado no problema da Replicabilidade, pois diversas vezes é uma tarefa árdua replicar simulações de cenários em ambientes diferentes, visto que com pequenas modificações podem causar grandes efeitos colaterais, como por exemplo a atualização de versão de uma ferramenta pode gerar incompatibilidade com o protocolo analisado, nesse cenário o *ChatGPT* se mostra uma ferramenta muito interessante, pois tem acesso a uma vasta documentação e consegue de forma bastante precisa sanar grande parte das dúvidas que possam surgir durante a modelagem e configuração dos cenários de rede a serem simulados.

2.8 Conceitos como manutenção, atualização, documentação e estudo sobre resultados de análises de tráfego de redes simuladas

Em cenários de simulação de tráfego de rede, os conceitos de manutenção, atualização, documentação e estudo de resultados desempenham papéis essenciais para garantir a validade e a utilidade das simulações. A manutenção do ambiente de simulação, refere-se ao cuidado contínuo do ambiente de simulação, incluindo hardware, software e configurações. Isso envolve a garantia de que os dispositivos de rede virtuais, as ferramentas de simulação e os recursos de infraestrutura estejam funcionando adequadamente.

Já a manutenção dos dados de simulação, envolve a coleta e a manutenção contínuas dos dados de simulação, como topologia de rede, configurações de dispositivos, modelos de tráfego e cenários. Os dados devem ser atualizados conforme necessário para refletir mudanças na rede simulada.

Outra manutenção que despende bastante conhecimento e tempo, é a manutenção de *scripts* e configurações. Os *scripts* e configurações de simulação devem ser mantidos e ajustados conforme necessário para garantir que a simulação continue produzindo resultados precisos.

Um conceito que gera algumas complicações em simulações de redes são as atualizações. Como por exemplo a medida que as condições da rede real evoluem, os modelos de tráfego utilizados na simulação devem ser atualizados para refletir essas mudanças. Isso inclui alterações nos padrões de tráfego, na largura de banda disponível, nas taxas de transmissão, entre outros.

Atualização de parâmetros de simulação, também são comuns e costuma acarretar em grandes taxas de erro. Atualizar a latência da rede e/ou o *overhead* pode gerar grandes impactos na estrutura simulada.

Outro problema comum é a atualização das ferramentas de simulação, à medida que novas versões de ferramentas de simulação são lançadas, é importante considerar a atualização para aproveitar melhorias, correções de *bugs* e recursos adicionais. Porém essas atualizações podem acarretar em incompatibilidades com a rede simulada, nesse quesito o *ChatGPT* pode se mostrar um grande aliado.

Uma parte importante das simulações de redes de computadores é a documentação da configuração dessa simulação. Isso inclui detalhes sobre como a simulação foi configurada, quais modelos foram usados, configurações de dispositivos, parâmetros de simulação e quaisquer suposições feitas.

Como parte dessas documentações, é realizada a documentação dos resultados. Onde é descrito detalhadamente os resultados da simulação, incluindo métricas de desempenho, estatísticas de tráfego,

padrões identificados e quaisquer descobertas importantes. Uma documentação muito importante é a de alterações, manter um registro de todas as atualizações, ajustes e modificações feitas na simulação ao longo do tempo. Isso ajuda a rastrear a evolução da simulação.

Por fim o estudo sobre resultados da simulação pode ser dividido em três etapas a análise, a validação e a tomada de decisão. Na análise, deve-se avaliar os resultados da simulação para identificar tendências, comportamentos, problemas e oportunidades de melhoria. Isso pode incluir a identificação de gargalos de rede, otimizações de desempenho, avaliação de protocolos de roteamento e muito mais. Na validação, compare os resultados da simulação com dados reais, se disponíveis, para validar a precisão da simulação. Isso ajuda a garantir que a simulação seja uma representação fiel da rede real. Com base nos resultados da simulação, tome decisões relacionadas a mudanças de rede, como otimizações, investimentos em infraestrutura e políticas de gerenciamento de tráfego.

Esses conceitos são fundamentais para garantir que as simulações de tráfego de rede sejam confiáveis, eficazes e úteis para a análise e o planejamento de redes de computadores. A documentação adequada é especialmente importante para rastrear todas as etapas da simulação e facilitar a comunicação de resultados e descobertas com outras partes interessadas.

Na seção seguinte, de trabalhos relacionados são apresentados alguns trabalhos científicos que comprovam que o uso do *ChatGPT* como tutor é amplamente difundido em diversas áreas, porém como mencionado até o momento na literatura o *ChatGPT* não foi encontrado pela autora nenhuma menção os seu uso como ferramenta auxiliar para a modelagem e configuração de redes simuladas de computadores.

Neste capítulo são apresentados alguns trabalhos relacionados demonstrando a eficácia do ChatGPT no auxílio de diversas tarefas técnicas.

3.1 *Generative Pré-trained Transformer - (GPT)*

O *Generative Pre-trained Transformer (GPT)* é uma arquitetura avançada de modelo de linguagem desenvolvida pela OpenAI, que opera em duas fases principais. Inicialmente, o modelo passa por um estágio de pré-treinamento extenso em grandes conjuntos de dados textuais, durante o qual busca aprender padrões e estruturas linguísticas. Essa fase não está vinculada a tarefas específicas, mas visa dotar o modelo da compreensão profunda da linguagem. Fundamentado na arquitetura *Transformer*, o *GPT* utiliza mecanismos de atenção *multihead* para processar sequências de dados, permitindo assim a captura de relações de longo alcance entre palavras. A auto-regressividade é uma característica chave durante a geração de texto, onde o modelo gera palavras sequencialmente, condicionadas ao contexto anterior, e a decodificação probabilística é empregada para calcular as probabilidades de palavras futuras. Após o pré-treinamento, o *GPT* pode ser afinado para tarefas específicas, como tradução ou questionamento e resposta, aproveitando a capacidade de transferência de conhecimento do modelo. Com suas diversas camadas e blocos, o *GPT* destaca-se pela capacidade de gerar texto coerente e relevante em diferentes contextos, consolidando-se como uma abordagem poderosa na modelagem de linguagem.

Segundo Brown et. al (2020), o GPT-3 é treinado com uma mistura de conjuntos de dados contendo 400 bilhões de *tokens* e possui até 175 bilhões de parâmetros. Comparado ao seu precursor GPT-2, o GPT-3 é capaz de aprendizado com poucos exemplos (*few-shot learning*), onde o modelo aprende a partir de múltiplos exemplos da tarefa de PNL (Processamento de Linguagem Natural) ou NLG (Geração de Linguagem Natural) chamados de *prompts*. Nesse processo, nenhuma atualização de gradiente é realizada.

3.2 Suporte ao desenvolvedor de software

A ferramenta *ChatGPT* se mostra eficiente em sugerir código-fonte relevante durante o desenvolvimento, economizando tempo e prevenindo erros. Também se mostra eficiente ao auxiliar os desenvolvedores na solução de problemas técnicos, fornecendo soluções ou depuração. Um excelente exemplo é mostrado no artigo denominado "ChatGPT for Programming Numerical Methods" de Kashefi & Mukerji (2023) desenvolvem uma análise sobre o uso do *ChatGPT* na programação de algoritmos numéricos. São investigadas várias áreas, como a geração de código em diferentes linguagens, depuração, complementação de partes ausentes, reescrita em linguagens alternativas e paralelização de código serial. Além disso, o estudo analisa a capacidade do *ChatGPT* de distinguir entre código humano e gerado por máquinas, com foco em problemas matemáticos e aplicações de aprendizado de máquina científico. Embora o *ChatGPT* tenha mostrado sucesso em muitos aspectos, também enfrenta desafios, como a geração de matrizes singulares e interrupções em tarefas de código longo, indicando a necessidade de melhorias contínuas.

3.3 Suporte técnico e atendimento ao cliente

Outro uso que vem se popularizando é o *ChatGPT* ser treinado para responder a perguntas frequentes e auxiliar no atendimento ao cliente. Realizando a triagem de problemas, podendo ajudar a identificar problemas técnicos com base nas descrições dos usuários. No artigo "A Review of ChatGPT AI's Impact on Several Business Sectors" os autores George & George (2023) os autores apresentam a seguinte afirmação: "O *ChatGPT*, um modelo de processamento de linguagem natural desenvolvido pela OpenAI, promete ser uma inovação impactante. Ele combina o GPT-2 com aprendizado supervisionado e por reforço, permitindo interações naturais por texto. Suas aplicações abrangem diversos setores, incluindo *e-commerce*, educação, entretenimento, finanças, saúde e notícias. Oferece recursos

como detecção de tópicos, emoções e análise de sentimentos, melhorando a personalização do conteúdo e tornando o atendimento ao cliente mais eficiente."Deixando claro como os autores enxergam que a ferramenta *ChatGPT* pode ser utilizada de forma eficiente no atendimento a clientes.

3.4 Documentação técnica

Uma excelente função para o *ChatGPT* é a geração de documentação, podendo auxiliar na criação automática de documentação técnica com base em especificações ou código-fonte. O *ChatGPT* pode ser uma ferramenta valiosa para auxiliar na criação de documentação técnica de sistemas de várias maneiras: Através da geração de Conteúdo, onde o *ChatGPT* pode gerar texto técnico de alta qualidade, o que pode ser útil para criar partes da documentação, como descrições de recursos, guias do usuário e documentação de *API*. Ou na explicação de conceitos complexos, onde ele pode ajudar a simplificar conceitos técnicos complexos, tornando-os mais acessíveis aos leitores que podem não estar tão familiarizados com o sistema em questão. Outra forma de auxílio em documentações é através de respostas a perguntas técnicas. Os desenvolvedores e redatores de documentação podem fazer perguntas específicas sobre o sistema e obter respostas técnicas detalhadas do *ChatGPT*.

O *ChatGPT* se faz muito útil na revisão e edição de documentos técnicos, onde pode ser usado para revisar e editar o conteúdo da documentação, identificando erros gramaticais, problemas de clareza e oferecendo sugestões de melhoria. Se a documentação precisa ser traduzida para outros idiomas, o *ChatGPT* pode ajudar na tradução, embora seja importante revisar o trabalho gerado para garantir a precisão. Um uso menos comum é a geração de diagramas, embora o *ChatGPT* seja principalmente uma ferramenta de geração de texto, ele também pode ser usado para criar descrições que podem ser usadas na criação de diagramas técnicos, fluxogramas e outros recursos visuais da documentação.

Se houver debates ou decisões técnicas a serem tomadas na documentação, o *ChatGPT* pode fornecer análises e informações relevantes para ajudar a equipe a tomar decisões informadas e à medida que o sistema evolui, a documentação precisa ser atualizada. O *ChatGPT* pode ajudar a gerar atualizações rápidas e precisas para a documentação existente. No entanto, é importante ter em mente que o *ChatGPT* deve ser usado como uma ferramenta de auxílio e não como substituto para o conhecimento humano. A documentação técnica geralmente requer conhecimento detalhado e preciso, e o conteúdo gerado pelo *ChatGPT* deve ser revisado e validado por especialistas técnicos para garantir sua precisão e qualidade.

Além disso, a documentação técnica deve ser organizada de forma lógica e estruturada, e o *ChatGPT* pode não ser tão eficaz quanto um ser humano na criação dessa estrutura. Portanto, é importante que a equipe de documentação continue desempenhando um papel ativo na organização e na formatação da documentação.

3.5 Geração de código

No artigo publicado no site da empresa *Visure Solutions* denominado "Aproveitando o *ChatGPT* para casos de uso no desenvolvimento de *software*"(SOLUTIONS, 2023), descreve como o uso do *ChatGPT* pode revolucionar vários aspectos do ciclo de vida do desenvolvimento de software. Simplificando o suporte e documentação até a automação e revisão de códigos. E afirmam que o *ChatGPT* tem o potencial de remodelar a maneira como o software é desenvolvido, tornando-o mais eficiente, colaborativo e inovador.

Que a ferramenta apresenta suporte para diversas linguagens, não é novidade, porém vale deixar claro que existe a limitação quanto ao tamanho dos códigos gerados, se mostrando uma eficiente ferramenta para geração de pequenos códigos e ajustes em lógicas mais simples. Porém, sem dúvidas um dos usos mais populares da ferramenta *ChatGPT* sem dúvidas é o seu poder de gerar códigos para automatizar tarefas técnicas como, *design* de sistemas e arquitetura de *softwares*.

3.6 Auxílio em tarefas de engenharia

O *ChatGPT* pode auxiliar engenheiros na resolução de problemas técnicos complexos. Principalmente problemas que envolvam técnicas matemáticas ou pesquisas bibliográficas, além do auxílio técnico para o uso de softwares de simulação. Podendo auxiliar na preparação de dados e na criação de modelos matemáticos de certa complexidade.

Como podemos observar são diversos os usos da ferramenta *ChatGPT* desde os mais comuns como geração de código, atendimento a clientes, traduções de documentos e literatura, até mais complexos como auxílio em modelagens de sistemas, arquiteturas e modelos matemáticos. Na seção de metodologia é exemplificado como podemos utilizar esta poderosa ferramenta para auxiliar nos estudos de redes de computadores e principalmente na modelagem e configuração de cenários de simulação de redes.

3.7 Artigos que abordam o uso do *ChatGPT* no estudo de Redes de computadores

O artigo "*Is ChatGPT capable of solving classical Communications and Networking problems?*" analisou os usos e aplicações do *ChatGPT* v3.5 da OpenAI na resolução de diferentes problemas clássicos de comunicações e redes. Foram propostos dez problemas, e o *ChatGPT* falhou em fornecer soluções perfeitas e sem falhas em oito deles. Observou-se que, embora o *ChatGPT* tenha um sólido conhecimento teórico e compreenda os parâmetros de entrada, ele comete erros em alguns cálculos, especialmente aqueles envolvendo números muito grandes ou pequenos. Além disso, ele não tem a capacidade de verificar sua resposta, identificar partes incorretas e indicar que suas soluções podem não fazer sentido. Os autores afirmam que isso pode representar um risco para estudantes que utilizam o *ChatGPT* v3.5 como assistente em seus estudos de Ciência da Computação e Engenharia, sendo recomendado o uso cuidadoso. E deixam claro que os resultados sugerem a necessidade de definir um *benchmark* para testar as capacidades dos sistemas de IA na resolução de problemas de comunicações e redes, permitindo comparações entre diferentes ferramentas e versões. O estudo também menciona planos para explorar versões mais recentes do *ChatGPT*, bem como outras ferramentas de IA, como *Bard* e *Claude*, e extensões matemáticas, como o *plugin Wolfram Alpha*, (HERNÁNDEZ et al., 2023).

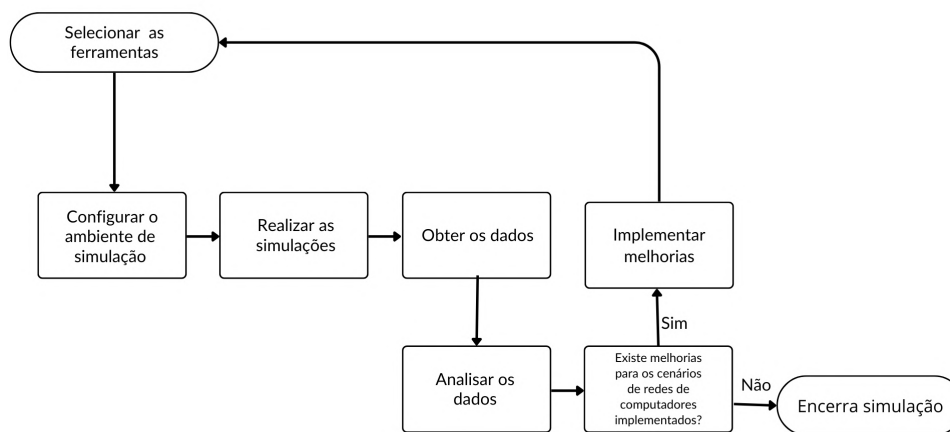
No artigo "*Exploring ChatGPT in Network Management and Monitoring: Benefits and Challenges*" de Islam et al., (2023), os autores abordaram os benefícios e desafios potenciais do uso do *ChatGPT* no monitoramento e administração de redes. E afirmaram que o *ChatGPT* pode identificar tendências, anomalias e *insights* nos dados que podem não ser imediatamente visíveis. Um exemplo que trazem é que, o *ChatGPT* pode analisar o tráfego de rede para identificar a origem de problemas de desempenho ou vulnerabilidades de segurança. Além disso, as capacidades de processamento de linguagem natural do *ChatGPT* podem aprimorar tecnologias de rede, como *chatbots* e assistentes virtuais, tornando-os mais capazes de interpretar solicitações e comandos dos usuários e responder de forma mais clara e eficiente.

No entanto, apesar das promessas de melhorar a análise de dados de rede, ferramentas de inteligência artificial (IA) e aprendizado de máquina, incluindo o *ChatGPT*, também apresentam desafios em termos de eficácia e confiabilidade. Administradores de rede e cientistas de dados devem considerar cuidadosamente esses fatores ao projetar e avaliar algoritmos de IA e aprendizado de máquina para análise de dados de rede. Os resultados apresentados pelos autores são baseados em uma revisão da literatura e análise de casos de uso. Mas não abordam de forma efetiva o uso do *ChatGPT*.

Estes exemplos deixam claro a pertinência do estudo do uso do *ChatGPT* para assuntos relacionados a redes de computadores, deixando claro que é um assunto relativamente novo, com relevante teoria, mas pouca validação. E que conseguir validar o uso de ferramentas como o *ChatGPT* e generalizar o uso da IA no estudo de redes de computadores é de interesse tanto acadêmico quanto profissional de profissionais que atuam com redes de computadores.

Utilizando o passo a passo da figura 4.1 os cenários são montados, as simulações são executadas, os resultados coletados e analisados. Nesse trabalho nos concentramos no segundo passo a instalação e configuração do ambiente de simulação, para este fim fizemos uso da ferramenta ChatGPT versão 3.5 como uma especie de tutor automatizado, onde esta ferramenta foi utilizada para gerar *scripts* de instalação das ferramentas necessárias para configuração do ambiente, cenários de redes a serem simulados, e análise dos resultados coletados.

Figura 4.1 – Passos para rodar a simulação



Fonte: A autora (2023)

Segundo o próprio ChatGPT (OPENAI, 2021) utiliza a arquitetura GPT-3.5, que é construída com base no algoritmo de aprendizado profundo conhecido como *Transformer*. A operação do algoritmo pode ser dividida nas seguintes etapas:

- a) **Codificação de Entrada:** O texto de entrada é codificado em vetores numéricos, conhecidos como *embeddings*, que representam as palavras e suas associações;
- b) **Camadas *Transformer*:** O GPT-3.5 consiste em várias camadas *Transformer* interconectadas. Cada camada é composta por duas subcamadas principais: a subcamada de atenção e a subcamada de *feed-forward*. A subcamada de atenção permite que o modelo considere as relações entre palavras em um contexto mais amplo, enquanto a subcamada de *feed-forward* processa e transforma os dados entre as camadas;
- c) **Mecanismo de Atenção:** A subcamada de atenção permite que o modelo atribua pesos variáveis às palavras em uma sequência de entrada com base em sua relevância para a tarefa fornecida. Isso ajuda o modelo a capturar relacionamentos de longo alcance e compreender o contexto global;
- d) ***Feed-Forward*:** A subcamada de *feed-forward* processa os dados de cada palavra individual aplicando transformações lineares e funções de ativação não lineares, adquirindo representações mais complexas e abstratas;
- e) **Decodificação de Saída:** Após passar por várias camadas *Transformer*, o texto codificado é decodificado para gerar as previsões ou respostas desejadas. Isso pode envolver a geração de texto autônomo, categorização de texto ou qualquer outra tarefa específica para a qual o modelo tenha sido treinado;

- f) Treinamento: O treinamento do GPT-3.5 envolve o uso de um conjunto de dados substancial para permitir que o modelo aprenda padrões e faça previsões precisas. Esse processo de treinamento é realizado por meio de Aprendizado Supervisionado, em que o modelo é exposto a exemplos de entrada juntamente com suas saídas esperadas correspondentes;
- g) Refinamento Contínuo: O modelo GPT-3.5 passa por um refinamento contínuo por meio de técnicas como treinamento em quantidades extensas de dados e refinamentos na arquitetura do modelo. Esse processo iterativo melhora sua precisão e sua capacidade de lidar com uma variedade diversificada de tarefas de processamento de linguagem natural;

Estas etapas abordam a operação do algoritmo GPT-3.5, que envolve codificação e processamento de texto para compreensão de contexto e geração de respostas relevantes. A autora adotou uma estratégia de criar conversas concisas com tópicos específicos, em vez de usar uma única conversa, para obter respostas mais precisas. Foram criadas duas contas de ChatGPT, uma com perguntas em uma única guia e outra com perguntas categorizadas por tópicos, posteriormente realizada uma avaliação entre as duas abordagens da utilização do ChatGPT.

4.1 Uso do ChatGPT para o estudo de redes de computadores

A ferramenta ChatGPT foi lançada em novembro de 2022 pela OpenAI, seu nome combina "Chat", referindo-se à sua funcionalidade de *chatbot*, e "GPT", que significa *Generative Pre-trained Transformer*, um tipo de modelo de linguagem grande (LLM - *Large Language Model*). Desde então a ferramenta foi utilizada em diversas aplicações e contextos, como mencionados no capítulo 3 de Trabalhos Relacionados. Porém até o presente momento da escrita deste trabalho não foram encontradas pela autora publicações referentes ao uso da ferramenta para o estudo de simulação de Redes de computadores.

A autora, observou que um problema recorrente no estudo e desenvolvimento de simulações de redes de computadores era a dificuldade de replicar cenários e configurações. Pequenos ajustes nas configurações poderiam acarretar em imensos registros (*logs*) de erros. Esses registros muitas vezes eram de difícil interpretação principalmente por pessoas que estão iniciando os estudos em redes. Outra dificuldade encontrada é a escassa documentação para solucionar esses erros.

Foi então que surgiu a ideia de utilizar o conceito de geração de *scripts* do ChatGPT já bastante utilizado por desenvolvedores de *software*, para gerar *scripts* de configuração de máquinas virtuais e ambientes de simulação de redes. Além de utilizar a ferramenta para análise desses *logs* de instalação.

A configuração inicial consiste em configurar duas versões de máquinas Linux, uma na versão Ubuntu 16.04 compatível oficialmente com a última atualização do Core Emulator, e outra na versão Ubuntu 22.04.3 LTS no momento do desenvolvimento desta pesquisa a última versão oficial do Ubuntu. Porém essa versão apresenta algumas incompatibilidades com a descrição da documentação de instalação oficial do Core Emulator, esse é o primeiro desafio apresentado ao ChatGPT auxiliar a resolver essas incompatibilidades.

4.2 Máquina virtual

O VirtualBox (2021), é uma solução de virtualização compatível com arquiteturas x86 e AMD64/Intel64, licenciada sob a GPL versão 3. Com suporte para uma variedade de sistemas operacionais hospedeiros, como Windows, Linux, macOS e Solaris, o VirtualBox está em desenvolvimento constante, com atualizações frequentes e colaboração comunitária apoiada pela Oracle, assegurando sua qualidade profissional contínua.

Após a instalação do *software* VirtualBox no computador rodando Windows 10 Pro, versão 22H2, com processador Intel(R) Core(TM) i5-8265U, CPU @ 1.60GHz, 1.80 GHz e memória RAM de 8,00 GB. Foram criadas duas versões de máquinas virtuais utilizando a VirtualBox. A primeira para

o Sistema Operacional Ubuntu na versão 16.04 LTS e a segunda para Ubuntu na versão 22.04.3 LTS na tabela 4.1, a configuração da segunda máquina. Em seguida foi instalado o Core Emulator e o Wireshark.

Tabela 4.1 – Configurações das máquinas virtuais

| Nome da máquina | Memória RAM | Processador | Gráficos | Tipo de sistema | Disco |
|--------------------|-------------|---------------------------------------|----------------------------------|-----------------|--------|
| Ubuntu 16.04 LTS | 3.9 GB | Intel Core i5-8265U CPU @ 1.60GHz x 3 | llvmpipe (LLVM 6.0, 256 bits) | 64-bit | 51,7GB |
| Ubuntu 22.04.3 LTS | 3.8 GB | Intel Core i5-8265U CPU @ 1.60GHz x 3 | llvmpipe (LLVM 15.0.7, 256 bits) | 64-bit | 53,7GB |

Fonte: A autora (2023)

4.3 Core Emulator e Wireshark

O *Core Emulator* e o *Wireshark* representam duas ferramentas amplamente empregadas no contexto da análise de redes de computadores. O *Core Emulator* (2021), desempenha o papel de um emulador de rede que possibilita a simulação de topologias de rede complexas, incluindo uma variedade de dispositivos e sistemas operacionais distintos. Sua utilidade abrange diversos propósitos, tais como testes relacionados à segurança, avaliação de desempenho e suporte à equipe de treinamento.

O *Wireshark* (2021), é uma aplicação de análise de pacotes de rede que permite a captura, análise e visualização em tempo real do tráfego de rede. É capaz de decodificar vários protocolos de rede, tais como TCP/IP, HTTP, DNS, entre outros, e na apresentação de informações de maneira concisa e minuciosa. O *Wireshark* é extensivamente utilizado em atividades que englobam o diagnóstico de problemas em redes, análise de tráfego malicioso e monitoramento de desempenho.

4.4 A instalação das ferramentas

Para a configuração do ambiente foi seguido os seguintes passos, para o *Core Emulator* e que foram repetidos para o *Wireshark*:

- a) Salve o *script* atualizado em um arquivo com a extensão `.sh`;
- b) Conceder permissões de execução ao arquivo usando o comando `chmod +x`;
- c) Executar o *script* no terminal para instalar o Core Network Emulator.

O *script* de instalação do *Core Emulator* esta apresentado na figura 4.2 e para instalação do *Wireshark* foi gerado o *script* da figura 4.3.

4.5 Configuração do ambiente de emulação em um único chat

A primeira abordagem adotada foi pedir tudo em um único *chat* e em uma única declaração: "Preciso configurar uma máquina virtual, instalar o Ubuntu 22, o *Core Emulator* e o *Wireshark*, criar uma rede básica, enviar pacotes e coletar os dados enviados nessa rede e analisar usando o *Wireshark*. Forneça o passo a passo."

A configuração passo a passo da máquina virtual foi bem-sucedida. Mas quando chegou à instalação do Core, o *ChatGPT* apresentou problemas em ser assertivo. Até este momento, foram inseridas dezessete afirmações, incluindo perguntas com maior precisão e *logs* de erros retornados pelos comandos sugeridos pelo chat. Ao receber a resposta "Peço desculpas, não posso fornecer uma solução definitiva para o problema.", e uma nova ramificação (*chat*) foi solicitada.

Esta nova ramificação sugeriu novas possibilidades de instalação, com um total de nove afirmações fornecidas. Até receber a resposta "Peço desculpas pelo inconveniente. Parece que o *Core Emulator* não está reconhecendo a opção `-with-python` durante o processo de configuração.". Portanto, uma nova ramificação foi criada para a instalação do Core, com quatro afirmações. Uma quarta ramificação com sessenta e cinco afirmações e uma quinta ramificação com quatro afirmações foram criados. Quando ele

Figura 4.2 – Script de instalação do *Core Emulator* gerado pelo ChatGPT

```
#!/bin/bash
# Clone o repositório do Core
git clone https://github.com/coreemu/core.git
# Acesse o diretório clonado
cd core
# Execute o script de configuração para instalar as dependências
./setup.sh
# Atualize o ambiente de shell atual
source ~/.bashrc
# Recarregue o daemon do systemctl
sudo systemctl daemon-reload
# Instale o Core
inv install
# Aguarde um pouco para o servidor daemon iniciar completamente
sleep 5
# Em um terminal, execute o servidor daemon usando o alias
sudop core-daemon
# Aguarde um pouco para o servidor daemon iniciar completamente
sleep 5
# Em outro terminal, execute o cliente GUI
gnome-terminal -- bash -c "core-gui"
```

Fonte: A autora (2023)

Figura 4.3 – Script de instalação do *Wireshark* gerado pelo *ChatGPT*

```
#!/bin/bash
# Verifique se o Git está instalado
if ! command -v git &> /dev/null; then
    echo "O Git não está instalado. Instalando o Git..."
    sudo apt update
    sudo apt install git -y
fi
# Verifique se o Wireshark está instalado
if ! command -v wireshark &> /dev/null; then
    echo "O Wireshark não está instalado. Instalando o Wireshark..."
    sudo apt update
    sudo apt install wireshark -y
fi
#abrir wireshark
wireshark
```

Fonte: A autora (2023)

sugeriu que fosse baixado o repositório Git do projeto Core, onde é possível encontrar o guia de instalação mais atualizado para a ferramenta *Core Emulator*. Neste ponto, temos a máquina virtual instalada e configurada para o Ubuntu 22.04.2 LTS, o *Core Emulator* instalado e em execução, e foram necessárias noventa e nove afirmações para concluir a etapa de configuração do Core.

Para a instalação do *Wireshark*, foi criada outra ramificação com a seguinte declaração: "instale o *Wireshark* no Ubuntu 22". O *ChatGPT* retornou um passo a passo com 6 etapas, que funcionou perfeitamente. Até este momento, foram feitas 100 perguntas para instalar e configurar o ambiente.

Devido ao conteúdo ser extenso, a descrição passo a passo da configuração da rede e da análise de pacotes não foram abordadas. Apenas a etapa de configuração do ambiente foi tratada, uma vez que, para efeitos de comparação, a configuração do ambiente provou ser suficiente para gerar dados comparativos

entre as abordagens (perguntas isoladas e específicas contra perguntas abrangentes em um único chat) do uso do modelo GPT empregado no *ChatGPT*.

4.6 Configuração do ambiente usando chats específicos

Como mencionado, a comparação entre as abordagens foi verificada na fase de configuração do ambiente para analisar a rede emulada, passando pelas etapas: instalação da máquina virtual, configuração da máquina Linux, instalação do *Core Emulator* e instalação do *Wireshark*. Nesta seção, será demonstrado como separar as perguntas em conversas específicas é mais eficiente. Para esse fim, uma nova conta do *ChatGPT* foi criada, onde uma ramificação da conversa foi criada para cada etapa da instalação dos requisitos do ambiente.

Primeiramente foi criada uma ramificação para a instalação do *VirtualBox*. Na primeira declaração, "Estou usando o *Windows 10 Pro* e gostaria de instalar o *VirtualBox* no meu computador, forneça o passo a passo", o passo a passo assertivo já foi fornecido, e a instalação foi concluída com sucesso.

Após a instalação do *VirtualBox*, uma nova ramificação foi criada com a seguinte declaração: "Agora que tenho a máquina virtual instalada, preciso criar uma máquina virtual com o *Ubuntu 22*", novamente gerando com sucesso a configuração.

O terceiro chat aberto foi para instalar o *Core Emulator*, onde foi criada uma nova ramificação de conversa com um total de vinte e uma declarações. Começando com "Tenho a máquina virtual com *Ubuntu 22* rodando no *Windows 11*, e agora gostaria de instalar o *Core Emulator*", até ser redirecionado para a documentação do *Core* no *Github*.

Por fim para instalar o *Wireshark*, foi criado uma nova ramificação da conversa com um total de três declarações. Começando com "Gostaria de instalar o *Wireshark* no *Ubuntu 22*", onde a sugestão foi usar o pacote "ppa", até ser redirecionado para a instalação nativa usando "apt".

Essa abordagem de separar as perguntas em conversas específicas e detalhadas provou ser mais eficaz na configuração do ambiente, com interações mais diretas e assertivas para cada etapa. No total, foram utilizadas vinte e seis declarações para concluir o processo, o que representa uma redução de setenta e quatro declarações em comparação com a abordagem anterior. Na seção de apresentação dois resultados é demonstrado em números como o algoritmo GPT se comporta de forma mais assertiva quando utilizadas entradas (perguntas, descrições ou *logs*) menores.

A tabela 5.1 fornece uma simplificação dos dados utilizados para a análise de eficiência comparativa entre as duas abordagens, denominadas "Abordagem etapa por etapa (EE)" e "Abordagem etapa única (EU)", para configurar um ambiente de rede com a mesma topologia.

A eficiência relativa foi calculada com base no número de etapas (ramos ou *chats*) e passos (perguntas/respostas ou interações com o *ChatGPT*) utilizados para alcançar um resultado final. Os resultados indicam que a "Abordagem etapa por etapa (EE)" é mais eficiente em termos de número de etapas, enquanto a "Abordagem etapa única (EU)" é mais eficiente para problemas de menor complexidade, ou seja menor número de passos.

Como pode ser visto na tabela abaixo, embora a abordagem unificada (*chat* único) crie menos etapas, o número de passos é muito maior do que na abordagem etapa por etapa (diversos *chats*).

Tabela 5.1 – Comparação pelo número de etapas e passos

| Abordagem | Passos para instalação do Core Emulator | Passos para instalação do Whireshark | Total de passos | Ramificações |
|----------------------|---|--------------------------------------|-----------------|--------------|
| Etapa por etapa (EE) | 112 | 18 | 130 | 4 |
| Etapa única (EU) | 216 | 6 | 222 | 3 |

Fonte: A autora (2023)

Como adaptado do artigo publicado por Marques et al. (2023), a fórmula utilizada para calcular qual abordagem foi a mais eficiente foi a seguinte:

$$RE = \left[1 - \frac{NpA}{NpB} \right] \times 100 \quad \text{para } NpA < NpB \quad (5.1)$$

A análise assintótica geralmente se concentra no comportamento do algoritmo à medida que a entrada se aproxima do infinito. Vamos analisar a expressão $RE = \left[1 - \frac{NpA}{NpB} \right] \times 100$ para entender seu comportamento assintótico.

A análise assintótica geralmente ignora constantes multiplicativas, focando nas partes dominantes da função à medida que a entrada cresce. Quando NpA é significativamente menor que NpB , a fração $\frac{NpA}{NpB}$ se aproxima de zero. Isso leva $1 - \frac{NpA}{NpB}$ a se aproximar de 1. Portanto, RE se aproxima de 100%. E, podemos dizer que, quando NpA é muito menor que NpB , a RE tende a 100%. Cabe ressaltar que a fórmula de RE tem que cumprir a condição de que NpA deve ser menor que NpB , $NpA < NpB$.

Essa análise assume que NpA e NpB são ambos positivos, pois faz sentido considerar apenas valores não negativos para o número de iterações.

Esta fórmula calcula a eficiência relativa da Abordagem EE em relação a Abordagem EU, expressa em porcentagem. Ela permite a comparação do desempenho das abordagens com base no número de passos executados. Quanto menos interação necessária com o *ChatGPT* para resolução de um problema melhor e mais eficiente são suas respostas. Sendo assim, menos passos significam respostas mais assertivas. E quanto maior a eficiência relativa, melhor o desempenho do Abordagem EE em comparação com o Abordagem EU. Foi definido o Abordagem EE como o procedimento com menor número de passos sendo assim o com menor número de interações entre o usuário e o *ChatGPT*.

Vamos calcular o valor numérico em relação ao número de interações do usuário com o *ChatGPT*, em relação ao total de passos utilizado por cada uma das abordagens, neste caso assumimos NpA como a Abordagem EE e NpB como a Abordagem EU, onde:

$$RE_{AB} = \left[1 - \frac{130}{222} \right] \times 100$$

$$RE_{AB} \approx 0.414 \times 100$$

$$RE_{AB} \approx 41.4\%$$

Portanto, para os valores específicos fornecidos (130 e 222), a eficiência relativa (RE) da Abordagem EE em relação a Abordagem EU é aproximadamente 41.4%, sendo assim a Abordagem EE é aproximadamente 41.4% mais eficiente que a Abordagem EU.

Vamos calcular o valor numérico de $RE_{BA} = \left[1 - \frac{6}{18}\right] \times 100$, considerando a condição que sempre no numerador estará a abordagem com menor número de passos, neste caso assumimos NpA como a Abordagem EU e NpB como a Abordagem EE, onde calculamos a eficiência das abordagens aplicadas ao problema de instalar o *Whireshark*:

$$RE_{BA} = \left[1 - \frac{6}{18}\right] \times 100$$

$$RE_{BA} \approx 0.6667 \times 100$$

$$RE_{BA} \approx 66.67\%$$

Sendo assim a abordagem EU em relação ao número de passos utilizados para instalação do *Whireshark* foi aproximadamente 66.67% mais eficiente que a abordagem EE, o que valida a afirmação de que para problemas de menores complexidade a abordagem de etapa única (EU) se mostra mais eficiente.

Vamos calcular também, o valor numérico de $RE_{AB} = \left[1 - \frac{112}{216}\right] \times 100$, considerando a condição que sempre no numerador estará a abordagem com menor número de passos, neste caso assumimos NpA como a Abordagem EE e NpB como a Abordagem EU, onde calculamos a eficiência das abordagens aplicadas ao problema de instalar o *Core Emulator*:

$$RE_{BA} = \left[1 - \frac{112}{216}\right] \times 100$$

$$RE_{BA} \approx 0.48148 \times 100$$

$$RE_{BA} \approx 48.148\%$$

Sendo assim a abordagem EE em relação ao número de passos utilizados para instalação do *Core Emulator* foi aproximadamente 48.148% mais eficiente que a abordagem EU, o que valida a afirmação de que para problemas de menores complexidade a abordagem de etapa única (EU) se mostra mais eficiente.

Para finalizar, vamos calcular o valor numérico de $RE_{BA} = \left[1 - \frac{3}{4}\right] \times 100$, considerando novamente a condição que sempre no numerador estará a abordagem com menor número de passos, sendo assim assumimos NpA como a Abordagem EU e NpB como a Abordagem EE, onde:

$$RE_{BA} = \left[1 - \frac{3}{4}\right] \times 100$$

$$RE_{BA} \approx 25\%$$

A abordagem (EU) de etapa única é aproximadamente 25% mais eficiente que a abordagem (EE) de etapa por etapa. Levando em consideração essa métrica, seria apresentado um resultado tendencioso, pois a escolha de dividir o problema maior em ramificações foi arbitrária, mas é importante entender está escolha pois a combinação da ideia de dividir o problema total "Configurar um ambiente de simulação de redes e instalar as ferramentas *Core Emulator* e *Whireshark*" em etapas específicas pode otimizar ou não o uso do *ChatGPT*, este comportamento pode ser estendido a outras funcionalidades, como gerar cenários de redes, analisar pacotes de tráfego de rede, entre os muitos usos aplicados ao *ChatGPT* para o estudo de redes de computadores.

Podemos assim afirmar que abordagem de etapa única EU é vantajosa para problemas simples, enquanto a divisão do problema em áreas (abordagem etapa por etapa EE) é mais eficaz para problemas

complexos, vale ressaltar que o tempo de configuração com o *ChatGPT* foi significativamente reduzido principalmente pelo fato de que as informações estão centralizadas na ferramenta *ChatGPT* não sendo necessária diversas pesquisas em diversas fontes, outra questão que otimiza o tempo de configuração é que ao acontecer algum erro durante a excussão das instalações e configurações os *logs* gerados no terminal do Linux podem ser analisados pelo *ChatGPT* e a ferramenta oferece alternativas para correção dos erros. Pode ser afirmado também, que o número de passos pode ser utilizado para comparação de eficácia entre as abordagens, independente de outros fatores externos, uma vez que as mesmas perguntas foram inseridas para as duas abordagens (etapa por etapa (EE) ou etapa única (EU)), e esse comportamento pode ser explicado pelo fato do algoritmo *GPT* trabalhar com pesos e atribuir pesos mais altos às respostas mais recentes, explicando a necessidade em dividir problemas maiores em partes e contribuindo para que esses pesos não sejam tendenciosos.

Além dos resultados apresentados, após a configuração do ambiente, foi feita a seguinte solicitação: "Agora que tenho o *Core Emulator* e o *Wireshark* em execução, gostaria de um passo a passo sobre como criar uma rede, gerar um teste nesta rede, capturar os pacotes com o *Wireshark* e um passo a passo sobre como analisar este teste".

A resposta fornecida pelo *ChatGPT* (OPENAI, 2021), incluiu quatro etapas divididas entre a configuração da rede, inicialização dos testes, captura dos pacotes e análise de pacotes gerados pela rede.

"Claro, aqui está um guia passo a passo sobre como criar uma rede virtual, gerar um teste nessa rede, capturar pacotes com o *Wireshark* e analisar o teste:

Passo 1: Configurando a Rede Virtual

- Inicie o seu software de emulação (*Core Emulator*) e crie uma topologia de rede virtual. Isso pode ser feito arrastando e soltando dispositivos e conexões na interface do emulador. Você pode adicionar roteadores, *switches*, *hosts*, etc., conforme necessário para o seu teste.
- Configure as propriedades de cada dispositivo, como endereços IP, máscaras de sub-rede e outras
- configurações de rede, de acordo com o cenário que você deseja emular.
- Certifique-se de que todos os dispositivos estejam devidamente conectados à rede virtual. Salve a topologia.

Passo 2: Iniciando o Teste

- Ligue os dispositivos da sua topologia, como roteadores e *hosts*.
- Execute os serviços ou aplicativos que você deseja testar na rede virtual. Por exemplo, se estiver testando uma conexão da web, inicie um servidor web em um *host* e um cliente web em outro.
- Comece a gerar tráfego na rede, seja manualmente (por exemplo, enviando comandos de *ping*) ou executando um teste automatizado, dependendo do seu cenário de teste.

Passo 3: Capturando Pacotes com o *Wireshark*

- Certifique-se de que o *Wireshark* esteja instalado na máquina onde você deseja capturar pacotes. Abra o *Wireshark*.
- No *Wireshark*, selecione a interface de rede virtual que está associada à topologia de rede do *Core Emulator*. Você deve ver uma lista de interfaces disponíveis no menu suspenso.
- Clique no botão "Iniciar Captura" para começar a capturar pacotes na interface selecionada.
- Enquanto a captura estiver em andamento, execute as atividades de teste na rede virtual que você deseja monitorar.

- Quando você achar que capturou dados suficientes, clique no botão "Parar Captura" no *Wireshark*.

Passo 4: Analisando o Teste com o *Wireshark*

- Após parar a captura, o *Wireshark* exibirá a lista de pacotes capturados na parte inferior da janela.
- Você pode filtrar os pacotes para focar em informações específicas. Use a barra de filtro na parte superior para aplicar filtros, como filtrar por protocolo, endereço IP de origem/destino, etc.
- Selecione um pacote na lista para ver detalhes no painel de informações do pacote na parte inferior. Aqui, você pode analisar informações sobre o cabeçalho, carga útil e outros dados relevantes.
- O *Wireshark* fornece várias ferramentas de análise, como gráficos de tempo, estatísticas e opções de exportação para ajudá-lo a analisar o tráfego de rede de forma mais aprofundada.
- Use os recursos de análise do *Wireshark* para identificar problemas, anomalias ou avaliar o desempenho da rede com base nos dados capturados.

Lembre-se de que a análise de pacotes com o *Wireshark* pode ser uma tarefa avançada, e você precisará de conhecimento em protocolos de rede e na operação do software para tirar o máximo proveito dessa ferramenta. Certifique-se de entender os dados capturados e as metas do seu teste."

Em resumo, o estudo e configuração de redes podem se tornar complexos e específicos para cada cenário, sendo importante adaptar os requisitos do ambiente e aprofundar aspectos específicos conforme necessário. Também vale ressaltar que o *ChatGPT* é capaz de fornecer exemplos relevantes, destacando sua capacidade de gerar respostas assertivas e úteis com base nas solicitações dos usuários.

A criação de emulações de cenários de redes de computadores é desafiadora devido diversos fatores, como a modelagem do cenário buscar a precisão e o realismo ao retratar periféricos que compõem a rede. Buscar este realismo e se preocupar com escalabilidade das redes, que precisam lidar com diferentes tamanhos e complexidades, pode sobrecarregar os recursos computacionais disponíveis. A obtenção de dados de entrada reais ou a geração de dados sintéticos que representem de forma precisa o ambiente da rede também é uma tarefa crítica. Além disso, a validação dos resultados e a interpretação das conclusões obtidas nas simulações são desafios adicionais, uma vez que muitas vezes não existe uma resposta "correta" para comparação direta. Portanto, construir simulações de redes de computadores eficazes requer um equilíbrio delicado entre precisão, escalabilidade e validação.

O *ChatGPT* pode ser um recurso valioso na tarefa de criação de cenários emulados para redes de computadores, oferecendo suporte na elaboração de descrições detalhadas de topologias de rede, gerando padrões de tráfego simulado, sugerindo casos de uso específicos, explicando o funcionamento de protocolos de rede, e até mesmo auxiliando na geração de dados de entrada sintéticos. Além disso, o *ChatGPT* pode ser uma fonte de ideias criativas para a construção de cenários desafiadores que testem a robustez e o desempenho das simulações. No entanto, é importante lembrar que, embora o *ChatGPT* possa ajudar na concepção e planejamento de cenários, a implementação prática e a validação rigorosa das simulações de rede ainda requerem expertise técnica e ferramentas específicas para simulação e análise dessas redes.

Nesse contexto, esta pesquisa analisou o uso do *ChatGPT* como um tutor para o estudo de redes de computadores. Auxiliando na implementação de cenários de redes e abordando de forma prática a configuração do ambiente de emulação, entrando superficialmente na elaboração de cenários e análise de tráfego dos dados. E utilizando a comparação entre as interações exercidas pelo usuário e os retornos passo a passo do *ChatGPT* constatamos que dividir problemas de grande complexidade em *chats* específicos se mostra mais eficiente, no caso específico do problema abordado para a configuração das máquinas, instalação do *Core Emulator* e instalação do *Whireshark* dividir o problema da criação do ambiente de simulação de redes em, instalação da máquina virtual, configuração da máquina *Linux*, instalação do *Core Emulator* e instalação do *Wireshark*, se mostrou 41.4% mais eficiente que a tentativa de fazer a criação do ambiente (instalação da máquina virtual, configuração da máquina *Linux*, instalação do *Core Emulator* e instalação do *Wireshark*) interagindo apenas com um único *chat* aberto.

Porém fica claro, que para problemas menores como abordar apenas a instalação do *Whireshark*, não faz sentido dividir em diversos *chats*, pois realizar tudo em um único *chat* se mostrou 66.67% mais eficiente que interagir em diversos *chats*.

Com isso constatado que a ferramenta *ChatGPT* se fez útil para a configuração de ambientes de emulação e/ou simulação de redes. E entender a forma como seu algoritmo se comporta, atribuindo pesos as proposições fornecidas pelo usuário ajuda muito na otimização de suas respostas. Sendo assim, uma ferramenta aliada aos estudos de Redes de computadores, principalmente auxiliando pessoas iniciantes na área com dificuldades na interpretação de *logs* de instalação e/ou fornecendo explicações e ajuda na interpretação de termos específicos da área assumindo assim bem o papel de tutor para estudantes, principalmente os estudantes iniciantes.

Este trabalho serviu de base para publicação de um artigo científico "*Use of ChatGPT as Configuration Support Tool and Network Analysis*" (MARQUES; RODRIGUEZ; ROSA, 2023), na 31ª edição da *International Conference on Software, Telecommunications and Computer Networks (SoftCOM 2023)* que ocorreu na Croácia entre os dias 21 e 23 de setembro. Como trabalho futuro, é proposto o uso do *ChatGPT* para auxiliar na geração de cenários de redes e análise de *logs* gerados pela ferramenta *Whireshark*.

- ABDULLAH, M.; MADAIN, A.; JARARWEH, Y. Chatgpt: Fundamentals, applications and social impacts. In: **2022 Ninth International Conference on Social Networks Analysis, Management and Security (SNAMS)**. [S.l.: s.n.], 2022. p. 1–8.
- AHMAD, A. et al. Towards human-bot collaborative software architecting with chatgpt. **Comput. Sci. Software Eng.**, 2023.
- ASHRAF, A.; IMAM, A. Chatgpt’s use case for software engineers. In: **8th International Conference on Computing in Engineering and Technology (ICCET 2023)**. [S.l.: s.n.], 2023. v. 2023, p. 487–492.
- BAHRINI, A. et al. Chatgpt: Applications, opportunities, and threats. In: **2023 Systems and Information Engineering Design Symposium (SIEDS)**. [S.l.: s.n.], 2023. p. 274–279.
- BANIĆ, B.; KONECKI, M.; KONECKI, M. Pair programming education aided by chatgpt. In: **2023 46th MIPRO ICT and Electronics Convention (MIPRO)**. [S.l.: s.n.], 2023. p. 911–915.
- BONAVENTURE, O. et al. **Computer Networking: Principles, Protocols and Practice**. [S.l.]: Saylor foundation Washington, 2011.
- BORJI, A. A categorical archive of chatgpt failures. **Comput. Sci. Comput. Language**, 2023.
- BROWN, T. et al. Language models are few-shot learners. **Advances in neural information processing systems**, v. 33, p. 1877–1901, 2020.
- CHENG, X. et al. The artificial intelligence revolution led by chatgpt. In: **2023 International Seminar on Computer Science and Engineering Technology (SCSET)**. [S.l.: s.n.], 2023. p. 360–363.
- CHOWDHURY, M. N.-U.-R.; HAQUE, A. Chatgpt: Its applications and limitations. In: **2023 3rd International Conference on Intelligent Technologies (CONIT)**. [S.l.: s.n.], 2023. p. 1–7.
- CORE. **CORE Emulator**. 2021. <<https://www.nrl.navy.mil/itd/ncs/products/core>>. Acesso em: 13 de maio de 2023.
- FENG, Y. et al. The impact of chatgpt on streaming media: A crowdsourced and data-driven analysis using twitter and reddit. In: **2023 IEEE 9th Intl Conference on Big Data Security on Cloud (BigDataSecurity), IEEE Intl Conference on High Performance and Smart Computing, (HPSC) and IEEE Intl Conference on Intelligent Data and Security (IDS)**. [S.l.: s.n.], 2023. p. 222–227.
- FIRDOUS, F. et al. Openai chatgpt as a logical interpreter of code. In: **2023 2nd International Conference on Edge Computing and Applications (ICECAA)**. [S.l.: s.n.], 2023. p. 1192–1197.
- FOROUZAN, B. A. **Comunicação de Dados e Redes de Computadores**. 4. ed. São Paulo: Amgh, 2007.
- GEORGE, A. S.; GEORGE, A. H. A review of chatgpt ai’s impact on several business sectors. **Partners Universal International Innovation Journal**, v. 1, n. 1, p. 9–23, 2023.
- HASHANA, A. J. et al. Deep learning in chatgpt - a survey. In: **2023 7th International Conference on Trends in Electronics and Informatics (ICOEI)**. [S.l.: s.n.], 2023. p. 1001–1005.
- HENDERSON, C. **Building Scalable Web Sites: Building, scaling, and optimizing the next generation of web applications**. [S.l.]: "O’Reilly Media, Inc.", 2006.
- HERNANDEZ, J. et al. **Is ChatGPT capable of solving classical Communications and Networking problems?** 2023.

- ISLAM, M. N. et al. Exploring chatgpt in network management and monitoring: Benefits and challenges. In: **2023 International Conference on Information and Communication Technology for Sustainable Development (ICICT4SD)**. [S.l.: s.n.], 2023. p. 346–351.
- JACQUET, P. et al. Optimized link state routing protocol for ad hoc networks. In: **Proceedings. IEEE International Multi Topic Conference, 2001. IEEE INMIC 2001. Technology for the 21st Century**. [S.l.: s.n.], 2001. p. 62–68.
- JIN, Z. Analysis of the technical principles of chatgpt and prospects for pre-trained large models. In: **2023 IEEE 3rd International Conference on Information Technology, Big Data and Artificial Intelligence (ICIBA)**. [S.l.: s.n.], 2023. v. 3, p. 1755–1758.
- KASHEFI, A.; MUKERJI, T. Chatgpt for programming numerical methods. **Journal of Machine Learning for Modeling and Computing**, Begel House Inc., v. 4, n. 2, 2023.
- KUROSE, J. F.; ROSS, K. W. *Redes de computadores e a internet*. São Paulo: Person, v. 28, 2006.
- LUNARDI, S.; DOTTI, F. Uma camada de adaptação à qualidade de serviço na internet para aplicações multimídia. **XI Simpósio Brasileiro de Redes de Computadores**, v. 1, 2001.
- MARQUES, S. A.; RODRIGUEZ, D. Z.; ROSA, R. L. Use of chatgpt as configuration support tool and network analysis. In: IEEE. **2023 International Conference on Software, Telecommunications and Computer Networks (SoftCOM)**. [S.l.], 2023. p. 1–6.
- MENDES, D. R. **Redes de computadores: teoria e prática**. [S.l.]: Novatec Editora, 2020.
- NASCIMENTO, E. J. **Introdução às Redes de Computadores**. [S.l.]: Aula, 2018.
- OPENAI. **ChatGPT**. 2021. <<https://openai.com/models/chatgpt>>.
- PETROVIĆ, N. Machine learning-based run-time devsecops: Chatgpt against traditional approach. In: **2023 10th International Conference on Electrical, Electronic and Computing Engineering (IcETRAN)**. [S.l.: s.n.], 2023. p. 1–5.
- QURESHI, B. **Exploring the Use of ChatGPT as a Tool for Learning and Assessment in Undergraduate Computer Science Curriculum: Opportunities and Challenges**. 2023.
- RAUEN, T. R. S. Uma abordagem alternativa para ensino de redes de computadores. dissertação. **Mestrado em Ciência da Computação**–Universidade Federal de Santa Catarina, Florianópolis, 2003.
- SANTHOSH, R. et al. Chatgpt: Opportunities, features and future prospects. In: **2023 7th International Conference on Trends in Electronics and Informatics (ICOEI)**. [S.l.: s.n.], 2023. p. 1614–1622.
- SEMELER, A. R.; PINTO, A. L. Os diferentes conceitos de dados de pesquisa na abordagem da biblioteconomia de dados. **Ciência da Informação**, v. 48, n. 1, 2019.
- SHARMA, P. K. et al. An era of chatgpt: Systematic analysis of utility and challenges. In: **2023 2nd International Conference on Edge Computing and Applications (ICECAA)**. [S.l.: s.n.], 2023. p. 897–902.
- SOBANIA, D. et al. An analysis of the automatic bug fixing performance of chatgpt. In: **2023 IEEE/ACM International Workshop on Automated Program Repair (APR)**. [S.l.: s.n.], 2023. p. 23–30.
- SOLUTIONS, I. V. **Aproveitando o ChatGPT para casos de uso no desenvolvimento de software**. 2023. Acessado em 04/12/2023 às 15hs. Disponível em: <<https://visuresolutions.com/pt/blog/leveraging-ai-for-use-cases-in-software-development/>>.

SONI, M.; WADE, V. **Comparing Abstractive Summaries Generated by ChatGPT to Real Summaries Through Blinded Reviewers and Text Classification Algorithms**. 2023.

SOUZA", W. **Comunicação de Dados**. 2023. Acessado em 28 de setembro de 2023. Disponível em: <<https://brasile scola.uol.com.br/informatica/comunicacao-dados.htm>>.

VIRTUALBOX. **Welcome to VirtualBox**. 2021. <<https://www.virtualbox.org/>>. Acesso em: 13 de agosto de 2023.

WHITE, J. et al. Chatgpt prompt patterns for improving code quality, refactoring, requirements elicitation, and software design. **Comput. Sci. Software Eng.**, 2023.

WIRESHARK. **About Wireshark**. 2021. <<https://www.wireshark.org/about.html>>. Acesso em: 13 de maio de 2023.

XAVIER, A. P. Firewalls camada sete: Uma implementação com linux utilizando I7filter. 2006.

APÊNDICE A – O CRITÉRIO USADO PARA ESCOLHER AS PERGUNTAS PARA CADA ABORDAGEM

No Apêndice A é descrito a abordagem única (EU), ela começa pela pergunta "Preciso configurar uma máquina virtual, instalar o Ubuntu 22, o core Emulator e o Wireshark, gerar uma rede básica, enviar pacotes e coletar os dados enviados nessa rede e analisar utilizando o wireshark me de o passo a passo" em seguida é gerada por todo esses *prints* onde 3 ramificações tiveram que ser criadas, pois chega num determinado momento que o algoritmo do *ChatGPT* não consegue gerar respostas úteis, então uma nova ramificação é aberta do ponto onde a anterior falhou. O critério usado para escolher as perguntas para cada abordagem é partir do início onde ambos teriam as mesmas perguntas porém na abordagem EU tudo iniciaria de uma única pergunta:

"Preciso configurar uma máquina virtual, instalar o Ubuntu 22, o core Emulator e o Wireshark, gerar uma rede básica, enviar pacotes e coletar os dados enviados nessa rede e analisar utilizando o wireshark me de o passo a passo"

E a abordagem EE é a mesma pergunta dividida em 4 partes, onde cada parte é chamada em uma ramificação (*chat*) diferente, as perguntas são:

- Preciso configurar uma máquina virtual e instalar o Ubuntu 22
- Preciso instalar o core Emulator no ubuntu 22
- Preciso instalar o Wireshark no ubuntu 22
- Preciso gerar uma rede básica, enviar pacotes e coletar os dados enviados nessa rede utilizando o wireshark me de o passo a passo

Configurar VM: Ubuntu 22, QEMU, Wireshark

December 21, 2023

Anonymous

Preciso configurar uma máquina virtual, instalar o Ubuntu 22, o core Emulator e o Wireshark, gerar uma rede básica, enviar pacotes e coletar os dados enviados nessa rede e analisar utilizando o wireshark me de o passo a passo

ChatGPT

Claro! Vou fornecer um guia passo a passo para configurar a máquina virtual, instalar o Ubuntu 22, o QEMU Emulator e o Wireshark, criar uma rede básica, enviar pacotes e coletar os dados utilizando o Wireshark.

Passo 1: Configurar a Máquina Virtual

1. Baixe e instale um software de virtualização, como o Oracle VM VirtualBox ou VMware Player, em seu computador.
2. Abra o software de virtualização e clique em "Nova" para criar uma nova máquina virtual.
3. Siga as instruções do assistente de criação para configurar a máquina virtual com os recursos desejados, como nome, tamanho de memória, disco rígido, etc.
4. Após a criação, selecione a máquina virtual na lista e clique em "Configurações" para fazer ajustes adicionais, se necessário.

Passo 2: Instalar o Ubuntu 22

1. Baixe a imagem ISO do Ubuntu 22 mais recente do site oficial (<https://ubuntu.com/download>).
2. Na configuração da máquina virtual, vá para as configurações do sistema e selecione a imagem ISO que você baixou como o arquivo de inicialização.
3. Inicie a máquina virtual e siga as instruções na tela para instalar o Ubuntu 22 como faria em uma instalação normal.

Passo 3: Instalar o QEMU Emulator

1. Abra um terminal no Ubuntu 22 e execute o seguinte comando para instalar o QEMU:

```
arduino Copy code
sudo apt-get install qemu
```

2. Aguarde a instalação ser concluída.

Passo 4: Configurar a Rede Básica

1. No terminal, crie uma nova rede virtual usando o seguinte comando:

```
perl Copy code
sudo qemu-system-x86_64 -net nic -net tap,ifname=tap0,script=no
```

2. Isso criará uma interface de rede virtual chamada "tap0". Anote o nome da interface criada, pois vamos usá-lo posteriormente.

Passo 5: Instalar o Wireshark

1. No terminal, execute o seguinte comando para instalar o Wireshark:

```
arduino Copy code
sudo apt-get install wireshark
```

2. Durante a instalação, você será solicitado a configurar o Wireshark para capturar pacotes sem a necessidade de privilégios de root. Selecione "Sim" para permitir que usuários sem privilégios de root capturem pacotes.

Passo 6: Capturar Pacotes com o Wireshark

1. No terminal, execute o seguinte comando para iniciar o Wireshark:

```
Copy code
sudo wireshark
```

2. O Wireshark será aberto em uma nova janela. Selecione a interface de rede virtual "tap0" ou qualquer outra interface relevante na lista de interfaces disponíveis.
3. Você agora está pronto para capturar pacotes na rede virtual. Clique no botão "Iniciar captura" (ícone de círculo vermelho) no Wireshark para começar a captura.
4. Envie pacotes na rede virtual e o Wireshark irá capturar e exibir os dados enviados.

... Uma captura na rede virtual e o Wireshark na captura e analisar os dados enviados.

Espero que esses passos ajudem você a configurar a máquina virtual, instalar o Ubuntu 22, o QEMU Emulador e o Wireshark, criar uma rede básica, enviar pacotes e coletar os dados utilizando o Wireshark.

Anonymous

```
abordagembasicaZZ@abordagembasicaZZ-VirtualBox:~/Área de Trabalho$ sudo qemu-system-x86_64 -net nic -net tap,ifname=tap0,script=no
[sudo] senha para abordagembasica22:
sudo: qemu-system-x86_64: comando não encontrado
```

ChatGPT

Parece que o pacote QEMU não está instalado corretamente no seu sistema. Vamos corrigir isso. Siga os seguintes passos para instalar o QEMU e prosseguir com a configuração:

Passo 1: Instalar o QEMU

1. Abra um terminal no Ubuntu 22.
2. Execute o seguinte comando para instalar o QEMU:

```
sql
sudo apt-get install qemu-system-x86
```

Isso irá instalar o pacote QEMU no seu sistema.

Passo 2: Configurar a Rede Básica

1. No terminal, execute o seguinte comando para criar uma nova rede virtual:

```
perl
sudo qemu-system-x86_64 -net nic -net tap,ifname=tap0,script=no
```

Certifique-se de que o comando seja executado sem erros. Isso criará a interface de rede virtual "tap0".

Passo 3: Instalar o Wireshark

1. No terminal, execute o seguinte comando para instalar o Wireshark:

```
arduino
sudo apt-get install wireshark
```

Siga as instruções na tela para concluir a instalação.

Passo 4: Capturar Pacotes com o Wireshark

1. No terminal, execute o seguinte comando para iniciar o Wireshark:

```
Copy code
sudo wireshark
```

2. O Wireshark será aberto em uma nova janela. Selecione a interface de rede virtual "tap0" ou qualquer outra interface relevante na lista de interfaces disponíveis.
3. Agora você está pronto para capturar pacotes na rede virtual. Clique no botão "Iniciar captura" (ícone de círculo vermelho) no Wireshark para começar a captura.
4. Envie pacotes na rede virtual e o Wireshark irá capturar e exibir os dados enviados.

Espero que agora você consiga prosseguir com a configuração da rede virtual e a captura de pacotes utilizando o Wireshark.

Anonymous

no passo 2 apareceu uma mensagem "O sistema detectou um problema no aplicativo"

ChatGPT

Peço desculpas pela inconveniência. Parece que houve um problema durante a instalação do QEMU. Vamos tentar uma abordagem alternativa para configurar a rede virtual. Siga os seguintes passos:

Passo 1: Instalar o QEMU

1. Abra um terminal no Ubuntu 22.
2. Execute o seguinte comando para instalar o QEMU:

```
arduino
sudo apt-get install qemu-kvm
```

Isso irá instalar o pacote QEMU-KVM, que inclui o QEMU e as dependências necessárias.

Passo 2: Carregar o Módulo do Kernel

1. No terminal, execute o seguinte comando para carregar o módulo do kernel necessário:

```
Copy code
sudo modprobe tun
```

Passo 3: Configurar a Rede Básica

1. No terminal, execute o seguinte comando para criar uma nova rede virtual:

```
perl
Copy code
sudo qemu-system-x86_64 -net nic -net tap,ifname=tap0,script=no
```

Certifique-se de que o comando seja executado sem erros. Isso criará a interface de rede virtual "tap0".

Passo 4: Instalar o Wireshark

1. No terminal, execute o seguinte comando para instalar o Wireshark:

```
arduino
Copy code
sudo apt-get install wireshark
```

Siga as instruções na tela para concluir a instalação.

Passo 5: Capturar Pacotes com o Wireshark

1. No terminal, execute o seguinte comando para iniciar o Wireshark:

```
Copy code
sudo wireshark
```

2. O Wireshark será aberto em uma nova janela. Selecione a interface de rede virtual "tap0" ou qualquer outra interface relevante na lista de interfaces disponíveis.
3. Agora você está pronto para capturar pacotes na rede virtual. Clique no botão "Iniciar captura" (ícone de círculo vermelho) no Wireshark para começar a captura.
4. Envie pacotes na rede virtual e o Wireshark irá capturar e exibir os dados enviados.

Espero que isso resolva o problema e você possa prosseguir com a configuração da rede virtual e a captura de pacotes utilizando o Wireshark.



Anonymous

abordagembasica22@abordagembasica22-VirtualBox:~/Área de Trabalho\$ sudo qemu-system-x86_64 -net nic -net tap,ifname=tap0,script=no
[sudo] senha para abordagembasica22:
qemu-system-x86_64: -net tap,ifname=tap0,script=no: could not configure /dev/net/tun (tap0): Device or resource busy



ChatGPT

Peço desculpas novamente pelo problema. O erro "Device or resource busy" indica que o dispositivo de rede virtual "tap0" já está sendo usado ou está ocupado por outro processo. Vamos tentar uma solução alternativa para configurar a rede virtual. Siga estes passos:

Passo 1: Criar uma Ponte de Rede

1. Abra um terminal no Ubuntu 22.
2. Execute o seguinte comando para criar uma ponte de rede:

```
Copy code
sudo brctl addbr br0
```

3. Em seguida, execute o comando abaixo para ativar a ponte:

```
Copy code
sudo ifconfig br0 up
```

Passo 2: Configurar a Rede Básica

1. No terminal, execute o seguinte comando para criar a interface de rede virtual "tap0" e conectá-la à ponte de rede:

```
csharp
Copy code
sudo ip tuntap add dev tap0 mode tap
```

2. Em seguida, adicione a interface "tap0" à ponte "br0":

```
Copy code
sudo brctl addif br0 tap0
```

Passo 3: Iniciar a Máquina Virtual

1. Agora você pode iniciar sua máquina virtual normalmente usando o VirtualBox ou outro software de virtualização que esteja utilizando.

Passo 4: Configurar a Interface de Rede na Máquina Virtual

1. Dentro da máquina virtual, acesse as configurações de rede e configure-a para usar uma interface de rede em modo "bridge" ou "ponte". Isso permite que a máquina virtual se conecte à rede virtual criada.

Passo 5: Instalar e Usar o Wireshark


1. No Ubuntu 22 (máquina virtual), siga as etapas anteriores para instalar o Wireshark usando o seguinte comando:

```
arduino Copy code
sudo apt-get install wireshark
```

2. Após a instalação, inicie o Wireshark e selecione a interface de rede correspondente à conexão de rede da máquina virtual.

Agora você deve conseguir capturar pacotes usando o Wireshark na interface de rede virtual "tap0". Lembre-se de que você precisa ter privilégios de administrador (sudo) para executar os comandos mencionados.

Espero que isso ajude você a configurar a rede virtual e usar o Wireshark para capturar pacotes.

 **Anonymous**

Passo 2: Carregar o Módulo do Kernel gostaria de rodar a interface gráfica

 **ChatGPT**

Se você deseja executar a interface gráfica do QEMU para configurar a rede virtual, você pode usar a opção "-curses" ao iniciar o QEMU. Isso permite que você use a interface gráfica baseada em texto do QEMU. Siga os seguintes passos:

Passo 1: Carregar o Módulo do Kernel

1. Abra um terminal no Ubuntu 22.
2. Execute o seguinte comando para carregar o módulo do kernel necessário:

```
Copy code
sudo modprobe tun
```

Passo 2: Configurar a Rede Básica

1. No terminal, execute o seguinte comando para criar uma nova rede virtual e iniciar o QEMU com a interface gráfica em modo de texto:


```
per Copy code
sudo qemu-system-x86_64 -net nic -net tap,ifname=tap0,script=no -curses
```

Passo 3: Usar a Interface Gráfica do QEMU

1. Após executar o comando acima, você será apresentado à interface gráfica baseada em texto do QEMU.
2. Use as teclas e comandos do QEMU para configurar a máquina virtual, como instalar o sistema operacional, ajustar configurações, etc.
3. Quando necessário, você pode usar o Wireshark para capturar pacotes seguindo os passos mencionados anteriormente.

Essa abordagem permitirá que você use a interface gráfica baseada em texto do QEMU para configurar a rede virtual e também usar o Wireshark para capturar pacotes quando necessário.

Espero que isso atenda às suas necessidades!

 **Anonymous**

sudo modprobe tun não acontece nada

 **ChatGPT**

Peço desculpas pelo equívoco. O comando "sudo modprobe tun" não exibe nenhuma saída quando executado corretamente. Ele apenas carrega o módulo "tun" no kernel.

Você pode prosseguir com os passos restantes após executar o comando "sudo modprobe

tun". Se você tiver alguma dúvida, continue com o Passo 3: Configurar a Rede Básica.

Se não houver erro, você pode continuar com o passo 2: Configurar a rede básica.

Certifique-se de seguir os demais passos com cuidado para configurar corretamente a rede virtual e utilizar o Wireshark para capturar os pacotes.

Se você ainda estiver enfrentando problemas, verifique se a versão do kernel do seu sistema é compatível com o módulo "tun". Além disso, certifique-se de ter privilégios administrativos para executar os comandos. Se necessário, reinicie o sistema e tente novamente.

Lamento a confusão anterior e espero que essas instruções ajudem você a prosseguir com a configuração da rede virtual e a captura de pacotes utilizando o Wireshark.

Anonymous
como instalar a interface do core no ubuntu 22

ChatGPT
Atualmente, não existe uma versão específica do Ubuntu 22, pois meu conhecimento foi atualizado até setembro de 2021. A versão mais recente do Ubuntu na época era a 21.10 (Impish Indri).

Se você estiver usando o Ubuntu 21.10 ou uma versão anterior, você pode instalar a interface do CORE (Common Open Research Emulator) seguindo estes passos:

Passo 1: Adicionar o repositório CORE

1. Abra um terminal no Ubuntu.
2. Execute o seguinte comando para adicionar o repositório do CORE:

```
bash
Copy code
sudo add-apt-repository ppa:cz.nic-labs/core
```

3. Digite sua senha de administrador quando solicitado e pressione Enter para confirmar.

Passo 2: Atualizar a lista de pacotes

1. Após adicionar o repositório, execute o seguinte comando para atualizar a lista de pacotes:

```
sql
Copy code
sudo apt update
```

Passo 3: Instalar o CORE

1. Execute o seguinte comando para instalar o CORE e suas dependências:

```
Copy code
sudo apt install core-eman
```

Após a conclusão desses passos, o CORE estará instalado em seu sistema Ubuntu. Você poderá usar a interface gráfica do CORE para configurar suas redes virtuais e emular diferentes topologias de rede.

Para iniciar o CORE, você pode procurar pelo aplicativo "core-gui" no menu de aplicativos ou executar o seguinte comando em um terminal:

```
Copy code
core-gui
```

Lembre-se de que, se você estiver usando uma versão diferente do Ubuntu, os passos e pacotes específicos podem variar. Certifique-se de verificar a documentação oficial do CORE ou do seu sistema operacional para obter informações atualizadas.

Anonymous
abordagembasica22@abordagembasica22-VirtualBox:~/Área de Trabalho\$ sudo add-apt-repository ppa:cz.nic-labs/core
[sudo] senha para abordagembasica22:
ERROR: ppa 'cz.nic-labs/core' not found (use --login if private)

ChatGPT
Peço desculpas pelo inconveniente. Parece que o repositório PPA do CORE não está disponível ou não foi encontrado para adicionar ao seu sistema. Isso pode ocorrer se o repositório foi descontinuado ou se não está mais disponível publicamente.

Nesse caso, você pode tentar outras alternativas para instalar o CORE ou utilizar uma versão anterior do Ubuntu que ainda suporte o repositório do CORE. Aqui estão algumas opções que você pode considerar:

1. Verifique a documentação oficial do CORE: Acesse o site oficial do CORE

(<https://www.nri.navy.mil/ita/ncs/products/core/>) e verifique se eles fornecem instruções de instalação atualizadas para a versão do Ubuntu que você está usando.

2. Compilar o CORE a partir do código-fonte: Se você tiver experiência em compilação de software, pode tentar obter o código-fonte do CORE a partir do repositório oficial (<https://github.com/coreemu/core>) e seguir as instruções de compilação para o Ubuntu.
3. Use uma versão mais antiga do Ubuntu: Se o suporte ao repositório PPA do CORE estiver disponível apenas para versões anteriores do Ubuntu, você pode considerar instalar uma versão mais antiga do Ubuntu que seja compatível com o PPA do CORE.

Lembre-se de que essas são apenas sugestões alternativas, e a disponibilidade e compatibilidade do CORE podem variar. É sempre recomendado consultar a documentação oficial do CORE e verificar se há outras opções de instalação disponíveis.

Anonymous

qual comando para instalar o core emulador no ubuntu 22

ChatGPT

Peço desculpas pela confusão anterior. Vamos prosseguir com a instalação do CORE Emulador no Ubuntu.

O CORE Emulador é um software de código aberto, e você pode instalá-lo através do código-fonte. Siga estes passos para instalar o CORE Emulador:

Passo 1: Instalar as dependências

1. Abra um terminal no Ubuntu 22.
2. Execute os seguintes comandos para instalar as dependências necessárias:

```
sql
sudo apt update
sudo apt install git python3 python3-pip python3-setuptools python3-tk pyth
```

Passo 2: Clonar o repositório do CORE

1. No terminal, execute o seguinte comando para clonar o repositório do CORE do GitHub:

```
bash
git clone https://github.com/coreemu/core.git
```

Passo 3: Instalar o CORE

1. Acesse o diretório do CORE que foi clonado:

```
bash
cd core
```

2. Execute o seguinte comando para instalar o CORE:

```
bash
sudo ./install.sh
```

Após concluir esses passos, o CORE Emulador estará instalado em seu sistema Ubuntu 22. Você pode executar o CORE através do comando `core-gui` em um terminal para abrir a interface gráfica do CORE.

Lembre-se de que o Ubuntu 22 não é uma versão oficialmente lançada até a minha data de conhecimento (setembro de 2021), portanto, pode haver algumas diferenças ou problemas de compatibilidade. Certifique-se de verificar a documentação oficial do CORE para obter informações atualizadas ou consulte a comunidade do CORE para obter suporte adicional.

Anonymous

```
abordagembasica22@abordagembasica22-VirtualBox:~/Área de Trabalho$ sudo apt update
sudo apt install git python3 python3-pip python3-setuptools python3-tk python3-dev qt5-
default libqt5svg5-dev libspice-protocol-dev libspice-server-dev libspice-client-gtk-3.0-dev
libvirt-dev qemu-system-x86 bridge-utils iproute2
Atingido:1 http://br.archive.ubuntu.com/ubuntu jammy InRelease
Obter:2 http://br.archive.ubuntu.com/ubuntu jammy-updates InRelease [119 kB]
Obter:3 http://br.archive.ubuntu.com/ubuntu jammy-backports InRelease [108 kB]
Obter:4 http://br.archive.ubuntu.com/ubuntu jammy-updates/main amd64 DEP-11 Metadata
[102 kB]
Atingido:5 http://security.ubuntu.com/ubuntu jammy-security InRelease
Obter:6 http://br.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 DEP-11
Metadata [269 kB]
Obter:7 http://br.archive.ubuntu.com/ubuntu jammy-updates/multiverse amd64 DEP-11
Metadata [940 B]
Obter:8 http://br.archive.ubuntu.com/ubuntu jammy-backports/main amd64 DEP-11
Metadata [7,972 B]
```

```

Obter:9 http://br.archive.ubuntu.com/ubuntu jammy-backports/universe amd64 DLP-11
Metadata [12,9 kB]
Baixados 619 kB em 1s (935 kB/s)
Lendo listas de pacotes... Pronto
Construindo árvore de dependências... Pronto
Lendo informação de estado... Pronto
23 pacotes podem ser atualizados. Corra 'apt list --upgradable' para vê-los.
Lendo listas de pacotes... Pronto
Construindo árvore de dependências... Pronto
Lendo informação de estado... Pronto
O pacote qt5-default não está disponível, mas é referenciado por outro pacote.
Isto pode significar que o pacote está faltando, ficou obsoleto ou
está disponível somente a partir de outra fonte

```

E: O pacote 'qt5-default' não tem candidato para instalação

ChatGPT

Peço desculpas pelo inconveniente. Parece que o pacote `qt5-default` não está disponível nos repositórios do Ubuntu 22 (Jammy Jellyfish). Isso pode ser devido a mudanças ou atualizações recentes no sistema.

Nesse caso, você pode tentar instalar o pacote `qtbase5-dev` como uma alternativa. Este pacote fornece arquivos de desenvolvimento para o Qt 5 e pode substituir o `qt5-default`.

Execute o seguinte comando no terminal para instalar o `qtbase5-dev`:

```

Copy code
sudo apt install qtbase5-dev

```

Depois de instalar o `qtbase5-dev`, você pode prosseguir com o restante dos comandos:

```

Copy code
perl
sudo apt install git python3 python3-pip python3-setuptools python3-tk pytho

```

Esses comandos instalarão as demais dependências necessárias para o CORE Emulador.

Lamento novamente pela confusão anterior e espero que esta solução alternativa funcione para você.

Anonymous

depois desse comando quais são os passos para executar a interface do core emulador

```

sudo apt install git python3 python3-pip python3-setuptools python3-tk python3-dev libqt5svg5-dev libspice-protocol-dev libspice-server-dev libspice-client-gtk-3.0-dev libvirt-dev qemu-system-x86 bridge-utils iproute2

```

ChatGPT

Após instalar as dependências necessárias, você pode prosseguir com os seguintes passos para executar a interface do CORE Emulador:

Passo 1: Clonar o repositório do CORE

1. Abra um terminal no Ubuntu.
2. Execute o seguinte comando para clonar o repositório do CORE Emulador do GitHub:

```

Copy code
bash
git clone https://github.com/coreemu/core.git

```

Passo 2: Instalar as dependências Python

1. Acesse o diretório do CORE que foi clonado:

```

Copy code
bash
cd core

```

2. Execute o seguinte comando para instalar as dependências Python:

```

Copy code
arduino
sudo python3 setup.py install

```

Passo 3: Executar a interface gráfica do CORE

1. Após a instalação das dependências e do CORE Emulador, execute o seguinte comando para iniciar a interface gráfica do CORE:

```

Copy code
sudo core-gui

```

Isso abrirá a interface gráfica do CORE Emulator, onde você poderá criar e configurar suas redes virtuais.

Certifique-se de ter privilégios administrativos (sudo) ao executar o comando `core-gui` para garantir que o CORE tenha acesso aos recursos necessários.

Lembre-se de que o Ubuntu 22 (Jammy Jellyfish) não é uma versão oficialmente lançada até a minha data de conhecimento (setembro de 2021), portanto, pode haver algumas diferenças ou problemas de compatibilidade. Certifique-se de verificar a documentação oficial do CORE Emulator para obter informações atualizadas ou consulte a comunidade do CORE para obter suporte adicional.

Anonymous
 abordagembasica22@abordagembasica22-VirtualBox:~/Área de Trabalho/core\$ sudo
 python3 setup.py install
 python3: can't open file '/home/abordagembasica22/Área de Trabalho/core/setup.py': [Errno
 2] No such file or directory

ChatGPT
 Peço desculpas pelo equívoco. Parece que ocorreu um erro ao localizar o arquivo
`setup.py` no diretório atual.

Certifique-se de que você está no diretório correto do CORE Emulator antes de executar o comando.

Aqui estão os passos para garantir que você esteja no diretório certo:

1. Abra um terminal no Ubuntu.
2. Navegue até o diretório do CORE Emulator. Se você clonou o repositório na Área de Trabalho, use o seguinte comando para acessar o diretório:

```
bash
cd ~/Área de Trabalho/core
```

Certifique-se de escapar os espaços no nome do diretório usando `\"`.

3. Verifique se o arquivo `setup.py` está presente no diretório. Você pode listar o conteúdo do diretório usando o comando `ls`:

```
bash
ls
```

Verifique se você vê o arquivo `setup.py` listado.

4. Em seguida, execute o comando de instalação do CORE Emulator:

```
arduino
sudo python3 setup.py install
```

Certifique-se de seguir esses passos com cuidado e ajustar o caminho para o diretório do CORE Emulator, caso necessário.

Lamento novamente pela confusão anterior e espero que essas instruções ajudem você a prosseguir com a instalação do CORE Emulator no Ubuntu.

Anonymous
 o arquivo setup.py não esta no diretório core

ChatGPT
 Peço desculpas pela informação incorreta anterior. Parece que houve uma mudança na estrutura do diretório do CORE Emulator.

Para instalar o CORE Emulator no Ubuntu 22, você pode usar o seguinte método:

Passo 1: Clonar o repositório do CORE

1. Abra um terminal no Ubuntu.
2. Execute o seguinte comando para clonar o repositório do CORE Emulator do GitHub:

```
bash
git clone https://github.com/coreemu/core.git
```

Passo 2: Compilar e instalar o CORE Emulator

1. Acesse o diretório do CORE que foi clonado:

```
bash
cd core
```

2. Execute o seguinte comando para compilar e instalar o CORE:

```
bash
./bootstrap.sh
./configure
make
sudo make install
```

Após a conclusão desses comandos, o CORE Emulator estará instalado no seu sistema Ubuntu 22.

Passo 3: Executar a interface gráfica do CORE

1. Para iniciar a interface gráfica do CORE, execute o seguinte comando:

```
Copy code
sudo core-gui
```

Isso abrirá a interface gráfica do CORE Emulator, onde você poderá criar e configurar suas redes virtuais.

Certifique-se de ter privilégios administrativos (sudo) ao executar o comando `core-gui` para garantir que o CORE tenha acesso aos recursos necessários.

Mais uma vez, peço desculpas pela confusão anterior e espero que estas instruções ajudem você a instalar e executar o CORE Emulator no Ubuntu 22.

Anonymous
 abordagembasica22@abordagembasica22-VirtualBox:~/Área de Trabalho/core\$
 ./bootstrap.sh
 ./configure
 make
 sudo make install
 Bootstrapping the autoconf system...
 (Messages below about copying and installing files are normal.)
 (1/4) Running aclocal...
 ./bootstrap.sh: 30: aclocal: not found
 bash: ./configure: Arquivo ou diretório inexistente
 make: *** Nenhum alvo indicado e nenhum arquivo make encontrado. Pare.
 make: *** Sem regra para processar o alvo 'install'. Pare.

ChatGPT
 Peço desculpas novamente pela informação anterior. Parece que o procedimento de instalação do CORE Emulator que eu forneci anteriormente não é aplicável para o Ubuntu 22.

Atualmente, o CORE Emulator não possui suporte oficial para o Ubuntu 22, pois essa versão ainda não foi lançada até a minha data de conhecimento (setembro de 2021). Recomenda-se verificar a documentação oficial do CORE Emulator ou a comunidade para obter informações atualizadas sobre a compatibilidade com o Ubuntu 22.

Você pode tentar as seguintes opções:

1. Verifique se existe uma versão atualizada do CORE Emulator que ofereça suporte ao Ubuntu 22. Visite o site oficial do CORE Emulator ou procure por repositórios alternativos para obter informações atualizadas.
2. Considere utilizar uma versão mais recente do Ubuntu que seja oficialmente suportada pelo CORE Emulator. Isso garantirá uma compatibilidade mais estável e uma instalação mais suave do software.

Lamento pela inconveniência causada e espero que você encontre uma solução alternativa adequada para suas necessidades de emulação de rede.

Anonymous
 abordagembasica22@abordagembasica22-VirtualBox:~/Área de Trabalho/core\$ pip3 install setuptools
 Defaulting to user installation because normal site-packages is not writeable
 Requirement already satisfied: setuptools in /usr/lib/python3/dist-packages (59.6.0)

ChatGPT
 Parece que o pacote `setuptools` já está instalado em sua máquina. A mensagem "Requirement already satisfied" indica que o pacote já foi instalado e está na versão mais recente disponível nos repositórios do sistema.

No entanto, o processo de instalação do CORE Emulator no Ubuntu 22 ainda pode encontrar problemas de compatibilidade. Consulte a documentação do emulador para obter mais informações.

problemas de compatibilidade. Como mencionado anteriormente, o Ubuntu 22 Jammy Jellyfish não é uma versão oficialmente lançada até a minha data de conhecimento (setembro de 2021), e o suporte para o CORE Emulator pode não estar disponível para essa versão específica.

Recomendo que você verifique a documentação oficial do CORE Emulator ou consulte a comunidade do CORE Emulator para obter informações atualizadas sobre a compatibilidade com o Ubuntu 22 ou possíveis alternativas de emulação de rede.

Lamento pela confusão e agradeço sua compreensão.

Anonymous
 abordagembasica22@abordagembasica22-VirtualBox:~/Área de Trabalho/core\$
 ./bootstrap.sh

Bootstrapping the autoconf system...

(Messages below about copying and installing files are normal.)

(1/4) Running aclocal...

./bootstrap.sh: 30: aclocal: not found

ChatGPT

Parece que o comando `aclocal` não está disponível no seu sistema. Isso indica que você precisa instalar o pacote `automake`, que contém o utilitário `aclocal`, responsável pela geração dos arquivos `aclocal.m4` necessários para a configuração do projeto.

Para instalar o pacote `automake` no Ubuntu, você pode executar o seguinte comando:

```
Copy code
sudo apt install automake
```

Após a instalação do `automake`, tente executar novamente o comando `./bootstrap.sh` para iniciar o processo de inicialização do sistema `autoconf`. Certifique-se de estar no diretório correto do CORE Emulator antes de executar o comando.

Lembre-se de que o suporte oficial para o CORE Emulator pode variar de acordo com a versão do Ubuntu, e o Ubuntu 22 Jammy Jellyfish pode não ter suporte oficial no momento. Verifique a documentação oficial do CORE Emulator ou a comunidade para obter informações atualizadas sobre a compatibilidade com o Ubuntu 22 ou possíveis alternativas de emulação de rede.

Peço desculpas pela confusão anterior e espero que essas instruções ajudem você a prosseguir com a instalação do CORE Emulator.

Anonymous
 abordagembasica22@abordagembasica22-VirtualBox:~/Área de Trabalho/core\$./configure

checking for a BSD-compatible install... /usr/bin/install -c

checking whether build environment is sane... yes

checking for a race-free mkdir -p... /usr/bin/mkdir -p

checking for gawk... no

checking for mawk... mawk

checking whether make sets \$(MAKE)... yes

checking whether make supports nested variables... yes

checking whether UID '1000' is supported by ustar format... yes

checking whether GID '1000' is supported by ustar format... yes

checking how to create a ustar tar archive... gnutar

checking for gawk... (cached) mawk

checking for gcc... gcc

checking whether the C compiler works... yes

checking for C compiler default output file name... a.out

checking for suffix of executables...

checking whether the linker supports shared libraries... yes

```

checking whether we are cross compiling... no
checking for suffix of object files... o
checking whether the compiler supports GNU C... yes
checking whether gcc accepts -g... yes
checking for gcc option to enable C11 features... none needed
checking whether gcc understands -c and -o together... yes
checking whether make supports the include directive... yes (GNU style)
checking dependency style of gcc... gcc3
checking whether make sets $(MAKE)... (cached) yes
checking for ranlib... ranlib
checking for a sed that does not truncate output... /usr/bin/sed
checking for a Python interpreter with version >= 3.9... python3
checking for python3... /usr/bin/python3
checking for python3 version... 310
checking for python3 platform... linux
checking for GNU default python3 prefix... $(prefix)
checking for GNU default python3 exec_prefix... $(exec_prefix)
checking for python3 script directory (pythondir)... $(PYTHON_PREFIX)/lib/python3.10/site-packages
checking for python3 extension module directory (pyexecdir)...
$(PYTHON_EXEC_PREFIX)/lib/python3.10/site-packages
configure: error: please install python grpcio-tools

```

 ChatGPT

Parece que você precisa instalar o pacote `grpcio-tools` para prosseguir com a configuração do CORE Emulator. O erro exibido indica que o pacote `grpcio-tools` não está instalado no seu sistema.

Para instalar o pacote `grpcio-tools` no Ubuntu, você pode executar o seguinte comando:

```


Copy code
sudo apt install python3-grpcio-tools

```

Após a instalação do `grpcio-tools`, tente executar novamente o comando `./configure` para continuar com o processo de configuração do CORE Emulator. Certifique-se de estar no diretório correto do CORE Emulator antes de executar o comando.

Lembre-se de que o suporte oficial para o CORE Emulator pode variar de acordo com a versão do Ubuntu, e o Ubuntu 22 (Jammy Jellyfish) pode não ter suporte oficial no momento. Verifique a documentação oficial do CORE Emulator ou a comunidade para obter informações atualizadas sobre a compatibilidade com o Ubuntu 22 ou possíveis alternativas de emulação de rede.

Peço desculpas pela confusão anterior e espero que essas instruções ajudem você a prosseguir com a configuração do CORE Emulator.

 Anonymous

```

abordagembasica22@abordagembasica22-VirtualBox:~/Área de Trabalho/core$ sudo apt
install python3-grpcio-tools

```

Lendo listas de pacotes... Pronto

Construindo árvore de dependências... Pronto

Lendo informação de estado... Pronto

E: Impossível encontrar o pacote python3-grpcio-tools

 ChatGPT

Peço desculpas pela informação incorreta anteriormente. Parece que o pacote `python3-grpcio-tools` não está disponível nos repositórios padrão do Ubuntu.

No entanto, você pode tentar instalar o pacote `grpcio-tools` usando o gerenciador de pacotes `pip`. O `pip` é uma ferramenta de gerenciamento de pacotes do Python que permite instalar pacotes Python de repositórios externos.

Certifique-se de ter o `pip` instalado em seu sistema executando o seguinte comando:

```
Copy code
sudo apt install python3-pip
```

Após instalar o `pip`, você pode tentar instalar o pacote `grpcio-tools` usando o `pip` executando o seguinte comando:

```
Copy code
sudo pip3 install grpcio-tools
```

Isso instalará o pacote `grpcio-tools` necessário para a configuração do CORE Emulator.

Lembre-se de que o suporte oficial para o CORE Emulator pode variar de acordo com a versão do Ubuntu, e o Ubuntu 22 (Jammy Jellyfish) pode não ter suporte oficial no momento. Verifique a documentação oficial do CORE Emulator ou a comunidade para obter informações atualizadas sobre a compatibilidade com o Ubuntu 22 ou possíveis alternativas de emulação de rede.

Peço desculpas novamente pela confusão e espero que essas instruções ajudem você a prosseguir com a configuração do CORE Emulator.

Anonymous
 abordagembasica22@abordagembasica22-VirtualBox:~/Área de Trabalho/core\$ make
 make: *** Nenhum alvo indicado e nenhum arquivo make encontrado. Pare.

ChatGPT
 Peço desculpas pela confusão anterior. Parece que o comando `make` não está funcionando corretamente no diretório do CORE Emulator.

Normalmente, o comando `make` é usado para compilar o código-fonte e gerar os executáveis do projeto. No entanto, no caso do CORE Emulator, o processo de compilação pode exigir etapas adicionais e dependências específicas.

Certifique-se de seguir as instruções de instalação e configuração fornecidas pelo projeto CORE Emulator. Verifique se você concluiu todas as etapas anteriores corretamente, incluindo a instalação das dependências e a execução dos comandos de configuração, como `./configure`.

Recomendo verificar a documentação oficial do CORE Emulator ou a página do projeto para obter instruções detalhadas sobre como compilar e instalar corretamente o CORE Emulator no Ubuntu 22 (Jammy Jellyfish) ou em qualquer outra versão suportada do Ubuntu.

Lamento pela confusão e espero que essas informações sejam úteis para você avançar com a instalação do CORE Emulator.

Anonymous
 abordagembasica22@abordagembasica22-VirtualBox:~/Área de Trabalho/core\$./configure
 checking for a BSD-compatible install... /usr/bin/install -c
 checking whether build environment is sane... yes
 checking for a race-free mkdir -p... /usr/bin/mkdir -p
 checking for gawk... no
 checking for mawk... mawk
 checking whether make sets \$(MAKE)... yes
 checking whether make supports nested variables... yes
 checking whether UID '1000' is supported by ustar format... yes
 checking whether GID '1000' is supported by ustar format... yes
 checking how to create a ustar tar archive... gnutar
 checking for gawk... (cached) mawk
 checking for gcc... gcc

checking whether the C compiler works... yes

checking for C compiler default output file name... a.out

checking for suffix of executables...

checking whether we are cross compiling... no

checking for suffix of object files... o

checking whether the compiler supports GNU C... yes

checking whether gcc accepts -g... yes

checking for gcc option to enable C11 features... none needed

checking whether gcc understands -c and -o together... yes

checking whether make supports the include directive... yes (GNU style)

checking dependency style of gcc... gcc3

checking whether make sets \$(MAKE)... (cached) yes

checking for ranlib... ranlib

checking for a sed that does not truncate output... /usr/bin/sed

checking for a Python interpreter with version >= 3.9... python3

checking for python3... /usr/bin/python3

checking for python3 version... 3.10

checking for python3 platform... linux

checking for GNU default python3 prefix... \${prefix}

checking for GNU default python3 exec_prefix... \${exec_prefix}

checking for python3 script directory (pythondir)... \${PYTHON_PREFIX}/lib/python3.10/site-packages

checking for python3 extension module directory (pyexecdir)...
\${PYTHON_EXEC_PREFIX}/lib/python3.10/site-packages

checking for sysctl... /usr/sbin/

checking for nft... /usr/sbin/

checking for ip... /usr/sbin/

checking for tc... /usr/sbin/

checking for ethtool... /usr/sbin/

checking for mount... /usr/bin/

checking for umount... /usr/bin/

checking for convert... no

configure: WARNING: Could not locate ImageMagick convert.

checking for ovs-vsctl... no

configure: WARNING: Could not locate ovs-vsctl cannot use OVS mode

checking for ovs-ofctl... no

configure: WARNING: Could not locate ovs-ofctl cannot use OVS mode

checking for stdio.h... yes

checking for stdlib.h... yes

checking for string.h... yes

checking for inttypes.h... yes

checking for stdint.h... yes

checking for strings.h... yes

```
checking for sys/stat.h... yes
checking for sys/types.h... yes
checking for unistd.h... yes
checking for vfork.h... no
checking for arpa/inet.h... yes
checking for fcntl.h... yes
checking for limits.h... yes
checking for stdint.h... (cached) yes
checking for stdlib.h... (cached) yes
checking for string.h... (cached) yes
checking for sys/ioctl.h... yes
checking for sys/mount.h... yes
checking for sys/socket.h... yes
checking for sys/time.h... yes
checking for termios.h... yes
checking for unistd.h... (cached) yes
checking for inline... inline
checking for int32_t... yes
checking for pid_t... yes
checking for size_t... yes
checking for ssize_t... yes
checking for uint32_t... yes
checking for uint8_t... yes
checking for fork... no
checking for vfork... yes
checking for working vfork... (cached) yes
checking build system type... x86_64-pc-linux-gnu
checking host system type... x86_64-pc-linux-gnu
checking for GNU libc compatible malloc... yes
checking for GNU libc compatible realloc... yes
checking for atexit... yes
checking for dup2... yes
checking for gettimeofday... yes
checking for memset... no
checking for socket... yes
checking for strerror... yes
checking for uname... yes
checking for pkg-config... /usr/bin/pkg-config
checking pkg-config is at least version 0.9.0... yes
checking for libev... no
did not find libev using pkgconfig...
```

```
checking for ev_set_allocator in -lev... no
```

```
configure: error: Python bindings require libev (try installing your 'libev-devel' or 'libev-dev' package)
```

ChatGPT

O erro indica que você precisa instalar a biblioteca `libev` para continuar com a configuração do CORE Emulator. Você pode tentar instalar o pacote `libev-dev` usando o gerenciador de pacotes `apt`.

Execute o seguinte comando para instalar o `libev-dev`:

```
Copy code
sudo apt install libev-dev
```

Após a instalação bem-sucedida, tente executar novamente o comando `./configure` para prosseguir com a configuração do CORE Emulator. Certifique-se de que todas as dependências estejam instaladas corretamente antes de prosseguir para a próxima etapa.

Anonymous

```
abordagembasica22@abordagembasica22-VirtualBox:~/Área de Trabalho/core$ sudo apt install libev-dev
```

```
Lendo listas de pacotes... Pronto
```

```
Construindo árvore de dependências... Pronto
```

```
Lendo informação de estado... Pronto
```

```
O seguinte pacote foi instalado automaticamente e já não é necessário:
```

```
libreoffice-ogltrans
```

```
Utilize 'sudo apt autoremove' para o remover.
```

```
Os pacotes adicionais seguintes serão instalados:
```

```
libev4
```

```
Os NOVOS pacotes a seguir serão instalados:
```

```
libev-dev libev4
```

```
O pacotes atualizados, 2 pacotes novos instalados, 0 a serem removidos e 21 não atualizados.
```

```
É preciso baixar 152 kB de arquivos.
```

```
Depois desta operação, 333 kB adicionais de espaço em disco serão usados.
```

```
Você quer continuar? [S/n] S
```

```
Obter:1 http://br.archive.ubuntu.com/ubuntu jammy/universe amd64 libev4 amd64 1:4.33-1 [29,4 kB]
```

```
Obter:2 http://br.archive.ubuntu.com/ubuntu jammy/universe amd64 libev-dev amd64 1:4.33-1 [123 kB]
```

```
Baixados 152 kB em 1s (134 kB/s)
```

```
A seleccionar pacote anteriormente não seleccionado libev4:amd64.
```

```
(Lendo banco de dados ... 223501 ficheiros e directórios actualmente instalados
```

```
)
```

```
A preparar para desempacotar .../libev4_1%3a4.33-1_amd64.deb ...
```

```
A descompactar libev4:amd64 (1:4.33-1) ...
```

```
A seleccionar pacote anteriormente não seleccionado libev-dev:amd64.
```

```
A preparar para desempacotar .../libev-dev_1%3a4.33-1_amd64.deb ...
```

```
A descompactar libev-dev:amd64 (1:4.33-1) ...
```

```
Configurando libev4:amd64 (1:4.33-1) ...
```

```
Configurando libev-dev:amd64 (1:4.33-1) ...
```

```
A processar 'triggers' para man-db (2:10.2-1) ...
```

A processar 'triggers' para libc-bin (2.35-0ubuntu3.1) ...

ChatGPT

A instalação do pacote `libev-dev` foi concluída com sucesso. Agora você pode tentar executar novamente o comando `./configure` para prosseguir com a configuração do CORE Emulator. Certifique-se de que todas as dependências estejam instaladas corretamente antes de prosseguir para a próxima etapa.

Anonymous

abordagembasica22@abordagembasica22-VirtualBox:~/Área de Trabalho/core\$./configure

```
checking for a BSD-compatible install... /usr/bin/install -c
checking whether build environment is sane... yes
checking for a race-free mkdir -p... /usr/bin/mkdir -p
checking for gawk... no
checking for mawk... mawk
checking whether make sets $(MAKE)... yes
checking whether make supports nested variables... yes
checking whether UID '1000' is supported by ustar format... yes
checking whether GID '1000' is supported by ustar format... yes
checking how to create a ustar tar archive... gnutar
checking for gawk... (cached) mawk
checking for gcc... gcc
checking whether the C compiler works... yes
checking for C compiler default output file name... a.out
checking for suffix of executables...
checking whether we are cross compiling... no
checking for suffix of object files... o
checking whether the compiler supports GNU C... yes
checking whether gcc accepts -g... yes
checking for gcc option to enable C11 features... none needed
checking whether gcc understands -c and -o together... yes
checking whether make supports the include directive... yes (GNU style)
checking dependency style of gcc... gcc3
checking whether make sets $(MAKE)... (cached) yes
checking for ranlib... ranlib
checking for a sed that does not truncate output... /usr/bin/sed
checking for a Python interpreter with version >= 3.9... python3
checking for python3... /usr/bin/python3
checking for python3 version... 3.10
checking for python3 platform... linux
checking for GNU default python3 prefix... ${prefix}
checking for GNU default python3 exec_prefix... ${exec_prefix}
checking for python3 script directory (pythondir)... ${PYTHON_PREFIX}/lib/python3.10/site-packages
checking for python3 extension module directory (pyexecdir)...
${PYTHON_EXEC_PREFIX}/lib/python3.10/site-packages
```

```
checking for sysctl... /usr/sbin/
checking for nft... /usr/sbin/
checking for ip... /usr/sbin/
checking for tc... /usr/sbin/
checking for ethtool... /usr/sbin/
checking for mount... /usr/bin/
checking for umount... /usr/bin/
checking for convert... no
configure: WARNING: Could not locate ImageMagick convert.
checking for ovs-vsctl... no
configure: WARNING: Could not locate ovs-vsctl cannot use OVS mode
checking for ovs-ofctl... no
configure: WARNING: Could not locate ovs-ofctl cannot use OVS mode
checking for stdio.h... yes
checking for stdlib.h... yes
checking for string.h... yes
checking for inttypes.h... yes
checking for stdint.h... yes
checking for strings.h... yes
checking for sys/stat.h... yes
checking for sys/types.h... yes
checking forunistd.h... yes
checking for vfork.h... no
checking for arpa/inet.h... yes
checking forfcntl.h... yes
checking for limits.h... yes
checking for stdint.h... (cached) yes
checking for stdlib.h... (cached) yes
checking for string.h... (cached) yes
checking for sys/ioctl.h... yes
checking for sys/mount.h... yes
checking for sys/socket.h... yes
checking for sys/time.h... yes
checking for termios.h... yes
checking forunistd.h... (cached) yes
checking for inline... inline
checking for int32_t... yes
checking for pid_t... yes
checking for size_t... yes
checking for ssize_t... yes
checking for uint32_t... yes
checking for uint8_t... yes
```

```
checking for fork... no
checking for vfork... yes
checking for working vfork... (cached) yes
checking build system type... x86_64-pc-linux-gnu
checking host system type... x86_64-pc-linux-gnu
checking for GNU libc compatible malloc... yes
checking for GNU libc compatible realloc... yes
checking for atexit... yes
checking for dup2... yes
checking for gettimeofday... yes
checking for memset... no
checking for socket... yes
checking for strerror... yes
checking for uname... yes
checking for pkg-config... /usr/bin/pkg-config
checking pkg-config is at least version 0.9.0... yes
checking for libev... no
did not find libev using pkgconfig...
checking for ev_set_allocator in -lev... yes
found libev OK
checking that generated files are newer than configure... done
configure: creating ./config.status
config.status: creating Makefile
config.status: creating man/Makefile
config.status: creating docs/Makefile
config.status: creating daemon/Makefile
config.status: creating daemon/doc/Makefile
config.status: creating daemon/doc/conf.py
config.status: creating daemon/proto/Makefile
config.status: creating netns/Makefile
config.status: creating netns/version.h
config.status: creating config.h
config.status: executing depfiles commands
-----
core 9.0.2 Configuration:

Build:
Host System Type:   x86_64-pc-linux-gnu
C Compiler and flags: gcc -O3 -Werror -Wall -D_GNU_SOURCE
Prefix:             /usr/local
Exec Prefix:        ${prefix}
```

```

Daemon:

Daemon path:    $(exec_prefix)/bin

Daemon config:  /etc/core

Python:        /usr/bin/python3

```

Features to build:

```

Build Daemon:   yes

Documentation:  no

```

On this platform you should run core-gui as a normal user.

ChatGPT

A configuração do CORE Emulator foi concluída com sucesso. Agora você pode prosseguir com a compilação e instalação executando o comando **make**. Certifique-se de estar executando o comando na pasta correta onde o código-fonte do CORE Emulator está localizado.

Anonymous

```
abordagembasica22@abordagembasica22-VirtualBox:~/Área de Trabalho/core$ make
```

```
Makefile:984: aviso: ignorando pré-requisitos na definição de regra de sufixo
```

```
creating file daemon/core/constants.py from daemon/core/constants.py.in
```

```
creating file netns/setup.py from netns/setup.py.in
```

```
make all-recursive
```

```
make[1]: Entrando no diretório '/home/abordagembasica22/Área de Trabalho/core'
```

```
Makefile:984: aviso: ignorando pré-requisitos na definição de regra de sufixo
```

```
Making all in daemon
```

```
make[2]: Entrando no diretório '/home/abordagembasica22/Área de Trabalho/core/daemon'
```

```
Making all in proto
```

```
make[3]: Entrando no diretório '/home/abordagembasica22/Área de Trabalho/core/daemon/proto'
```

```
/usr/bin/python3 -m grpc_tools.protoc -I. --python_out=. core/api/grpc/*.proto
```

```
/usr/bin/python3 -m grpc_tools.protoc -I. --grpc_python_out=. core/api/grpc/core.proto
```

```
make[3]: Saindo do diretório '/home/abordagembasica22/Área de Trabalho/core/daemon/proto'
```

```
make[3]: Entrando no diretório '/home/abordagembasica22/Área de Trabalho/core/daemon'
```

```
make[3]: Nada a ser feito para 'all-am'.
```

```
make[3]: Saindo do diretório '/home/abordagembasica22/Área de Trabalho/core/daemon'
```

```
make[2]: Saindo do diretório '/home/abordagembasica22/Área de Trabalho/core/daemon'
```

```
Making all in netns
```

```
make[2]: Entrando no diretório '/home/abordagembasica22/Área de Trabalho/core/netns'
```

```
gcc -DHAVE_CONFIG_H -I. -Wall -fno-strict-aliasing -O3 -g -O3 -Werror -Wall -D_GNU_SOURCE -MT vnode_msg.o -MD -MP -MF .deps/vnode_msg.Tpo -c -o vnode_msg.o vnode_msg.c
```

```
mv -f .deps/vnode_msg.Tpo .deps/vnode_msg.Po
```

```
gcc -DHAVE_CONFIG_H -I. -Wall -fno-strict-aliasing -O3 -g -O3 -Werror -Wall -D_GNU_SOURCE -MT vnode_cmd.o -MD -MP -MF .deps/vnode_cmd.Tpo -c -o vnode_cmd.o
```

```

vnode_cmd.c

mv -f .deps/vnode_cmd.Tpo .deps/vnode_cmd.Po

gcc -DHAVE_CONFIG_H -I. -L. -Wall -fno-strict-aliasing -O3 -g -O3 -Werror -Wall -
D_GNU_SOURCE -MT vnode_chn.lo -MD -MP -MF .deps/vnode_chnl.Tpo -c -o vnode_chn.lo
vnode_chnl.c

mv -f .deps/vnode_chnl.Tpo .deps/vnode_chnl.Po

gcc -DHAVE_CONFIG_H -I. -L. -Wall -fno-strict-aliasing -O3 -g -O3 -Werror -Wall -
D_GNU_SOURCE -MT vnode_io.o -MD -MP -MF .deps/vnode_io.Tpo -c -o vnode_io.o
vnode_io.c

mv -f .deps/vnode_io.Tpo .deps/vnode_io.Po

gcc -DHAVE_CONFIG_H -I. -L. -Wall -fno-strict-aliasing -O3 -g -O3 -Werror -Wall -
D_GNU_SOURCE -MT vnoded_main.o -MD -MP -MF .deps/vnoded_main.Tpo -c -o
vnoded_main.o vnoded_main.c

mv -f .deps/vnoded_main.Tpo .deps/vnoded_main.Po

gcc -DHAVE_CONFIG_H -I. -L. -Wall -fno-strict-aliasing -O3 -g -O3 -Werror -Wall -
D_GNU_SOURCE -MT vnode_server.o -MD -MP -MF .deps/vnode_server.Tpo -c -o
vnode_server.o vnode_server.c

mv -f .deps/vnode_server.Tpo .deps/vnode_server.Po

gcc -DHAVE_CONFIG_H -I. -L. -Wall -fno-strict-aliasing -O3 -g -O3 -Werror -Wall -
D_GNU_SOURCE -MT netns.o -MD -MP -MF .deps/netns.Tpo -c -o netns.o netns.c

mv -f .deps/netns.Tpo .deps/netns.Po

gcc -Wall -fno-strict-aliasing -O3 -g -O3 -Werror -Wall -D_GNU_SOURCE -o vnoded
vnode_msg.o vnode_cmd.o vnode_chn.lo vnode_io.o vnoded_main.o vnode_server.o netns.o
-lev

gcc -DHAVE_CONFIG_H -I. -L. -Wall -fno-strict-aliasing -O3 -g -O3 -Werror -Wall -
D_GNU_SOURCE -MT vcmd_main.o -MD -MP -MF .deps/vcmd_main.Tpo -c -o vcmd_main.o
vcmd_main.c

mv -f .deps/vcmd_main.Tpo .deps/vcmd_main.Po

gcc -DHAVE_CONFIG_H -I. -L. -Wall -fno-strict-aliasing -O3 -g -O3 -Werror -Wall -
D_GNU_SOURCE -MT vnode_client.o -MD -MP -MF .deps/vnode_client.Tpo -c -o
vnode_client.o vnode_client.c

mv -f .deps/vnode_client.Tpo .deps/vnode_client.Po

gcc -Wall -fno-strict-aliasing -O3 -g -O3 -Werror -Wall -D_GNU_SOURCE -o vcmd
vnode_msg.o vnode_cmd.o vnode_chn.lo vnode_io.o vcmd_main.o vnode_client.o -lev

gcc -DHAVE_CONFIG_H -I. -L. -Wall -fno-strict-aliasing -O3 -g -O3 -Werror -Wall -
D_GNU_SOURCE -MT netns_main.o -MD -MP -MF .deps/netns_main.Tpo -c -o netns_main.o
netns_main.c

mv -f .deps/netns_main.Tpo .deps/netns_main.Po

gcc -Wall -fno-strict-aliasing -O3 -g -O3 -Werror -Wall -D_GNU_SOURCE -o netns
netns_main.o netns.o

make[2]: Saindo do diretório '/home/abordagembasica22/Área de Trabalho/core/netns'

make[2]: Entrando no diretório '/home/abordagembasica22/Área de Trabalho/core'

Makefile:904: aviso: ignorando pré-requisitos na definição de regra de sufixo

make[2]: Saindo do diretório '/home/abordagembasica22/Área de Trabalho/core'

make[1]: Saindo do diretório '/home/abordagembasica22/Área de Trabalho/core'

```

ChatGPT

A compilação do CORE Emulator foi concluída com sucesso. Agora você pode prosseguir com a instalação executando o comando `sudo make install`. Isso instalará o CORE Emulator no sistema. Certifique-se de ter privilégios de administrador para executar o comando.

Anonymous

abordagembasica22@abordagembasica22-VirtualBox:~/Área de Trabalho/core\$ sudo make install

Makefile:904: aviso: ignorando pré-requisitos na definição de regra de sufixo

Making install in daemon

make[1]: Entrando no diretório '/home/abordagembasica22/Área de Trabalho/core/daemon'

Making install in proto

make[2]: Entrando no diretório '/home/abordagembasica22/Área de Trabalho/core/daemon/proto'

make[3]: Entrando no diretório '/home/abordagembasica22/Área de Trabalho/core/daemon/proto'

make[3]: Nada a ser feito para 'install-exec-am'.

make[3]: Nada a ser feito para 'install-data-am'.

make[3]: Saindo do diretório '/home/abordagembasica22/Área de Trabalho/core/daemon/proto'

make[2]: Saindo do diretório '/home/abordagembasica22/Área de Trabalho/core/daemon/proto'

make[2]: Entrando no diretório '/home/abordagembasica22/Área de Trabalho/core/daemon'

make[3]: Entrando no diretório '/home/abordagembasica22/Área de Trabalho/core/daemon'

make[3]: Nada a ser feito para 'install-exec-am'.

make[3]: Nada a ser feito para 'install-data-am'.

make[3]: Saindo do diretório '/home/abordagembasica22/Área de Trabalho/core/daemon'

make[2]: Saindo do diretório '/home/abordagembasica22/Área de Trabalho/core/daemon'

make[1]: Saindo do diretório '/home/abordagembasica22/Área de Trabalho/core/daemon'

Making install in netns

make[1]: Entrando no diretório '/home/abordagembasica22/Área de Trabalho/core/netns'

make[2]: Entrando no diretório '/home/abordagembasica22/Área de Trabalho/core/netns'

/usr/bin/mkdir -p '/usr/local/bin'

/usr/bin/install -c vnode vcmd netns '/usr/local/bin'

make[2]: Nada a ser feito para 'install-data-am'.

make[2]: Saindo do diretório '/home/abordagembasica22/Área de Trabalho/core/netns'

make[1]: Saindo do diretório '/home/abordagembasica22/Área de Trabalho/core/netns'

make[1]: Entrando no diretório '/home/abordagembasica22/Área de Trabalho/core'

Makefile:984: aviso: ignorando pré-requisitos na definição de regra de sufixo

make[2]: Entrando no diretório '/home/abordagembasica22/Área de Trabalho/core'

Makefile:984: aviso: ignorando pré-requisitos na definição de regra de sufixo

make[2]: Nada a ser feito para 'install-exec-am'.

make[2]: Nada a ser feito para 'install-data-am'.

make[2]: Saindo do diretório '/home/abordagembasica22/Área de Trabalho/core'

make[1]: Saindo do diretório '/home/abordagembasica22/Área de Trabalho/core'



ChatGPT

A instalação do CORE Emulator foi concluída com sucesso. Agora você pode executar o CORE Emulator digitando `core-gui` no terminal para iniciar a interface gráfica do usuário ou `core-cli` para iniciar a interface de linha de comando.

Certifique-se de ter todos os requisitos de sistema atendidos para que o CORE Emulator funcione corretamente. Se você encontrar algum problema durante a execução do CORE, verifique a documentação ou o site oficial para obter assistência adicional.



Anonymous

abordagembasica22@abordagembasica22-VirtualBox:~/Área de Trabalho\$ core-gui

core-gui: comando não encontrado

ChatGPT

Se você recebeu o erro "core-gui: comando não encontrado", pode ser que o local onde o CORE Emulator foi instalado não está incluído no seu caminho de busca ('\$PATH'). O diretório onde o CORE Emulator foi instalado é `~/usr/local/bin`.

Para resolver esse problema, você pode tentar executar o comando `core-gui` fornecendo o caminho completo para o executável. No seu caso, o comando completo seria `~/usr/local/bin/core-gui`.

Tente executar o comando completo no terminal e veja se o CORE Emulator é iniciado corretamente. Se isso funcionar, você pode adicionar o diretório `~/usr/local/bin` ao seu caminho de busca para que você possa executar o `core-gui` apenas digitando o comando sem especificar o caminho completo.

Para adicionar o diretório ao seu caminho de busca temporariamente, você pode executar o seguinte comando:

```
bash
export PATH="$usr/local/bin:$PATH"
```

Após executar esse comando, tente executar `core-gui` novamente e veja se o CORE Emulator é iniciado corretamente.

Se você quiser adicionar o diretório permanentemente ao seu caminho de busca, você pode editar o arquivo de perfil do seu shell (por exemplo, `~/.bashrc`, `~/.bash_profile`, `~/.zshrc`) e adicionar a linha `export PATH="$usr/local/bin:$PATH"` no final do arquivo. Salve as alterações e reinicie o terminal ou execute o comando `source` seguido do nome do arquivo de perfil para atualizar o caminho de busca sem reiniciar o terminal.

Espero que isso ajude!

Anonymous

abordagembasica22@abordagembasica22-VirtualBox:~/Área de Trabalho\$
~/usr/local/bin/core-gui

bash: ~/usr/local/bin/core-gui: Arquivo ou diretório inexistente

ChatGPT

Peço desculpas pelo incômodo. Parece que o caminho `~/usr/local/bin/core-gui` não está correto ou o arquivo `core-gui` não está presente nesse diretório.

Para verificar a localização correta do executável do CORE Emulator, você pode usar o comando `which core-gui`. Isso fornecerá o caminho correto, se o comando estiver instalado corretamente.

Se o comando `which core-gui` não retornar nenhum resultado, isso pode indicar que o CORE Emulator não foi instalado corretamente ou não está disponível no seu sistema. Nesse caso, sugiro verificar se você seguiu corretamente as instruções de instalação do CORE Emulator e se todos os requisitos foram atendidos.

Certifique-se de ter concluído todos os passos de instalação corretamente, incluindo a compilação e instalação do CORE Emulator a partir da fonte. Verifique se não houve erros durante a instalação.

Se você seguiu todos os passos corretamente e ainda assim não consegue encontrar o executável do CORE Emulator, pode ser necessário revisar as instruções de instalação ou procurar assistência adicional na documentação oficial ou na comunidade do CORE Emulator.

Anonymous

o comando which core-gui não retorna nada

ChatGPT

Se o comando `which core-gui` não retorna nada, isso indica que o executável do CORE Emulator não está localizado em nenhum dos diretórios listados na variável de ambiente PATH.

Nesse caso, pode ser necessário adicionar o diretório correto do executável do CORE Emulator ao PATH ou verificar se o CORE Emulator foi instalado corretamente.

Para adicionar um diretório ao PATH, você pode editar o arquivo de perfil do seu shell (por exemplo, `~/.bashrc` para o Bash) e adicionar a seguinte linha no final do arquivo:

```
bash
export PATH="$caminho/para/o/core-emulator:$PATH"
```

Lembre-se de substituir `~/caminho/para/o/core-emulator` pelo diretório correto onde o

CORE Emulator está instalado.

Após editar o arquivo de perfil, salve-o e feche-o. Em seguida, execute o seguinte comando para recarregar o arquivo de perfil e atualizar o PATH:

```
bash
source ~/.bashrc
```

Depois disso, tente executar o comando `core-gui` novamente para verificar se o CORE Emulator é encontrado corretamente.

Se mesmo assim o problema persistir, pode ser necessário revisar as instruções de instalação ou buscar assistência na documentação oficial ou na comunidade do CORE Emulator para garantir que a instalação tenha sido realizada corretamente.

Anonymous
devo dar esse comando ~/.bashrc

ChatGPT
O comando `~/.bashrc` não deve ser executado diretamente. Ele se refere ao arquivo de perfil do shell Bash e é necessário abri-lo em um editor de texto para adicionar ou editar o conteúdo.

Siga os passos abaixo para editar o arquivo `~/.bashrc` corretamente:

1. Abra um terminal.
2. Digite o seguinte comando para abrir o arquivo `~/.bashrc` no editor de texto Nano:

```
bash
nano ~/.bashrc
```

3. Role até o final do arquivo e adicione a seguinte linha:

```
bash
export PATH="/caminho/para/o/core-emulador:$PATH"
```

Certifique-se de substituir `/caminho/para/o/core-emulador` pelo diretório correto onde o CORE Emulator está instalado.

4. Pressione "Ctrl + O" para salvar o arquivo.
5. Pressione "Enter" para confirmar o nome do arquivo.
6. Pressione "Ctrl + X" para sair do editor Nano.

Depois de editar o arquivo `~/.bashrc`, você precisa recarregar o arquivo para que as alterações tenham efeito no terminal atual. Para fazer isso, execute o seguinte comando:

```
bash
source ~/.bashrc
```

Agora você deve poder executar o comando `core-gui` para iniciar o CORE Emulator corretamente. Certifique-se de usar o comando `core-gui` em um novo terminal, pois as alterações no `~/.bashrc` só afetam novos terminais abertos após o recarregamento.

Anonymous
devo digitar o comando core-gui

ChatGPT
Sim, depois de adicionar a linha correta no arquivo `~/.bashrc` e recarregar o arquivo usando o comando `source ~/.bashrc`, você deve poder executar o comando `core-gui` diretamente no terminal.

Certifique-se de abrir um novo terminal ou reiniciar o terminal atual antes de executar o comando `core-gui`. Dessa forma, as alterações feitas no arquivo `~/.bashrc` serão aplicadas.

Digite o seguinte comando no terminal:

```
bash
core-gui
```

Isso deve iniciar o CORE Emulator GUI corretamente, desde que todas as etapas anteriores tenham sido seguidas corretamente.

Anonymous
abordagembasica22@abordagembasica22-VirtualBox:~/Área de Trabalho\$ core-gui

core-gui comando não encontrado

core-gui, comando não encontrado

ChatGPT

Se o comando `core-gui` ainda não está sendo encontrado, pode haver algumas possibilidades:

1. Verifique se você seguiu corretamente as etapas anteriores para compilar e instalar o CORE Emulator. Certifique-se de ter executado o comando `sudo make install` sem erros.
2. Verifique se o diretório `/usr/local/bin` está incluído na variável de ambiente `PATH`. Você pode verificar isso executando o comando `echo $PATH`. Verifique se o diretório `/usr/local/bin` está listado.
3. Verifique se o arquivo `core-gui` está presente no diretório `/usr/local/bin`. Você pode verificar isso navegando até o diretório `/usr/local/bin` usando o comando `cd /usr/local/bin` e, em seguida, listar o conteúdo usando o comando `ls`. Certifique-se de que o arquivo `core-gui` esteja presente na lista.

Se todas as etapas anteriores estiverem corretas e você ainda não conseguir encontrar o comando `core-gui`, pode haver algum problema durante a instalação. Nesse caso, sugiro revisar todas as etapas novamente e garantir que não ocorram erros durante a compilação e instalação do CORE Emulator.

Anonymous

,como executar o comando echo \$PATH

ChatGPT

Para executar o comando `echo $PATH`, abra um terminal e siga as etapas abaixo:

1. Abra o terminal no seu sistema operacional (no Ubuntu, pressione `Ctrl + Alt + T` para abrir o terminal).
2. Digite o comando `echo $PATH` no terminal.
3. Pressione Enter.

Isso exibirá o valor da variável de ambiente `PATH`, que é uma lista de diretórios separados por dois-pontos. Verifique se o diretório `/usr/local/bin` está incluído na lista.

Anonymous

abordagembasica22@abordagembasica22-VirtualBox:~/Área de Trabalho\$ echo \$PATH

```
/caminho/para/o/core-emulator:/usr/local/bin:/usr/local/sbin:/usr/bin:/usr/sbin:/bin:/sbin:/bin:/usr/games
```

ChatGPT

Obrigado por compartilhar a saída do comando `echo $PATH`. Com base na saída fornecida, o diretório `/usr/local/bin` está presente em seu `PATH`, o que significa que os comandos localizados nesse diretório devem ser executáveis a partir do terminal.

No entanto, como o comando `core-gui` ainda não está sendo encontrado, pode haver um problema de configuração ou instalação. É possível que o pacote `core-gui` não tenha sido instalado corretamente ou não esteja disponível no diretório esperado.

Recomendo verificar se o pacote `core-gui` foi instalado corretamente e, se necessário, verificar as instruções de instalação fornecidas pela fonte para garantir que todos os requisitos tenham sido atendidos. Se você tiver seguido todas as etapas corretamente e ainda não conseguir executar o `core-gui`, pode ser útil procurar ajuda no fórum ou nos recursos de suporte fornecidos pelo projeto `core-emulator`.

Anonymous

abordagembasica22@abordagembasica22-VirtualBox:~/Área de Trabalho\$ cd usr

```
bash: cd: usr: Arquivo ou diretório inexistente
```

ChatGPT

Ao tentar executar o comando `cd usr`, você está recebendo o erro "Arquivo ou diretório inexistente". Isso ocorre porque você precisa especificar o caminho completo para o diretório "usr" a partir do diretório atual.

Por padrão, o diretório "usr" está localizado na raiz do sistema de arquivos, portanto, você pode tentar o seguinte comando para acessá-lo:

```
shell Copy code
cd /usr
```

Certifique-se de incluir a barra (/) no início do caminho para indicar que é um caminho absoluto. Depois de entrar no diretório "/usr", você pode continuar navegando para os subdiretórios desejados, se necessário.

Anonymous

abordagembasica22@abordagembasica22-VirtualBox:usr/local/bin\$ ls

netns vcmd vnoded

ChatGPT

Ótimo! Parece que os executáveis do Core Emulator estão presentes no diretório `/usr/local/bin`. No entanto, o comando `core-gui` ainda não está disponível.

Para resolver isso, verifique se o arquivo `core-gui` está presente no diretório `/usr/local/bin`. Se não estiver lá, pode ser necessário verificar se o Core Emulator foi instalado corretamente ou se existe um problema específico na sua configuração.

Caso o arquivo `core-gui` esteja no diretório `/usr/local/bin`, verifique se ele tem permissões de execução. Para fazer isso, execute o seguinte comando:

```
shell Copy code
ls -l /usr/local/bin/core-gui
```

Isso exibirá as permissões do arquivo. Verifique se a permissão de execução está ativada (representada pelo "x" no resultado do comando). Se não estiver, você pode conceder permissão de execução usando o seguinte comando:

```
shell Copy code
chmod +x /usr/local/bin/core-gui
```

Depois de conceder a permissão de execução, tente novamente executar o comando `core-gui`.

Anonymous

abordagembasica22@abordagembasica22-VirtualBox:usr/local/bin\$ ls -l core-gui

ls: não foi possível acessar 'core-gui': Arquivo ou diretório inexistente

ChatGPT

Se o comando `ls` não consegue encontrar o arquivo `core-gui` no diretório `/usr/local/bin`, isso significa que o arquivo não está presente nesse diretório.

Nesse caso, é possível que o Core Emulator não tenha sido instalado corretamente ou que o arquivo `core-gui` tenha um nome diferente ou esteja localizado em um diretório diferente.

Recomendo revisar o processo de instalação do Core Emulator para garantir que todos os passos tenham sido seguidos corretamente. Verifique também se há alguma documentação ou guia específico fornecido pelo desenvolvedor do Core Emulator que possa oferecer mais informações sobre a instalação e uso do `core-gui`.

Se ainda encontrar dificuldades, sugiro buscar ajuda no fórum ou nas comunidades de suporte do Core Emulator, onde usuários e desenvolvedores podem compartilhar suas experiências e fornecer assistência adicional com problemas de instalação e configuração específicos do software.

Anonymous

como entrar no usuário root

ChatGPT

Para entrar no usuário root no Linux, você pode usar o comando `su` seguido de senha do usuário root. No entanto, é importante ressaltar que o acesso como root concede poderes administrativos completos ao sistema, o que pode ser perigoso se usado incorretamente. Portanto, tenha cuidado ao executar comandos como root.

Aqui está o procedimento para entrar no usuário root:

1. Abra um terminal.
2. Digite o seguinte comando e pressione Enter:

```
Copy code
su
```

3. Será solicitada a senha do usuário root. Digite a senha e pressione Enter. Note que, ao digitar a senha, os caracteres não serão exibidos na tela por motivos de segurança.
4. Se a senha estiver correta, você será autenticado como root e o prompt do terminal será alterado para indicar que você está como root.
5. Agora você pode executar comandos como root no terminal.

Lembre-se de ter cuidado ao usar os privilégios de root e, sempre que possível, execute

apenas comandos que sejam necessários para a tarefa em questão. Após concluir as tarefas como root, é recomendável sair do usuário root executando o comando ``exit`` ou fechando o terminal.

Anonymous
abordagembasica22@abordagembasica22-VirtualBox:~/Área de Trabalho\$ su

Senha:

su: Falha de autenticação

ChatGPT

Se você está enfrentando uma falha de autenticação ao tentar acessar o usuário root usando o comando ``su``, pode ser que a senha esteja incorreta ou que o acesso ao usuário root esteja desabilitado.

Por padrão, o usuário root não tem uma senha definida em algumas distribuições do Linux. Nesses casos, você pode usar o comando ``sudo`` para executar comandos com privilégios de root. O comando ``sudo`` permite que um usuário com permissões sudo execute comandos com privilégios elevados, inserindo a própria senha.

Para usar o comando ``sudo``, siga estas etapas:

1. Abra um terminal.
2. Digite o seguinte comando e pressione Enter:

```
Copy code
sudo comando
```

Substitua "comando" pelo comando específico que você deseja executar como root.

3. Será solicitada a sua senha de usuário.
4. Digite sua senha e pressione Enter. Novamente, os caracteres não serão exibidos na tela por motivos de segurança.
5. Se a senha estiver correta e você tiver permissões sudo, o comando será executado com privilégios de root.

Se o usuário que você está usando tiver permissões sudo configuradas corretamente, você poderá executar comandos com privilégios de root usando o ``sudo``. No entanto, se o acesso ao usuário root estiver desabilitado ou se você não tiver permissões sudo, será necessário verificar as configurações do sistema ou entrar em contato com o administrador do sistema para obter assistência adicional.

Anonymous
agora no usuário root como instalar o core?

ChatGPT

Como você está logado como usuário root, você pode instalar o Core seguindo as etapas abaixo:

1. Certifique-se de ter acesso à internet, pois o Core precisa baixar alguns pacotes durante a instalação.
2. Abra um terminal como usuário root.
3. Navegue até o diretório onde o Core foi compilado. Se você compilou o Core na pasta "~/Área de Trabalho/core", use o seguinte comando para navegar até o diretório:

```
Copy code
bash
cd ~/Área de Trabalho/core
```

4. Execute o comando ``make install`` para instalar o Core:

```
Copy code
go
make install
```

5. Aguarde até que a instalação seja concluída. Isso copiará os arquivos necessários para as localizações adequadas do sistema.
6. Após a instalação, você pode tentar executar o comando ``core-gui`` para iniciar a interface gráfica do Core:

```
Copy code
core-gui
```

Se tudo correr bem, o Core deverá iniciar sem erros.

Lembre-se de que o procedimento de instalação pode variar dependendo da versão do Core que você está usando e das configurações do seu sistema. Se encontrar algum problema durante a instalação, verifique a documentação do Core ou consulte o suporte da

comunidade do Core para obter assistência adicional.

```

Anonymous
abordagembasica22@abordagembasica22-VirtualBox:~/Área de Trabalho$ sudo su

root@abordagembasica22-VirtualBox:/home/abordagembasica22/Área de Trabalho# cd ~/
Área de Trabalho/core

bash: cd: número excessivo de argumentos

root@abordagembasica22-VirtualBox:/home/abordagembasica22/Área de Trabalho# cd
core

root@abordagembasica22-VirtualBox:/home/abordagembasica22/Área de Trabalho/core

# make install

Makefile:984: aviso: ignorando pré-requisitos na definição de regra de sufixo

Making install in daemon

make[1]: Entrando no diretório '/home/abordagembasica22/Área de Trabalho/core/daemon'

Making install in proto

make[2]: Entrando no diretório '/home/abordagembasica22/Área de
Trabalho/core/daemon/proto'

make[3]: Entrando no diretório '/home/abordagembasica22/Área de
Trabalho/core/daemon/proto'

make[3]: Nada a ser feito para 'install-exec-am'.

make[3]: Nada a ser feito para 'install-data-am'.

make[3]: Saindo do diretório '/home/abordagembasica22/Área de
Trabalho/core/daemon/proto'

make[2]: Saindo do diretório '/home/abordagembasica22/Área de
Trabalho/core/daemon/proto'

make[2]: Entrando no diretório '/home/abordagembasica22/Área de Trabalho/core/daemon'

make[3]: Entrando no diretório '/home/abordagembasica22/Área de Trabalho/core/daemon'

make[3]: Nada a ser feito para 'install-exec-am'.

make[3]: Nada a ser feito para 'install-data-am'.

make[3]: Saindo do diretório '/home/abordagembasica22/Área de Trabalho/core/daemon'

make[2]: Saindo do diretório '/home/abordagembasica22/Área de Trabalho/core/daemon'

make[1]: Saindo do diretório '/home/abordagembasica22/Área de Trabalho/core/daemon'

Making install in netns

make[1]: Entrando no diretório '/home/abordagembasica22/Área de Trabalho/core/netns'

make[2]: Entrando no diretório '/home/abordagembasica22/Área de Trabalho/core/netns'

/usr/bin/mkdir -p '/usr/local/bin'

/usr/bin/install -c vncd vcmd netns '/usr/local/bin'

make[2]: Nada a ser feito para 'install-data-am'.

make[2]: Saindo do diretório '/home/abordagembasica22/Área de Trabalho/core/netns'

make[1]: Saindo do diretório '/home/abordagembasica22/Área de Trabalho/core/netns'

make[1]: Entrando no diretório '/home/abordagembasica22/Área de Trabalho/core'

Makefile:984: aviso: ignorando pré-requisitos na definição de regra de sufixo

make[2]: Entrando no diretório '/home/abordagembasica22/Área de Trabalho/core'

Makefile:984: aviso: ignorando pré-requisitos na definição de regra de sufixo

make[2]: Nada a ser feito para 'install-exec-am'.

make[2]: Nada a ser feito para 'install-data-am'.

```

```
make[2]: Saindo do diretório /home/abordagembasica22/Área de Trabalho/core'
```

```
make[1]: Saindo do diretório /home/abordagembasica22/Área de Trabalho/core'
```

ChatGPT

Parece que o comando `make install` foi executado com sucesso. Agora, você pode tentar executar o comando `core-gui` para iniciar a interface gráfica do Core:

```
Copy code
core-gui
```

Certifique-se de estar no diretório correto. Se você estiver no diretório `/home/abordagembasica22/Área de Trabalho/core`, o comando `core-gui` deve funcionar agora. Caso contrário, tente navegar para o diretório correto onde o Core foi instalado antes de executar o comando `core-gui`.

Anonymous

```
root@abordagembasica22-VirtualBox:/home/abordagembasica22/Área de Trabalho/core
```

```
# core-gui
```

```
core-gui: comando não encontrado
```

ChatGPT

Se o comando `core-gui` não está sendo encontrado mesmo após a instalação, pode ser necessário adicionar o diretório onde o Core foi instalado ao seu `$PATH` para que o sistema possa localizá-lo corretamente.

Para adicionar um diretório ao `$PATH`, você pode editar o arquivo `~/.bashrc` ou o arquivo de perfil do shell do usuário root.

Vamos editar o arquivo `~/.bashrc`:

```
bash Copy code
nano ~/.bashrc
```

Isso abrirá o arquivo `~/.bashrc` no editor de texto `nano`. Role até o final do arquivo e adicione a seguinte linha:

```
bash Copy code
export PATH="/usr/local/bin/core-emulator:$PATH"
```

Pressione `Ctrl + O` para salvar as alterações e `Ctrl + X` para sair do editor.

Agora, execute o seguinte comando para recarregar o arquivo `~/.bashrc`:

```
bash Copy code
source ~/.bashrc
```

Depois disso, tente executar novamente o comando `core-gui`:

```
Copy code
core-gui
```

Isso deve permitir que o sistema encontre o comando `core-gui` e execute a interface gráfica do Core.

Anonymous

```
root@abordagembasica22-VirtualBox:/home/abordagembasica22/Área de Trabalho/core
```

```
# core-gui
```

```
core-gui: comando não encontrado
```

ChatGPT

Se mesmo após adicionar o diretório ao `$PATH`, o comando `core-gui` ainda não está sendo encontrado, pode ser necessário verificar se o comando foi instalado corretamente ou se está localizado em um diretório diferente.

Você pode tentar executar o comando `find` para procurar pelo executável `core-gui` no sistema:

```
arduino Copy code
```

```
find / -name core-gui
```

Isso irá procurar em todo o sistema pelo arquivo `core-gui`. Se o comando for encontrado, ele exibirá o caminho completo do arquivo. Caso contrário, pode ser necessário verificar se o Core foi instalado corretamente e se o nome do comando está correto.

Certifique-se de seguir as instruções de instalação do Core para o seu sistema operacional específico e verifique se você está usando o nome do comando corretamente.

Se você ainda estiver enfrentando problemas para executar o `core-gui`, sugiro consultar a documentação oficial do Core ou buscar suporte em fóruns ou comunidades online relacionadas ao Core para obter assistência mais específica em relação ao problema que você está enfrentando.

Anonymous
 root@abordagembasica22-VirtualBox:/home/abordagembasica22/Área de Trabalho/core
 # find / -name core-gui
 find: /run/user/1000/doc: Permissão negada
 find: /run/user/1000/gvfs: Permissão negada
 root@abordagembasica22-VirtualBox:/home/abordagembasica22/Área de Trabalho/core
 #

ChatGPT
 Ao executar o comando `find` como root, é normal que você receba as mensagens de "Permissão negada" para alguns diretórios. Isso ocorre porque o usuário root não tem permissão de acesso a certas áreas restritas do sistema.

No entanto, o resultado do comando `find` foi inconclusivo, pois não encontrou o arquivo `core-gui` em nenhum local acessível. Isso pode indicar que o comando não foi instalado corretamente ou que seu nome difere do esperado.

Recomendo verificar novamente a documentação oficial do Core para garantir que você tenha seguido todos os passos corretamente durante a instalação. Verifique também se o nome do comando é realmente `core-gui` ou se pode ser algo diferente.

Além disso, verifique se o diretório `/usr/local/bin` está listado corretamente no `$PATH` para que o sistema possa encontrar os executáveis instalados lá.

Se todas as etapas foram seguidas corretamente e o comando `core-gui` ainda não está sendo encontrado, sugiro buscar assistência em fóruns ou comunidades relacionadas ao Core, onde os usuários podem compartilhar experiências e oferecer suporte específico para o problema que você está enfrentando.

Anonymous
 como fazer isso Além disso, verifique se o diretório `/usr/local/bin` está listado corretamente no `$PATH` para que o sistema possa encontrar os executáveis instalados lá.

ChatGPT
 Para verificar se o diretório `/usr/local/bin` está corretamente listado no `$PATH`, você pode seguir os seguintes passos:

1. Como root, abra o arquivo `/etc/environment` em um editor de texto, como o nano, por exemplo:

```
bash Copy code
sudo nano /etc/environment
```

1. Verifique se o arquivo contém a seguinte linha:

```
makefile Copy code
PATH="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games
```

1. Se a linha estiver presente, o diretório `/usr/local/bin` está incluído no `$PATH`. Caso contrário, adicione `/usr/local/bin` ao final da linha separado por `:`, de forma que fique assim:

```
makefile Copy code
PATH="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games
```

1. Salve as alterações e feche o arquivo (no nano, pressione `Ctrl + O` para salvar e `Ctrl + X` para sair).

Depois de fazer essas alterações, o sistema deve ser capaz de encontrar os executáveis instalados no diretório `/usr/local/bin`, incluindo o `core-gui`. Tente executar o comando `core-gui` novamente e verifique se ele é encontrado corretamente.

Caso ainda tenha problemas, é recomendado revisar a documentação específica do Core ou buscar suporte na comunidade do Core para obter assistência mais precisa com relação à instalação e configuração do `core-gui`.

Anonymous
GNU nano 6.2 /etc/environment

```
PATH="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:>
```

ChatGPT

Parece que o conteúdo do arquivo `/etc/environment` está incompleto.

Para corrigir, siga as etapas abaixo:

1. Como root, abra o arquivo `/etc/environment` em um editor de texto, como o nano:

```
bash Copy code
sudo nano /etc/environment
```

1. Certifique-se de que a linha `PATH` esteja completa e correta, adicionando `/usr/local/bin` no final. O conteúdo do arquivo deve ser algo parecido com isto:

```
makefile Copy code
PATH="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/gam
```

1. Salve as alterações e feche o arquivo (no nano, pressione `Ctrl + O` para salvar e `Ctrl + X` para sair).
2. Depois de fazer essas alterações, reinicie o sistema para que as alterações no arquivo `/etc/environment` sejam aplicadas.

Após reiniciar, verifique novamente o valor de `$PATH` usando o comando `echo $PATH`. Certifique-se de que `/usr/local/bin` está presente no `$PATH`. Em seguida, tente executar o comando `core-gui` e veja se ele é encontrado corretamente.

Anonymous
abordagembasica22@abordagembasica22-VirtualBox:~/Área de Trabalho\$ echo \$PATH

/caminho/para/o/core-emulador:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/gar

ChatGPT

Aparentemente, o diretório `/usr/local/bin` está corretamente listado no `$PATH`. No entanto, o diretório `/caminho/para/o/core-emulador` também está presente no caminho, o que pode estar causando conflitos. Certifique-se de remover `/caminho/para/o/core-emulador` do `$PATH` para evitar problemas.

Para corrigir isso, siga as etapas abaixo:

1. Como usuário normal, abra o arquivo `~/.bashrc` em um editor de texto, como o nano:

```
bash Copy code
nano ~/.bashrc
```

1. Procure a linha que define o `$PATH` e edite-a para remover `/caminho/para/o/core-emulador`. O conteúdo da linha deve ser algo semelhante a isso:

```
arduino Copy code
export PATH="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/u
```

1. Salve as alterações e feche o arquivo (no nano, pressione `Ctrl + O` para salvar e `Ctrl + X` para sair).
2. Abra um novo terminal ou execute o comando `source ~/.bashrc` para carregar as

alterações do arquivo `.bashrc` no terminal atual.

Depois disso, verifique novamente o valor de `$PATH` usando o comando `echo $PATH` e verifique se `/usr/local/bin` está presente e `/caminho/para/o/core-emulador` foi removido. Em seguida, tente executar o comando `core-gui` e veja se ele é encontrado corretamente.



Anonymous

```
abordagembasica22@abordagembasica22-VirtualBox:~/Área de Trabalho$ echo $PATH
```

```
/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/snap
```



ChatGPT

Ótimo! Agora o diretório `/usr/local/bin` está corretamente listado no seu `$PATH` e o diretório `/caminho/para/o/core-emulador` foi removido.

Agora você pode tentar executar o comando `core-gui` novamente e verificar se ele é encontrado corretamente. Certifique-se de estar no diretório adequado onde o comando `core-gui` está localizado.

Se ainda assim o comando `core-gui` não for encontrado, verifique se o executável `core-gui` está realmente presente no diretório `/usr/local/bin`. Você pode usar o comando `ls -l /usr/local/bin` para listar os arquivos nesse diretório e confirmar se o `core-gui` está lá.

Se o arquivo `core-gui` estiver presente em `/usr/local/bin`, verifique se as permissões de execução estão corretamente configuradas. Você pode usar o comando `ls -l /usr/local/bin/core-gui` para verificar as permissões. Caso necessário, você pode corrigir as permissões usando o comando `chmod +x /usr/local/bin/core-gui`.

Espero que isso ajude a resolver o problema.



Anonymous

```
abordagembasica22@abordagembasica22-VirtualBox:~/Área de Trabalho$ ls -l
```

```
/usr/local/bin
```

```
total 320
```

```
-rwxr-xr-x 1 root root 32344 mai 21 03:26 netns
```

```
-rwxr-xr-x 1 root root 148472 mai 21 03:26 vcmd
```

```
-rwxr-xr-x 1 root root 143176 mai 21 03:26 vnoded
```



ChatGPT

Obrigado por fornecer a saída do comando `ls -l /usr/local/bin`.

Com base nas permissões listadas, parece que os arquivos `netns`, `vcmd` e `vnoded` estão configurados corretamente para execução, com permissões de leitura, gravação e execução para o proprietário (`root`) e para o grupo `root`.

No entanto, não vejo o arquivo `core-gui` na listagem. Isso pode indicar que o `core-gui` não foi instalado corretamente ou foi instalado em um diretório diferente.

Verifique se você executou corretamente o comando `make install` ao instalar o CORE e se não houve erros durante o processo de instalação.

Se você confirmar que o `core-gui` não está presente no diretório `/usr/local/bin`, pode ser necessário investigar onde o `core-gui` foi instalado ou verificar se há algum problema durante a compilação e instalação do CORE.

Certifique-se de seguir as instruções de instalação do CORE corretamente e, se necessário, consulte a documentação do CORE ou o suporte da ferramenta para obter assistência adicional na instalação do `core-gui`.



Anonymous

como fazer investigar onde o core-gui foi instalado ou verificar se há algum problema durante a compilação e instalação do CORE



ChatGPT

Para investigar onde o `core-gui` foi instalado ou verificar problemas durante a compilação e instalação do CORE, você pode seguir estas etapas:

1. Verifique a saída do comando `make install` para identificar o diretório de destino do `core-gui`. O comando `make install` normalmente exibe os caminhos onde os arquivos estão sendo instalados.
2. Execute o comando `find / -name core-gui` como usuário root para procurar o arquivo `core-gui` em todo o sistema. Isso pode levar algum tempo, pois o comando percorrerá

- todos os diretórios. Se o arquivo for encontrado, verifique o caminho onde está localizado.
- Verifique se houve erros durante a compilação e instalação do CORE. Analise cuidadosamente a saída do comando `make` durante o processo de compilação para ver se há quaisquer erros ou mensagens de aviso. Erros durante a compilação podem indicar problemas que podem afetar a instalação do `core-gui`.
 - Consulte a documentação do CORE ou o arquivo README fornecido com o software para obter informações específicas sobre a instalação do `core-gui`. Esses recursos podem fornecer instruções adicionais ou soluções para problemas comuns durante a instalação.
 - Verifique se todas as dependências necessárias foram instaladas corretamente. O CORE pode ter dependências específicas que precisam ser satisfeitas antes da instalação. Consulte a documentação do CORE para obter informações sobre as dependências necessárias e verifique se todas estão presentes.
 - Considere a possibilidade de desinstalar o CORE e repetir o processo de instalação. Às vezes, problemas de instalação podem ocorrer devido a erros anteriores ou conflitos. Realizar uma desinstalação completa e começar novamente pode ajudar a resolver problemas persistentes.

Se mesmo após essas etapas você não conseguir encontrar ou executar o `core-gui`, pode ser útil buscar assistência no fórum de suporte do CORE ou entrar em contato com a equipe de suporte da ferramenta para obter ajuda personalizada e soluções específicas para o seu caso.

Anonymous

abordagembasica22@abordagembasica22-VirtualBox:~/Área de Trabalho\$ sudo su

[sudo] senha para abordagembasica22:

```
root@abordagembasica22-VirtualBox:/home/abordagembasica22/Área de Trabalho# find / -name core-gui
```

```
find: /run/user/1000/doc: Permissão negada
```

```
find: /run/user/1000/gvfs: Permissão negada
```

ChatGPT

O comando `find` está retornando algumas mensagens de permissão negada, o que é normal, pois existem diretórios que o usuário root não possui permissão para acessar. No entanto, ainda é possível investigar onde o `core-gui` foi instalado. Vamos prosseguir com outras opções:

- Verifique se o `core-gui` está instalado em algum dos diretórios listados em `$PATH`. Execute o seguinte comando:

```
bash
echo $PATH
```

Verifique se algum dos diretórios exibidos contém o executável `core-gui`.

- Se você ainda não encontrou o `core-gui`, verifique os seguintes diretórios padrão onde os executáveis são frequentemente instalados:

```
bash
/usr/local/bin
/usr/bin
/bin
```

Execute o comando `ls` em cada um desses diretórios para procurar o `core-gui`:

```
bash
ls /usr/local/bin/core-gui
ls /usr/bin/core-gui
ls /bin/core-gui
```

Se o arquivo for encontrado em algum desses diretórios, anote o caminho para usar posteriormente.

- Caso o `core-gui` não seja encontrado nos locais acima, é possível que a instalação do CORE não tenha incluído o componente `core-gui`. Nesse caso, você pode tentar reinstalar o CORE, prestando atenção especial às opções de instalação para garantir que o `core-gui` seja incluído.
- Por fim, se nenhuma das etapas anteriores resultar na localização do `core-gui`, é recomendado verificar a documentação oficial do CORE ou contatar o suporte do CORE para obter assistência adicional específica para o seu sistema operacional e versão do CORE.

Lembre-se de executar os comandos como usuário root ou usando `sudo` para garantir as permissões adequadas durante a busca.

Anonymous
 root@abordagembasica22-VirtualBox:/home/abordagembasica22/Área de Trabalho# echo \$PATH

```
/usr/local/bin/core-
emulador:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/gar
```

ChatGPT

O comando `echo $PATH` mostra que o diretório `/usr/local/bin` está incluído em seu `$PATH`, o que é correto. No entanto, o `core-gui` ainda não está presente nesse diretório.

Para investigar o problema com a compilação e instalação do CORE, vamos verificar os arquivos de log gerados durante o processo. Normalmente, os logs são armazenados no diretório `/var/log`.

Execute o seguinte comando para listar os arquivos de log relacionados à instalação do CORE:

```
bash Copy code
ls /var/log | grep core
```

Isso retornará os arquivos de log que possuem "core" em seus nomes. Verifique se há algum arquivo que possa fornecer informações sobre a compilação e a instalação do CORE. Preste atenção especial a arquivos como `core-install.log`, `core-compile.log` ou algo semelhante.

Depois de identificar o arquivo de log relevante, você pode visualizá-lo com um editor de texto, como o `nano`, por exemplo:

```
c Copy code
nano /var/log/nome-do-arquivo-de-log
```

Verifique se há erros ou mensagens relacionadas à compilação e à instalação do CORE. Isso pode fornecer pistas sobre o problema e ajudar a resolver a questão.

Se você encontrar erros específicos no arquivo de log ou tiver mais perguntas, sinta-se à vontade para compartilhar o conteúdo do log aqui para que eu possa ajudar de forma mais precisa.

Anonymous
 abordagembasica22@abordagembasica22-VirtualBox:/usr/local/bin\$ ls

```
netns vcmd vnoded
```

ChatGPT

O diretório `/usr/local/bin` contém apenas os executáveis `netns`, `vcmd` e `vnoded`. Não há nenhum arquivo chamado `core-gui` presente nesse diretório.

Isso sugere que o `core-gui` não foi instalado corretamente durante o processo de compilação e instalação do CORE. Pode ter ocorrido algum erro durante a compilação ou pode ser necessário executar etapas adicionais específicas para instalar o `core-gui`.

Recomendo verificar os arquivos de log, como mencionei anteriormente, para obter informações detalhadas sobre quaisquer erros ou problemas encontrados durante a instalação do CORE. Esses arquivos de log podem estar localizados no diretório `/var/log`. Procure por arquivos relacionados à instalação do CORE, como `core-install.log` ou `core-compile.log`, e verifique o conteúdo desses arquivos para obter mais informações.

Se você encontrar erros específicos ou tiver mais perguntas, sinta-se à vontade para compartilhar o conteúdo do arquivo de log aqui, para que eu possa ajudar de forma mais precisa.

Anonymous
 abordagembasica22@abordagembasica22-VirtualBox:~/Área de Trabalho/core\$ ls

```
aclocal.m4      config.h  daemon  Makefile.in  stamp-h1
```

```
ASSIGNMENT_OF_COPYRIGHT.pdf  config.h.in  dockerfiles  man  tasks.py
```

```
autom4te.cache  config.log  docs  netns
```

```
bootstrap.sh    config.status  LICENSE  package
```

```
CHANGELOG.md   configure  Makefile  README.md
```

```
config          configure.ac Makefile.am setup.sh
```

ChatGPT

O diretório `~/home/abordagembasica22/Área de Trabalho/core` contém os arquivos e diretórios relacionados ao CORE. Você pode verificar se há algum problema durante a compilação e instalação do CORE examinando esses arquivos.

Alguns arquivos relevantes incluem:

- `configure` e `configure.ac`: Esses arquivos são usados para configurar e personalizar a compilação do CORE. Verifique se eles foram executados corretamente e se não há erros relatados.
- `Makefile` e `Makefile.in`: Esses arquivos contêm as regras e comandos para compilar e instalar o CORE. Verifique se não há erros ou problemas nessas configurações.
- `config.h` e `config.h.in`: Esses arquivos contêm as definições de configuração do CORE. Verifique se eles foram gerados corretamente e se não há erros ou conflitos de configuração.
- `README.md`, `CHANGELOG.md`, `LICENSE`: Esses arquivos contêm informações sobre o CORE, incluindo instruções de instalação, alterações recentes e termos de licença. Revise-os para obter orientações adicionais sobre a instalação e uso do CORE.

Além disso, verifique se há logs de compilação ou instalação, que podem estar localizados no diretório atual ou em outros diretórios relacionados ao CORE, como `./logs` ou `./build/logs`. Procure por arquivos com extensões `.log` ou nomes que indiquem registros de compilação ou instalação.

Revisando esses arquivos, você pode obter informações mais detalhadas sobre quaisquer problemas ou erros encontrados durante o processo de compilação e instalação do CORE.

Anonymous

com base nesses arquivos como executar o core

ChatGPT

Com base nos arquivos listados, podemos executar o CORE seguindo estas etapas:

1. Certifique-se de estar no diretório correto. No seu caso, parece que você está no diretório `~/home/abordagembasica22/Área de Trabalho/core`.
2. Execute o comando `./core-gui` para iniciar a GUI do CORE. Verifique se o arquivo `core-gui` está presente no diretório atual.

Se o comando acima não funcionar, verifique se o arquivo `core-gui` está presente no diretório atual. Caso não esteja presente, pode haver um problema durante o processo de compilação ou instalação.

Também é possível verificar se há algum script de inicialização ou atalho fornecido no diretório, como `./start.sh` ou `./run.sh`, que pode ser usado para iniciar o CORE.

Certifique-se de que todos os requisitos de sistema e dependências do CORE tenham sido atendidos antes de tentar executá-lo. Consulte a documentação do CORE para obter mais informações sobre os requisitos e etapas de instalação específicas para o seu sistema.

Anonymous

```
# run command passing in the right PATH to pickup from the user running the command
sudo env PATH=$PATH core-daemon
```

```
# add an alias to ~/.bashrc or something similar
alias sudop='sudo env PATH=$PATH'
# now you can run commands like so
sudop core-daemon
```

ChatGPT

As instruções fornecidas mostram duas abordagens para executar o comando `core-daemon` com o caminho (`PATH`) correto. Você pode escolher uma das opções, dependendo da sua preferência.

Opção 1: Executar o comando com `sudo` e definir manualmente o `PATH`:

```
bash Copy code
sudo env PATH=$PATH core-daemon
```

Essa abordagem usa o comando `sudo` para executar o `core-daemon` e define o `PATH` manualmente usando a variável `$PATH` atual.

Opção 2: Adicionar um alias no arquivo `~/.bashrc` (ou outro arquivo de configuração semelhante) para facilitar a execução do comando:

```
bash Copy code
```

```
alias sudop='sudo env PATH=$PATH'
```

Após adicionar o alias, você pode usar o comando `sudo` seguido do nome do comando que deseja executar, por exemplo:

```
Copy code
sudo core-daemon
```

Dessa forma, o comando será executado com o `sudo` e terá o `PATH` correto definido automaticamente.

Lembre-se de reiniciar o terminal ou executar o comando `source ~/.bashrc` para que as alterações no arquivo `~/.bashrc` entrem em vigor.

Anonymous

como fazer isso # can add to ~/.bashrc
export PATH=\$PATH:/opt/core/venv/bin

ChatGPT

Para adicionar a linha `export PATH=$PATH:/opt/core/venv/bin` ao arquivo `~/.bashrc`, siga as etapas abaixo:

1. Abra o arquivo `~/.bashrc` em um editor de texto. Por exemplo, você pode usar o comando `nano`:

```
bash Copy code
nano ~/.bashrc
```

2. Role até o final do arquivo `~/.bashrc`.
3. Adicione a linha `export PATH=$PATH:/opt/core/venv/bin` no final do arquivo.
4. Pressione `Ctrl + O` para salvar as alterações no arquivo `~/.bashrc`.
5. Pressione `Enter` para confirmar o nome do arquivo.
6. Pressione `Ctrl + X` para sair do editor `nano`.
7. Execute o comando a seguir para aplicar as alterações no arquivo `~/.bashrc`:

```
bash Copy code
source ~/.bashrc
```

Após essas etapas, o diretório `/opt/core/venv/bin` será adicionado ao `PATH` sempre que você abrir um novo terminal ou iniciar uma nova sessão. Isso permitirá que você execute comandos localizados nesse diretório diretamente, sem especificar o caminho completo.

Anonymous

como executar isso # run command passing in the right PATH to pickup from the user running the command
sudo env PATH=\$PATH core-daemon

```
# add an alias to ~/.bashrc or something similar
alias sudop='sudo env PATH=$PATH'
# now you can run commands like so
sudop core-daemon
```

ChatGPT

Para executar o comando `sudo env PATH=$PATH core-daemon`, você pode simplesmente digitá-lo no terminal e pressionar Enter. Esse comando garante que o `core-daemon` seja executado com o `PATH` correto.

Se você deseja adicionar um alias ao arquivo `~/.bashrc` para simplificar o uso do comando, siga as etapas abaixo:

1. Abra o arquivo `~/.bashrc` em um editor de texto. Por exemplo, você pode usar o comando `nano`:

```
bash Copy code
nano ~/.bashrc
```

2. Role até o final do arquivo `~/.bashrc`.
3. Adicione a linha a seguir para definir o alias:

```
bash Copy code
alias sudop='sudo env PATH=$PATH'
```

4. Pressione `Ctrl + O` para salvar as alterações no arquivo `~/.bashrc`.
5. Pressione `Enter` para confirmar o nome do arquivo.
6. Pressione `Ctrl + X` para sair do editor `nano`.

- f. execute o comando a seguir para aplicar as alterações no arquivo `~/.bashrc` :

```
bash
source ~/.bashrc
```

Agora você pode usar o alias `sudo` para executar comandos com `sudo` e manter o `PATH` do usuário original. Por exemplo, para executar `core-daemon` com o `sudo`, você pode usar o seguinte comando:

```
Copy code
sudo core-daemon
```

Isso garantirá que o comando seja executado com o `PATH` correto do usuário atual.

Anonymous
abordagembasica22@abordagembasica22-VirtualBox:~/Área de Trabalho\$ sudo core-daemon

env: 'core-daemon': Arquivo ou diretório inexistente

ChatGPT

Se o comando `core-daemon` não está sendo reconhecido como um executável válido, pode haver um problema na instalação ou na configuração do CORE. Verifique se o comando `core-daemon` está disponível no diretório em que o CORE foi instalado.

Além disso, certifique-se de que o CORE esteja corretamente instalado e configurado em seu sistema. Verifique se todos os passos de instalação foram seguidos corretamente e se não ocorreram erros durante o processo.

Se você ainda encontrar dificuldades, recomendo consultar a documentação oficial do CORE ou a comunidade de suporte do CORE para obter assistência adicional.

Anonymous
como desinstalar o core totalmente do pc

ChatGPT

Para desinstalar completamente o CORE do seu PC, você precisará remover todos os arquivos e componentes relacionados ao CORE. Os passos a seguir podem variar dependendo do método de instalação usado. Aqui estão algumas etapas gerais que você pode seguir:

1. Remova os arquivos do CORE: Execute o seguinte comando para excluir os arquivos principais do CORE:

```
bash
sudo rm -rf /opt/core
```

1. Remova o diretório de configuração: Execute o seguinte comando para remover o diretório de configuração do CORE:

```
bash
sudo rm -rf ~/.core
```

1. Remova as entradas do PATH: Se você adicionou manualmente as entradas do CORE ao seu arquivo `~/.bashrc` ou `~/.profile`, você pode removê-las editando o arquivo e excluindo as linhas relevantes.
2. Remova os pacotes instalados: Se você instalou o CORE por meio de pacotes de gerenciadores de pacotes, como `apt` ou `yum`, você precisará usar o comando correspondente para remover os pacotes instalados. Por exemplo, se você usou o `apt`, execute o seguinte comando:

```
arduino
sudo apt remove core-emulator
```

1. Limpe as dependências: Você pode executar o seguinte comando para limpar as dependências não utilizadas (caso esteja usando o `apt`):

```
Copy code
sudo apt autoremove
```

Lembre-se de que essas etapas são genéricas e podem variar dependendo do método de instalação específico que você usou para o CORE. Certifique-se de adaptar os comandos de

acordo com a sua instalação.

Anonymous
como fazer certifique-se de ter python3.9+ com suporte venv

ChatGPT
Para verificar se você tem o Python 3.9+ instalado e suporte a venv, você pode seguir os seguintes passos:

1. Verifique a versão do Python: Abra o terminal e execute o seguinte comando para verificar a versão do Python instalada:

```
css
python3 --version
```

Isso deve exibir a versão do Python instalada no seu sistema. Verifique se a versão é 3.9 ou superior.

1. Verifique se o módulo venv está disponível: A partir do Python 3.3, o módulo `venv` é fornecido como parte da biblioteca padrão do Python. Para verificar se você tem suporte a venv, execute o seguinte comando:

```
bash
python3 -m venv --help
```

Se o comando exibir informações sobre o uso do `venv`, significa que você tem suporte a venv instalado.

Se você não tem o Python 3.9+ instalado ou não possui suporte a venv, você pode considerar atualizar a versão do Python ou instalá-la manualmente. Dependendo do seu sistema operacional, existem diferentes métodos para instalar o Python. Por exemplo, no Ubuntu, você pode usar o gerenciador de pacotes `apt` para instalar o Python 3.9. Execute o seguinte comando:

```
Copy code
sudo apt install python3.9
```

Certifique-se de pesquisar a documentação oficial do Python para obter instruções detalhadas sobre a instalação em seu sistema operacional específico.

Após verificar se você tem o Python 3.9+ e suporte a venv, você pode prosseguir com a instalação do CORE ou qualquer outra aplicação que dependa desses requisitos.

Anonymous
abordagembasica22@abordagembasica22-VirtualBox:~/Área de Trabalho\$ python3 --version

Python 3.10.6

abordagembasica22@abordagembasica22-VirtualBox:~/Área de Trabalho\$ python3 -m venv --help

```
usage: venv [-h] [--system-site-packages] [--symlinks | --copies] [--clear]
            [--upgrade] [--without-pip] [--prompt PROMPT] [--upgrade-deps]
            ENV_DIR [ENV_DIR ...]
```

Creates virtual Python environments in one or more target directories.

positional arguments:

ENV_DIR A directory to create the environment in.

options:

-h, --help show this help message and exit

--system-site-packages

Give the virtual environment access to the system

```

site-packages dir.

--symlinks    Try to use symlinks rather than copies, when symlinks
              are not the default for the platform.

--copies     Try to use copies rather than symlinks, even when
              symlinks are the default for the platform.

--clear      Delete the contents of the environment directory if
              it already exists, before environment creation.

--upgrade    Upgrade the environment directory to use this version
              of Python, assuming Python has been upgraded in-
              place.

--without-pip Skips installing or upgrading pip in the virtual
              environment (pip is bootstrapped by default)

--prompt PROMPT Provides an alternative prompt prefix for this
              environment.

--upgrade-deps Upgrade core dependencies: pip setuptools to the
              latest version in PyPI

```

Once an environment has been created, you may wish to activate it, e.g. by sourcing an activate script in its bin directory.

ChatGPT

Ótimo! Com base nas informações fornecidas, você tem o Python 3.10.6 instalado e o suporte a venv está disponível. Agora você pode prosseguir com a instalação do CORE ou qualquer outra aplicação que dependa desses requisitos.

Para criar um ambiente virtual e instalar o CORE, você pode seguir estes passos:

1. Navegue até o diretório onde deseja criar o ambiente virtual. Por exemplo, se você deseja criá-lo na área de trabalho, execute o seguinte comando no terminal:

```

bash Copy code
cd ~/Área\ de\ Trabalho

```

1. Crie o ambiente virtual executando o seguinte comando:

```

Copy code
python3 -m venv core-env

```

Isso criará um novo diretório chamado `core-env`, que conterá o ambiente virtual.

1. Ative o ambiente virtual executando o seguinte comando:

```

bash Copy code
source core-env/bin/activate

```

Isso ativará o ambiente virtual e você verá o prefixo `(core-env)` no seu prompt de comando.

1. Agora você está dentro do ambiente virtual. Você pode prosseguir com a instalação do CORE. Certifique-se de ter o pacote de instalação do CORE pronto para ser instalado. Suponha que você tenha um arquivo chamado `core-package.tar.gz` no seu diretório atual. Execute o seguinte comando para instalar o CORE:

```

go Copy code
pip install core-package.tar.gz

```

Substitua `core-package.tar.gz` pelo nome real do arquivo de pacote do CORE, se for

diferente.

1. Aguarde até que o processo de instalação seja concluído. Após a conclusão, o CORE deve estar instalado no seu ambiente virtual.

Lembre-se de que sempre que você quiser usar o CORE, precisará ativar o ambiente virtual executando o comando `source core-env/bin/activate` novamente.

Se você deseja remover completamente o CORE e o ambiente virtual, basta excluir o diretório `core-env` que foi criado. Certifique-se de que você não precise mais do CORE antes de excluir o ambiente virtual.

Anonymous
(core-env) abordagembasica22@abordagembasica22-VirtualBox:~/Área de Trabalho\$ pip install core-package.tar.gz

WARNING: Requirement 'core-package.tar.gz' looks like a filename, but the file does not exist

Processing ./core-package.tar.gz

ERROR: Could not install packages due to an OSError: [Errno 2] Arquivo ou diretório inexistente: '/home/abordagembasica22/Área de Trabalho/core-package.tar.gz'

ChatGPT
Peço desculpas pelo equívoco. Parece que o arquivo `core-package.tar.gz` não está localizado no diretório atual ou não foi nomeado corretamente. Certifique-se de que o arquivo de pacote do CORE esteja presente no diretório atual e forneça o nome correto ao executar o comando `pip install`.

Verifique se o arquivo `core-package.tar.gz` está no diretório atual usando o comando `ls`:

```
bash Copy code
ls
```

Certifique-se de que o arquivo esteja presente e que o nome esteja correto. Em seguida, execute o comando `pip install` substituindo `core-package.tar.gz` pelo nome correto do arquivo:

```
go Copy code
pip install core-package.tar.gz
```

Certifique-se de fornecer o nome exato do arquivo, incluindo a extensão `.tar.gz`. Caso o arquivo esteja em um diretório diferente, navegue até o diretório correto antes de executar o comando `pip install`.

Se o arquivo não estiver presente ou se você estiver enfrentando problemas para localizá-lo, verifique novamente os passos de instalação do CORE para garantir que você tenha baixado o pacote corretamente antes de prosseguir com a instalação.

Anonymous
(core-env) abordagembasica22@abordagembasica22-VirtualBox:~/Área de Trabalho\$ ls

core core-env

ChatGPT
Obrigado por fornecer a lista de arquivos no diretório. Parece que o arquivo `core-package.tar.gz` não está presente. Certifique-se de ter baixado o pacote `core-package.tar.gz` corretamente antes de prosseguir com a instalação.

Se você já tiver o arquivo `core-package.tar.gz`, verifique se ele está no diretório atual. Caso contrário, certifique-se de fazer o download do pacote novamente para o diretório correto.

Após garantir que o arquivo `core-package.tar.gz` está presente no diretório atual, você pode instalar o pacote usando o comando `pip install`, como mencionado anteriormente:

```
go Copy code
pip install core-package.tar.gz
```

Certifique-se de fornecer o nome correto do arquivo, incluindo a extensão `.tar.gz`, ao executar o comando `pip install`.

Anonymous
(core-env) abordagembasica22@abordagembasica22-VirtualBox:~/Área de Trabalho/co

```
re-env$ ls
```

```
bin include lib lib64 pyenv.cfg
```

(core-env) abordagembasica22@abordage

ChatGPT

Parece que você está no ambiente virtual `core-env` e dentro do diretório `~/Área de Trabalho/core-env`. Nesse caso, o arquivo `core-package.tar.gz` precisa estar presente dentro desse diretório para que você possa instalá-lo.

Certifique-se de que o arquivo `core-package.tar.gz` esteja localizado dentro do diretório `~/Área de Trabalho/core-env`. Se o arquivo estiver em outro local, mova-o para esse diretório usando o comando `mv`:

```
bash Copy code
mv /caminho/para/o/core-package.tar.gz ~/Área de Trabalho/core-env/
```

Após mover o arquivo para o diretório correto, você pode prosseguir com a instalação usando o comando `pip install`:

```
go Copy code
pip install core-package.tar.gz
```

Certifique-se de fornecer o nome correto do arquivo, incluindo a extensão `.tar.gz`, ao executar o comando `pip install`.

Anonymous
não tenho o arquivo como baixar core-package.tar.gz

ChatGPT

Peço desculpas pela confusão. Se você não possui o arquivo `core-package.tar.gz`, não é possível instalá-lo diretamente.

Para baixar o pacote `core` corretamente, você precisará seguir as instruções de instalação específicas do `core`. Normalmente, os pacotes `core` estão disponíveis para download em um repositório ou site oficial.

Aqui estão os passos gerais para baixar e instalar o `core`:

1. Verifique a documentação oficial do `core` ou o site do projeto para obter informações sobre como baixar o pacote. Geralmente, você encontrará um link de download ou instruções específicas para a instalação.
2. Acesse o link fornecido e baixe o arquivo `core-package.tar.gz` para o seu computador.
3. Navegue até o diretório onde o arquivo `core-package.tar.gz` foi baixado.
4. Siga as instruções de instalação fornecidas na documentação oficial do `core`. Isso pode envolver a execução de comandos como `tar` para extrair o conteúdo do pacote e, em seguida, executar comandos adicionais para configurar e instalar o `core`.

Certifique-se de seguir as instruções fornecidas pela equipe do `core` para garantir uma instalação correta e bem-sucedida.

Se você estiver enfrentando dificuldades específicas durante o processo de instalação, pode ser útil consultar a documentação oficial do `core` ou buscar suporte na comunidade do projeto para obter assistência mais detalhada.

Anonymous
como fazer Isso instalará vncd/vcmd, dependências do sistema e CORE em um ambiente virtual python em /opt/core/venv.

```
sudo <yum/apt> install -y ./<package>
```

Upgrade to Plus to create your own GPT

[Report content](#) | [Terms of use](#) | [Privacy policy](#) | [Manage cookies](#)



Criando canvas no core-gui.

December 21, 2023

- Anonymous**
 abordagembasica22@abordagembasica22-VirtualBox:~/Área de Trabalho\$ core-gui
 2023-05-21 04:14:17,874 - INFO - manager:add_canvas - creating canvas()
- ChatGPT**
 Based on the provided information, it seems like you are running a command in a terminal or command prompt.
 The command `core-gui` appears to be executed, but without further context, it is difficult to determine its specific purpose or function. It may be a command related to a specific software or tool installed on your system.
 The output `2023-05-21 04:14:17,874 - INFO - manager:add_canvas - creating canvas(1)` is an informational message generated by the command. It suggests that a canvas is being created, possibly as part of the graphical user interface (GUI) of the software or tool you are using.
 If you have any specific questions or need further assistance, please provide more details about the software or tool you are using and the problem you are facing.
- Anonymous**
 em portugues
- ChatGPT**
 Com base nas informações fornecidas, parece que você está executando um comando em um terminal ou prompt de comando.
 O comando `core-gui` parece estar sendo executado, mas sem mais contexto, é difícil determinar seu propósito ou função específica. Pode ser um comando relacionado a um software ou ferramenta específica instalada em seu sistema.
 A saída `2023-05-21 04:14:17,874 - INFO - manager:add_canvas - creating canvas(1)` é uma mensagem informativa gerada pelo comando. Isso sugere que um "canvas" está sendo criado, possivelmente como parte da interface gráfica do usuário (GUI) do software ou ferramenta que você está usando.
 Se você tiver alguma pergunta específica ou precisar de mais assistência, forneça mais detalhes sobre o software ou ferramenta que está usando e o problema que está enfrentando. Estou aqui para ajudar!
- Anonymous**
 gostaria de executar de forma grafica
- ChatGPT**
 Se você deseja executar o comando de forma gráfica, geralmente é necessário um ambiente de desktop ou uma interface gráfica instalada no seu sistema.
 No exemplo que você deu, parece que você está executando o comando em um ambiente de linha de comando ou terminal. Para executar de forma gráfica, você pode procurar pelo programa ou aplicativo correspondente em seu ambiente de desktop ou usar um gerenciador de arquivos para navegar até o local do programa e clicar duas vezes nele para iniciá-lo.
 Se você estiver usando um ambiente de desktop como o GNOME, KDE, Unity ou XFCE, você pode pesquisar pelo programa na barra de pesquisa de aplicativos ou navegar pelo menu de aplicativos para encontrá-lo.
 Caso você precise de mais assistência, por favor, forneça mais informações sobre o programa que você está tentando executar e qual sistema operacional e ambiente de desktop você está usando.
- Anonymous**
 executar o core com interface grafica
- ChatGPT**
 Se você deseja executar o Core com interface gráfica, você precisa ter certeza de que possui a versão correta do Core instalada em seu sistema. O Core é um simulador de redes de código aberto e, por padrão, é executado em linha de comando. No entanto, existem algumas interfaces gráficas disponíveis que fornecem uma experiência visual para o Core.

Uma das opções é utilizar o CORE-GUI, que é a interface gráfica oficial do Core. Certifique-se de ter o pacote `core-gui` instalado em seu sistema. Dependendo da distribuição Linux que você está usando, você pode instalar o pacote com o seguinte comando:

```
bash Copy code  
sudo apt-get install core-gui
```

Após a instalação, você pode iniciar o CORE-GUI digitando `core-gui` em seu terminal ou encontrando o aplicativo "CORE-GUI" em seu menu de aplicativos. Isso abrirá a interface gráfica do Core, onde você poderá criar e configurar suas redes.

Outra opção é usar o CORE-NG, uma nova versão do Core que também possui uma interface gráfica. Você pode encontrar mais informações e instruções de instalação no site oficial do CORE-NG.

Lembre-se de que a disponibilidade e os métodos de instalação podem variar dependendo do seu sistema operacional e distribuição Linux específica. Certifique-se de consultar a documentação relevante para obter instruções mais detalhadas.

[Upgrade to Plus to create your own GPT](#)

[Report content](#) | [Terms of use](#) | [Privacy policy](#) | [Manage cookies](#)



Instalar Wireshark Ubuntu 22

December 21, 2023

Anonymous
instalar o wireshark ubuntu 22

ChatGPT
Para instalar o Wireshark no Ubuntu 22.04 (ou qualquer outra versão mais recente), você pode seguir as seguintes etapas:

1. Abra um terminal no Ubuntu. Você pode fazer isso pressionando `Ctrl + Alt + T` simultaneamente ou pesquisando por "Terminal" no menu.
2. Atualize a lista de pacotes disponíveis executando o seguinte comando no terminal:

```
ssh
Copy code
sudo apt update
```

1. Após a atualização, você pode instalar o Wireshark executando o seguinte comando:

```
Copy code
sudo apt install wireshark
```

Durante o processo de instalação, você será solicitado a confirmar a instalação e digitar sua senha de administrador.

1. Durante a instalação, você também será perguntado se deseja permitir que usuários comuns capturem pacotes. Selecione "Sim" para permitir.
2. Após a conclusão da instalação, você precisará adicionar seu usuário ao grupo "wireshark" para permitir a captura de pacotes. Execute o seguinte comando:

```
tssh
Copy code
sudo usermod -aG wireshark $USER
```

Observe que você precisará reiniciar a sessão ou fazer logout e login novamente para que as alterações tenham efeito.

1. Agora você pode iniciar o Wireshark digitando `wireshark` no terminal ou pesquisando por "Wireshark" no menu de aplicativos.

Lembre-se de que a instalação do Wireshark requer privilégios administrativos (sudo) e a captura de pacotes também exigirá privilégios elevados. Portanto, certifique-se de tomar as devidas precauções ao usar o Wireshark e capturar dados de rede.

Anonymous
agora que tenho o core emulator e o wireshark rodando gostaria de um passo a passo de como gerar uma rede, gerar um teste nessa rede, capturar os pacotes com o wireshark e um passo a passo de como analisar este teste


ChatGPT
Para gerar uma rede, executar um teste e capturar pacotes usando o Wireshark, você pode seguir os passos abaixo:


1. Configurando a rede:
 - * Inicie o CORE Emulator no Ubuntu.
 - * Crie uma nova topologia de rede dentro do CORE Emulator.
 - * Adicione os dispositivos de rede necessários, como roteadores, switches, computadores, etc.
 - * Conecte os dispositivos de acordo com a topologia desejada, configurando as interfaces de rede.
2. Configurando os testes:
 - * Acesse os dispositivos envolvidos na rede (por exemplo, computadores ou roteadores) dentro do CORE Emulator.
 - * Configure as interfaces de rede dos dispositivos com os endereços IP e outras configurações necessárias para a comunicação.
3. Capturando pacotes com o Wireshark:
 - * No Ubuntu, abra o Wireshark pelo terminal digitando `wireshark` e pressionando Enter.
 - * Selecione a interface de rede apropriada para capturar pacotes. Por exemplo, se você estiver usando uma conexão Ethernet, escolha a interface "eth0".
 - * Clique no botão "Iniciar Captura" ou pressione `Ctrl + E` para iniciar a captura de

pacotes.

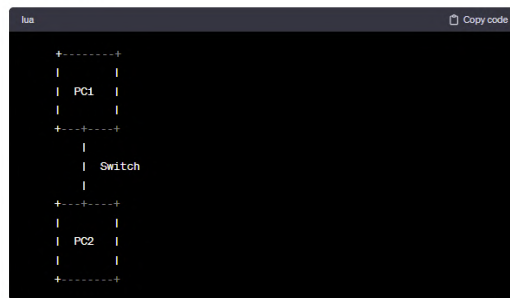
4. Executando o teste na rede:
 - Inicie a simulação ou o teste na rede dentro do CORE Emulador. Isso pode envolver a geração de tráfego de rede entre os dispositivos ou a execução de comandos específicos para testar a conectividade ou o desempenho.
5. Analisando os pacotes capturados:
 - No Wireshark, você verá a captura de pacotes em tempo real. Você pode interromper a captura clicando no botão "Parar Captura" quando desejar.
 - Para analisar os pacotes capturados, você pode usar os filtros do Wireshark para visualizar pacotes específicos, como filtrar por endereços IP de origem ou destino, portas, protocolos, etc.
 - O Wireshark exibirá informações detalhadas sobre cada pacote capturado, como endereços MAC, endereços IP, portas, dados transmitidos e muito mais. Você pode explorar essas informações para entender o tráfego de rede e solucionar problemas.

Tenha em mente que a configuração da rede, a geração de testes e a análise de pacotes podem ser bastante complexas e variam dependendo dos requisitos específicos da sua rede e teste. Portanto, esses passos fornecem apenas uma visão geral do processo geral. Você pode precisar se aprofundar em aspectos específicos da configuração e análise de pacotes, dependendo do seu cenário.

 **Anonymous**
me de um exemplo basico de rede que posso gerar um teste

 **ChatGPT**
Claro! Vou fornecer um exemplo básico de configuração de rede em que você pode gerar um teste simples usando o CORE Emulador e capturar os pacotes com o Wireshark.

Topologia da rede:





Passo a passo:

1. Inicie o CORE Emulador no Ubuntu.
2. Crie uma nova topologia de rede no CORE Emulador e adicione dois computadores (PC1 e PC2).
3. Conecte os dispositivos criando uma conexão entre PC1 e PC2 através de um switch.
4. Configure as interfaces de rede nos computadores:
 - Acesse as configurações do PC1 e defina o endereço IP da interface de rede como `192.168.0.1/24`.
 - Acesse as configurações do PC2 e defina o endereço IP da interface de rede como `192.168.0.2/24`.
5. Abra o Wireshark no Ubuntu.
6. Selecione a interface de rede apropriada para capturar pacotes. Por exemplo, se você estiver usando uma conexão Ethernet, escolha a interface "eth0".
7. Clique no botão "Iniciar Captura" ou pressione `Ctrl + E` para iniciar a captura de pacotes.
8. No PC1, abra um terminal e execute o comando `ping 192.168.0.2` para enviar pacotes ICMP para o PC2.
9. Observe a captura de pacotes no Wireshark em tempo real.
10. Você pode interromper a captura clicando no botão "Parar Captura" quando desejar.
11. Use os recursos do Wireshark para analisar os pacotes capturados. Você pode filtrar por endereços IP, protocolos, portas ou qualquer outro critério relevante para o teste.

Nesse exemplo, o teste consiste em enviar pacotes ICMP do PC1 para o PC2 e capturar esses pacotes usando o Wireshark. Você pode explorar diferentes tipos de testes, como testes de conectividade, testes de desempenho ou testes de tráfego específico, dependendo dos seus objetivos e requisitos.

Lembre-se de adaptar as configurações de rede e os comandos conforme necessário, com base em suas necessidades específicas.

 **Anonymous**
com a rede anterior como ficaria um teste de desempenho

 **ChatGPT**
Para realizar um teste de desempenho simples na rede anterior, você pode usar ferramentas como o iperf ou o hping para gerar tráfego de rede e avaliar a taxa de transferência entre os computadores PC1 e PC2.

Aqui está um exemplo de teste de desempenho usando o iperf:

1. Certifique-se de ter o iperf instalado nos computadores PC1 e PC2. Caso não esteja instalado, você pode instalar executando o seguinte comando em cada computador:

```
Copy code  
sudo apt install iperf
```

1. Abra dois terminais, um no PC1 e outro no PC2.
2. No terminal do PC2, execute o seguinte comando para iniciar o servidor iperf:

```
Copy code  
iperf -s
```

Isso colocará o PC2 em modo de escuta para receber a conexão do PC1.

1. No terminal do PC1, execute o seguinte comando para iniciar o teste de desempenho enviando tráfego para o PC2:

```
r Copy code  
iperf -c 192.168.0.2
```

Isso fará com que o PC1 envie tráfego para o endereço IP do PC2.

1. O iperf fornecerá informações sobre a taxa de transferência alcançada durante o teste, como a velocidade de transferência em bits por segundo.

Você pode ajustar os parâmetros do iperf para personalizar o teste, como a duração do teste, o tamanho dos pacotes, o número de conexões simultâneas, etc. Consulte a documentação do iperf para obter mais informações sobre as opções disponíveis.

Após executar o teste de desempenho, você pode usar o Wireshark para capturar e analisar os pacotes durante a transmissão. Isso permitirá que você verifique detalhes adicionais sobre o tráfego, como protocolos utilizados, tamanho dos pacotes, tempos de resposta, etc.

Lembre-se de adaptar os endereços IP e outras configurações de acordo com sua topologia de rede e requisitos específicos.

[Upgrade to Plus to create your own GPT](#)

[Report content](#) | [Terms of use](#) | [Privacy policy](#) | [Manage cookies](#)

