

Interpretador de Comandos

SISTEMAS OPERATIVOS
IADE-ENGENHARIA INFORMÁTICA

Camila Madatali | 20210832

Gonçalo Alves | 20210544

Tatiana Yermachenkava | 20210697

<https://github.com/camilamadatali/g03-projeto>

Conteúdo

Descrição do problema	2
Casos de Uso	2
Descrição da solução a implementar.....	2
Descrição genérica.....	3
Enquadramento nas áreas da Unidade Curricular	3
Requisitos Técnicos para desenvolvimento do projeto	3
Arquitetura de solução	4
Tecnologias a utilizar	4
Planeamento e calendarização	4
Bibliografia	5

Descrição do problema

O projeto visa desenvolver um interpretador de comandos para o sistema operativo Linux, incorporando características inovadoras para a melhor eficiência e a usabilidade. A motivação para este trabalho encontra-se em possibilitar aos utilizadores uma experiência mais intuitiva e eficaz no uso do terminal, estabelecendo uma ligação entre comandos internos e externos de forma dinâmica.

Casos de Uso

Um Interpretador de Comandos para Consulta de Condições Meteorológicas: a Sara pretende ir dar um passeio à tarde, mas ao ver que o tempo está acinzentado fica em dúvida se vai passear ou não, porque não quer apanhar chuva. Para tal, decide ir ao seu terminal Linux e verificar como vai estar a meteorologia daquela tarde, digitando um comando para chamar a API do IPMA no seu terminal, e após os dados processados a Sara vê que vai estar sol.

Um Interpretador de Comandos para Gestão de Tarefas e Lembretes: o Luís é um utilizador ativo do terminal Linux e, para aumentar a sua produtividade, gostava de poder organizar as suas tarefas e receber lembretes diretamente do terminal. Por isso, o Luís cria a sua tarefa a partir do comando 'criar_tarefa' no seu terminal e completa os parâmetros como a descrição, data-limite e prioridade. A seguir, o Luís recebe um lembrete que tem uma tarefa que está prestes a expirar, com isto escreve 'listar_tarefas' no terminal para visualizar o estado das suas tarefas.

Um Interpretador de Comandos para criptografar arquivos: a Alice pretende armazenar documentos sensíveis no seu computador, mas está preocupada com a segurança destes. Para tal, decide usar o comando de criptografia de arquivos disponível no shell para proteger os dados contra acesso não autorizado. Para isso apenas abre o terminal, chega até ao diretório onde se encontra o arquivo, executa o comando fornecendo o nome do arquivo que deseja criptografar e uma chave de criptografia. Após este processo a Alice tem o seu arquivo criptografado e seguro.

Descrição da solução a implementar

Consulta de Meteorologia: O interpretador irá fornecer informações meteorológicas diretamente no terminal do utilizador, consoante a localização desejada, e de forma a destacar dados essenciais, como a temperatura e a previsão dos próximos dias. Tudo isto, é possível utilizando uma API de meteorologia, como a IPMA.

Gestão de Tarefas e Lembretes: Um interpretador será desenvolvido para permitir aos utilizadores criar, visualizar e gerir tarefas e lembretes diretamente do terminal. Comandos internos específicos, como "criar_tarefa" e "listar_tarefas", serão incorporados para facilitar o processo de gerenciamento de tarefas, adicionar detalhes, e lembrar o utilizador a partir de lembretes.

Criptografia de Arquivos: Um outro interpretador será implementado para oferecer aos utilizadores a capacidade de criptografar e descriptografar arquivos sensíveis diretamente do terminal. Este comando permitirá que os utilizadores protejam os seus arquivos contra acesso não autorizado, fornecendo uma chave de criptografia. O processo de descriptografia será reversível, permitindo aos utilizadores a recuperação dos seus arquivos originais sempre que for necessário. Os utilizadores poderão especificar os arquivos de entrada e uma chave de criptografia.

Descrição genérica

O projeto visa desenvolver interpretadores de comandos que executam diversas funções, que vá além das funcionalidades tradicionais no sistema operativo Linux. Para a melhor interação entre utilizador e terminal decidimos incorporar o interpretador que será capaz de realizar tarefas específicas, como a consulta de meteorologia, gestão de tarefas e lembretes pelo terminal e criptografia de arquivos.

Enquadramento nas áreas da Unidade Curricular

Este projeto está alinhado com os conceitos abordados na unidade curricular de sistemas operativos. Com a implementação de interpretadores de comandos serão necessários conhecimentos em programação python para desenvolver os scripts, gestão de processos, manipulação de APIs, comunicação de rede, processamento de dados, interação utilizador-sistema e desenvolvimento de interfaces de utilizador.

Requisitos Técnicos para desenvolvimento do projeto

- Utilização da linguagem de script de Shell: Bash, para desenvolver os comandos necessários.
- Uso do editor de texto nano para escrever e editar o script de Shell.
- Disponibilidade de um sistema Linux para desenvolvimento, testes e implantação do script de Shell.
- Acesso à API de meteorologia IPMA e obtenção de uma chave de API válida para autenticação.
- Conhecimento sobre requisições HTTP e manipulação de respostas JSON para interagir com a API.

- Habilidades em manipulação de estrutura de dados para armazenamento e gestão de tarefas.

Arquitetura de solução

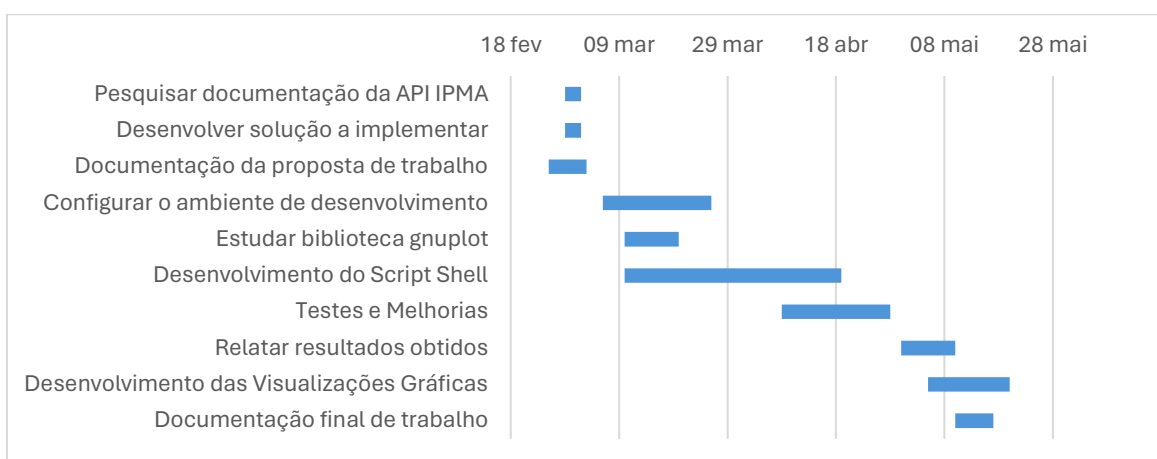


O desenvolvimento de um Shell num sistema Linux começa por uma análise léxica, onde recebe o comando do utilizador para depois ser dividido em tokens devidamente classificados. De seguida na análise sintática são recebidos os tokens, e é verificado se pertencem à linguagem (Shell), para que seja criada uma árvore sintática com base nas regras gramaticais de Shell, analisando a semântica. Por fim, o interpretador é implementado para correr a árvore sintática, interpretar os comandos e executar as ações correspondentes, neste caso, chamar a API IPMA, atualizar tarefas, e criptografar e descriptografar.

Tecnologias a utilizar

- Linguagem de programação python para o desenvolvimento do script.
- Linguagem de script de shell (Bash) para iteração do utilizador com a linha de comandos.
- Conhecimento sobre requisições HTTP e manipulação de JSON para interagir com a API de meteorologia.
- Acesso à API de meteorologia IPMA e obtenção de uma chave de API válida.
- Terminal Linux para testes e implementação.
- Estrutura de dados eficientes para armazenar e gerir tarefas no interpretador.

Planeamento e calendarização



Bibliografia

- Install, L. P. (2016, January 27). *Shell, Bash e Scripts: Tudo sobre Shell Scripting*. Desde Linux.
<https://blog.desdelinux.net/pt/shell-bash-e-scripts-sobre-scripts-de-shell/>
- Comandos do bash: ls, head, mv e cat explicados e com exemplos. (2023, November 14).
 FreeCodeCamp.org. <https://www.freecodecamp.org/portuguese/news/comandos-do-bash-ls-head-mv-e-cat-explicados-e-com-exemplos/>
- IPMA - api. (n.d.). Api.ipma.pt. <https://api.ipma.pt/>
- Weather from terminal. (n.d.). Ask Ubuntu. Retrieved March 2, 2024, from
<https://askubuntu.com/questions/390329/weather-from-terminal>
- OpenAI. (2022, November 30). *ChatGPT*. Chat.openai.com; OpenAI. <https://chat.openai.com/>
- Draw.io. (2024). *Flowchart Maker & Online Diagram Software*. App.diagrams.net.
<https://app.diagrams.net/>
- Jaydatt. (2024, February 21). *jaydattpatel/Linux-commands-and-Shell-Scripts*. GitHub.
<https://github.com/jaydattpatel/Linux-commands-and-Shell-Scripts>
- A Cozy Calendar Setup With Remind. (n.d.). Www.youtube.com. Retrieved March 3, 2024, from
<https://www.youtube.com/watch?v=4FdMrS4biVE>
- Saive, R. (2022, September 12). *6 Best To-Do List Managers for Linux Command Line*.
 Www.ubuntumint.com. <https://www.ubuntumint.com/command-line-to-do-list-managers/>
- Dancuk, M. (2023, November 16). *SCP Command in Linux {13 Examples}* | phoenixNAP KB.
 Knowledge Base by PhoenixNAP. <https://phoenixnap.com/kb/linux-scp-command>
- Let's Build a Linux Shell [Part I] | HackerNoon. (n.d.). Hackernoon.com. Retrieved March 3, 2024, from
<https://hackernoon.com/lets-build-a-linux-shell-part-i-bz3n3vg1>
- Developing a Linux based shell. (2018, June 18). GeeksforGeeks.
<https://www.geeksforgeeks.org/developing-linux-based-shell/>