



CREATING
CREATORS

Linux Operating System Utilities

Engenharia Informática

ANO/SEMESTRE: 2023-2024 / 6º Semestre

Unidade Curricular: Sistemas Operativos
Professor: Pedro Rosa

Carolynne Silva - 20210046

Gustavo Farinha – 20211115

github.com/CMS77/Linux-Operating-System-Utilities

Descrição do Problema

Atualmente, a capacidade de interagir eficazmente com sistemas Linux é fundamental para uma variedade de utilizadores e administradores de sistemas. Embora o Linux ofereça uma vasta gama de utilitários e comandos prontos a usar, há sempre espaço para melhorias e personalizações para satisfazer as necessidades específicas de cada indivíduo ou organização, promovendo assim uma maior produtividade e controlo sobre o sistema. Esta capacidade de personalização e melhoria contínua realça a natureza dinâmica e adaptável do ecossistema Linux, tornando-o uma escolha poderosa para utilizadores que valorizam a flexibilidade e a liberdade de personalização nos seus sistemas operativos.

Motivação: Desenvolver utilitários para o Linux em Linguagem C proporciona um excelente exercício de programação, permitindo uma compreensão mais profunda sobre como essas ferramentas funcionam. Para além de contribuir para a comunidade e disponibilizar um conjunto de utilitários open-source, oferecendo alternativas aos utilitários existentes, além de promover a colaboração e o compartilhamento de conhecimento.

Identificação de Casos de Uso

ls: Este utilitário será desenvolvido para listar os arquivos e diretórios presentes no diretório atual. Ele incluirá opções para exibir informações detalhadas, como permissões, tamanho e data de modificação dos arquivos.

cat: O utilitário cat será desenvolvido para concatenar e exibir o conteúdo de arquivos. Ele poderá receber um ou mais arquivos como argumentos e exibirá o conteúdo desses arquivos na saída padrão.

rm: O utilitário rm será desenvolvido para remover arquivos e diretórios do sistema de arquivos. Ele poderá aceitar opções para forçar a remoção de arquivos, ignorar mensagens de erro e recursivamente remover diretórios.

cp: O utilitário cp será desenvolvido para copiar arquivos e diretórios para um novo local no sistema de arquivos. Ele poderá aceitar opções para preservar metadados, como permissões e timestamps, recursivamente copiar diretórios e sobrescrever arquivos existentes.

mv: O utilitário mv será desenvolvido para mover arquivos e diretórios para um novo local no sistema de arquivos. Ele poderá aceitar opções para preservar metadados, como permissões e timestamps, e sobrescrever arquivos existentes no destino.

grep: O utilitário grep será desenvolvido para pesquisar por padrões em arquivos de texto. Ele poderá aceitar opções para realizar pesquisas sensíveis a maiúsculas e minúsculas, exibir linhas que correspondem ou não correspondem ao padrão especificado e recursivamente pesquisar diretórios.

tail: O utilitário tail será desenvolvido para exibir as últimas linhas de um arquivo de texto. Ele poderá aceitar opções para especificar o número de linhas a serem exibidas, acompanhar as adições ao arquivo em tempo real e exibir linhas invertidas.

head: O utilitário head será desenvolvido para exibir as primeiras linhas de um arquivo de texto. Ele poderá aceitar opções para especificar o número de linhas a serem exibidas e exibir linhas de múltiplos arquivos.

Enquadramento nas áreas da Unidade Curricular:

Este projeto está alinhado com os objetivos da unidade curricular de Sistemas Operativos, onde teremos a oportunidade de desenvolver um entendimento prático dos conceitos fundamentais de sistemas operativos e aprofundar nosso conhecimento através da implementação de utilitários básicos.

Também poderá ser incluído na unidade curricular de Compiladores visto que teremos de desenvolver a compilação de utilitários.

Requisitos Técnicos para desenvolvimento do projeto:

- Conhecimento básico em Linguagem C.
- Compreensão dos conceitos fundamentais de sistemas operativos, como manipulação de arquivos, gerenciamento de processos e comunicação entre processos.
- Ambiente de desenvolvimento Linux para compilar e testar os utilitários.

Arquitetura de Solução

Implementação Individual:

- Desenvolver cada utilitário no seu próprio arquivo de código-fonte (Linguagem C);
- Cada arquivo contém a implementação completa do utilitário, incluindo lógica e manipulação de entrada/saída;

Compilação:

- Utilização de uma ferramenta de compilação, como GCC, para compilar cada arquivo de código-fonte separadamente;
- Resultará na geração de um executável independente para cada utilitário;

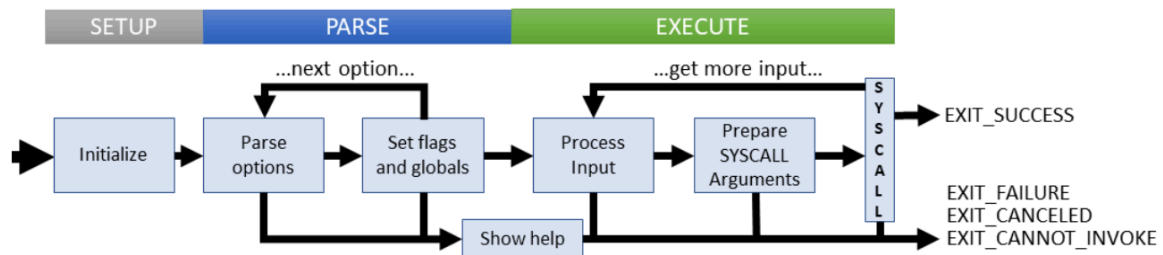
Execução na Linha de Comando:

- Após a compilação, os utilitários podem ser executados diretamente na linha de comando do Linux.

Testes:

- Implementação de testes para cada utilitário;
- Estes testes permitem garantir o funcionamento correto do utilitário;

Usaremos como guia a arquitetura disponibilizada no site da GNU. De acordo com os mesmos, esta arquitetura é utilizada para a maioria dos utilitários.



Tecnologias a utilizar

Linguagem de Programação - Linguagem C para o desenvolvimento dos utilitários visto a eficiência e capacidade de interagir diretamente com o sistema operacional.

Editor de Texto - Visual Studio Code

Compilação - GCC (GNU Compiler Collection)

Controle de versão do código fonte - GitHub

Planeamento

SEMANAS	SEMANA 18/03-24/03	SEMANA 25/03-31/03	SEMANA 01/04-07/04	SEMANA 08/04-14/04	SEMANA 15/04-21/04	SEMANA 22/04-28/04	SEMANA 29/04-05/05	SEMANA 06/05-12/05	SEMANA 13/05-19/05	SEMANA 20/05-24/05
Carolyne	Desenvolvimento de 4 utilitários									
Gustavo	Desenvolvimento de 4 utilitários									
Ambos						Compilação dos utilitários		Teste dos utilitários		Documentação Final e Testes finais

Bibliografia

<https://chat.openai.com/>

<https://www.canva.com/>

https://pt.wikipedia.org/wiki/GNU_Core_Uilities

<https://www.gnu.org/software/coreutils/>

Arquitetura GNU: <https://www.maizure.org/projects/decoded-gnu-coreutils/>