



Curso de Engenharia Informática

Projeto de Sistemas Operativos

“Linux Operating System Utilities”

Miguel Sousa Bernardo - 20211066

Pedro Miguel Ramos Mustra Cesar - 20211192

Roberta Pereira Santos, 20211091

<https://github.com/yxngczsr/SO>

Introdução:

No universo dos sistemas operativos, a otimização e simplificação de tarefas diárias são fatores cruciais para proporcionar uma experiência eficiente aos utilizadores. No âmbito da Unidade Curricular de Sistemas Operativos, surge a proposta de um projeto prático: o desenvolvimento de utilitários para o sistema operativo Linux. Esta iniciativa visa criar ferramentas de linha de comando para facilitar interações com o sistema de arquivos.

Num ambiente Linux, a execução de tarefas básicas, como a manipulação de arquivos, navegação em diretórios e gestão de espaço em disco, frequentemente envolve o uso de comandos complexos. Por isso este projeto surge da necessidade de simplificar essas interações diárias, promovendo assim a automação e a facilidade de uso.

Objetivos do Projeto:

O objetivo principal deste projeto é a implementação de utilitários de linha de comando do sistema operativo Linux, como “ls”, “mkdir”, “cp”, entre outros. Cada utilitário será desenvolvido para abordar tarefas específicas, desde listagem de conteúdos de diretórios até a compressão de arquivos.

Descrição do Problema:

O projeto proposto visa abordar desafios práticos no contexto do desenvolvimento de utilitários para o sistema operativo Linux. A motivação por trás deste trabalho está centrada na necessidade de facilitar tarefas comuns do utilizador e na automação de processos.

Motivação do Trabalho:

A Facilitação das tarefas diárias com a criação de utilitários como “ls”, “mkdir”, “cp”, “mv” entre outros, pois proporcionam uma interface amigável e eficiente para tarefas rotineiras de gestão de arquivos e diretórios.

A automatização de processos, pois ao implementar utilitários como “find” para busca de arquivos ou “du” para exibir espaço utilizado, onde o objetivo é simplificar processos que normalmente exigiriam comandos mais complexos ou múltiplas operações.

Identificação de Casos de Uso:

Cenário 1 - Organização de Arquivos:

Utilizadores podem utilizar o utilitário “mkdir” para criar estruturas de diretórios organizadas de forma eficiente.

Cenário 2 - Busca de Arquivos:

O utilitário “find” será útil para utilizadores que necessitam localizar rapidamente arquivos específicos em uma hierarquia de diretórios extensa.

Cenário 3 - Otimização de Espaço em Disco:

A utilização do utilitário “du” permitirá aos utilizadores identificar e otimizar o espaço utilizado por diretórios e arquivos no sistema de arquivos.

Benefícios Esperados:

Simplificação de comandos, pois os utilitários propostos reduzirão a complexidade de comandos mais avançados, proporcionando uma abordagem amigável e acessível.

Eficiência operacional, automatizando tarefas comuns, pois os utilitários visam tornar a utilização do sistema operativo mais eficiente e intuitiva para os utilizadores.

Descrição da Solução:

O projeto concentra-se no desenvolvimento de utilitários de linha de comando para interações dos utilizadores com o sistema operativo Linux. Cada utilitário será projetado para desempenhar funções específicas.

Enquadramento na Unidade Curricular Sistemas Operativos:

O projeto está relacionado aos objetivos da Unidade Curricular de Sistemas Operativos, proporcionando uma aplicação prática dos seguintes conceitos:

Processos e Threads: Utilização de processos ou threads para implementar os utilitários de forma concorrente quando necessário.

Sistema de Arquivos: Interação direta com o sistema de arquivos, explorando chamadas de sistema específicas para leitura, escrita e manipulação de diretórios.

Chamadas de Sistema: Implementação de funcionalidades críticas através de chamadas de sistema apropriadas, garantindo eficiência e conformidade com as características do kernel.

Requisitos Técnicos para Desenvolvimento do Projeto:

Ambiente Linux:

Desenvolvimento e teste em um ambiente Linux para garantir a compatibilidade e funcionalidade dos utilitários.

Compilador C:

Utilização de um compilador C para traduzir o código fonte em código executável, considerando a eficiência de execução.

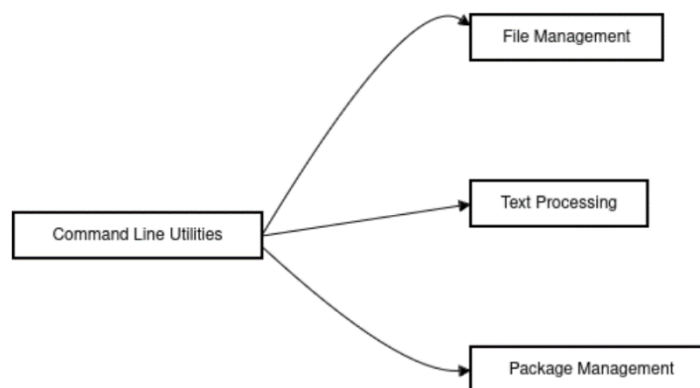
Arquitetura da Solução:

Utilitários Orientados a Processos:

Utilização de processos para implementar utilitários que demandam operações concorrentes.

Gestão Eficiente de Recursos:

Consideração cuidadosa da gestão de memória e recursos do sistema para otimizar o desempenho dos utilitários.



Tecnologias a Utilizar:

Editor de Texto: Visual Studio Code

Linguagem de Programação C:

Implementação em Linguagem C para aproveitar os recursos de baixo nível e garantir a eficiência do código.

Compilação - GCC (GNU Compiler Collection)

Lista de Utilitários a elaborar:

"ps": Exibe informações sobre processos em execução.

"top": Exibe informações em tempo real sobre a utilização do sistema e dos processos.

"find": Procura arquivos em uma hierarquia de diretórios.

"tar": Empacota e desempacota arquivos.

"gzip": Comprime ou descomprime arquivos.

"cp": Copia arquivos e diretórios.

"mv": Move (ou renomeia) arquivos e diretórios.

"du": Exibe o espaço utilizado por diretórios e arquivos.

"ls": Lista o conteúdo de um diretório.

"mkdir": Cria um diretório.

Comando dos utilitários:

1. Utilitário "ps":

ps: Exibe todos os processos em execução.

ps aux: Mostra todos os processos do sistema em um formato detalhado.

ps -u usuário: Mostra os processos do usuário especificado.

2. Utilitário "top":

top: Exibe uma lista dinâmica dos processos em execução, atualizando periodicamente.

powertop: Fornece informações sobre o consumo de energia do sistema e sugestões para otimizá-lo.

iostat: Exibe informações sobre a utilização de E/S (entrada/saída) de disco por processos em execução.

3. Utilitário "find":

locate: Localiza arquivos pelo nome de forma rápida usando um banco de dados indexado. É mais rápido que o find, mas menos preciso.

grep: Pode ser usado para procurar texto dentro de arquivos, sendo útil para localizar arquivos com base em seu conteúdo.

fdupes: Encontra e lista arquivos duplicados em uma hierarquia de diretórios.

4. Utilitário “gzip”:

gzip arquivo.txt: Comprime o arquivo especificado.

gzip -d arquivo.gz: Descomprime o arquivo especificado.

zcat: Exibe o conteúdo de arquivos comprimidos sem a necessidade de descompactar.

5. Utilitário “cp”:

mv: Move arquivos e diretórios de um local para outro, sendo útil quando a intenção é mover o arquivo em vez de copiá-lo.

cp -r diretorio_origem diretorio_destino: Copia recursivamente todo um diretório para outro destino.

cp -i arquivo.txt destino: Copia o arquivo para o destino, mas pede confirmação caso já exista um arquivo com o mesmo nome.

6. Utilitário “du”:

du -h: Exibe o espaço usado por cada diretório de forma humanamente legível.

du -sh /diretorio: Mostra o espaço total usado por um diretório de forma resumida.

du -ah /diretorio | sort -n: Lista todos os arquivos e diretórios dentro do diretório ordenados pelo tamanho.

7. Utilitário “ls”:

ls: Lista o conteúdo do diretório atual.

ls -l: Exibe informações detalhadas, incluindo permissões, dono, grupo e tamanho dos arquivos.

ls -a: Mostra todos os arquivos, incluindo os ocultos.

8. Utilitário “mkdir”:

mkdir novo_diretorio: Cria um novo diretório no diretório atual.

mkdir -p /caminho/para/novo/diretorio: Cria a estrutura de diretórios especificada, se não existir.

rmdir: Remove diretórios vazios. É útil quando se deseja excluir diretórios somente se estiverem vazios.

Conclusão:

Em resumo, a implementação dos utilitários para o sistema operativo Linux permitiu a aplicação concreta dos conceitos de Sistemas Operativos, resultando em ferramentas modulares que simplificam operações diárias.

Bibliografia:

<https://www.linuxnews.pro/popular-linux-commands-every-beginner-should-know/>

<https://techreviewer.co/blog/utility-software-15-excellent-examples-of-utility-software>

[O Sistema Operacional GNU e o Movimento de Software Livre](#)

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

<https://linux.die.net/man/>

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

Tanenbaum, A. S., & Bos, H. (2014). Modern Operating Systems. Pearson.

Stallings, W. (2014). Operating Systems: Internals and Design Principles. Pearson.

Kernighan, B. W., & Ritchie, D. M. (1988). The C Programming Language. Prentice Hall.