



Curso de Engenharia Informática

Projeto de Sistemas Operativos

“Linux Operating System Utilities”

Afonso Libano Caldeira da Silva - 20210859

Miguel Sousa Bernardo - 20211066

Pedro Miguel Ramos Mustra Cesar - 20211192

Roberta 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:

Linguagem de Programação C:

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

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.

Planeamento dos utilitários:

1. Utilitário "ps":

Objetivo: Exibir informações sobre processos em execução.

Entrada Esperada: Nenhuma entrada direta do usuário.

Processamento Necessário: Coletar informações sobre os processos em execução no sistema. Formatando e exibindo essas informações de maneira legível.

Saída Gerada: Lista de processos em execução, com detalhes como o estado, uso de CPU, etc.

Casos Especiais ou Tratamento de Erros: Lidar com permissões insuficientes ou situações em que a obtenção de informações de processos falha.

Recursos Adicionais: Opções de linha de comando para filtrar processos.

2. Utilitário "top":

Objetivo: Exibir informações em tempo real sobre a utilização do sistema e dos processos.

Entrada Esperada: Nenhuma entrada direta do usuário.

Processamento Necessário: Monitorar a atividade do sistema e dos processos e atualizar informações exibidas em tempo real.

Saída Gerada: Informações como uso de CPU, memória, processos em execução, etc.

Casos Especiais ou Tratamento de Erros: Lidar com situações em que as informações em tempo real não podem ser obtidas.

Recursos Adicionais: Ordenar a exibição por diferentes métricas.

3. Utilitário "find":

Objetivo: Procurar arquivos em uma hierarquia de diretórios.

Entrada Esperada: Diretório inicial e critérios de pesquisa.

Processamento Necessário: Percorrer recursivamente a hierarquia de diretórios.

Encontrar e listar arquivos que correspondam aos critérios de pesquisa.

Saída Gerada: Lista de caminhos de arquivos encontrados.

Casos Especiais ou Tratamento de Erros: Lidar com casos em que não há permissão para acessar determinados diretórios.

Recursos Adicionais: Opções de linha de comando para especificar critérios de pesquisa mais complexos.

4. Utilitário "tar":

Objetivo: Empacotar e desempacotar arquivos.

Entrada Esperada: Lista de arquivos a serem empacotados ou arquivo a ser desempacotado.

Processamento Necessário: Empacotar ou desempacotar os arquivos conforme especificado.

Saída Gerada: Arquivo "tar" gerado ou arquivos desempacotados.

Casos Especiais ou Tratamento de Erros: Lidar com casos em que os arquivos não existem ou não podem ser acessados.

Recursos Adicionais: Opções de linha de comando para especificar o nome do arquivo de saída, ajustar níveis de compressão, etc.

5. Utilitário "gzip":

Objetivo: Comprimir ou descomprimir arquivos.

Entrada Esperada: Arquivo a ser comprimido ou descomprimido.

Processamento Necessário: Aplicar algoritmo de compressão ou descompressão.

Saída Gerada: Arquivo comprimido ou descomprimido.

Casos Especiais ou Tratamento de Erros: Lidar com casos de compressão/descompressão falha.

Recursos Adicionais: Opções de linha de comando para ajustar níveis de compressão, especificar o nome do arquivo de saída, etc.

6. Utilitário “cp”:

Objetivo: Copiar arquivos e diretórios.

Entrada Esperada: Lista de arquivos/diretórios a serem copiados e destino.

Processamento Necessário: Criar cópias dos arquivos/diretórios conforme especificado.

Saída Gerada: Mensagem indicando se a operação foi bem-sucedida.

Casos Especiais ou Tratamento de Erros: Lidar com casos em que os arquivos não existem ou não podem ser acessados.

Recursos Adicionais: Opções de linha de comando para especificar comportamento (por exemplo, cópia recursiva).

7. Utilitário “mv”:

Objetivo: Mover (ou renomear) arquivos e diretórios.

Entrada Esperada: Lista de arquivos/diretórios a serem movidos e destino.

Processamento Necessário: Mover ou renomear os arquivos/diretórios conforme especificado.

Saída Gerada: Mensagem indicando se a operação foi bem-sucedida.

Casos Especiais ou Tratamento de Erros: Casos em que os arquivos não existam ou não podem ser acessados.

Recursos Adicionais: Opções de linha de comando para especificar comportamento.

8. Utilitário “du”:

Objetivo: Exibir o espaço utilizado por diretórios e arquivos.

Entrada Esperada: Diretório a ser analisado.

Processamento Necessário: Calcular o espaço utilizado pelos diretórios e arquivos no diretório especificado.

Saída Gerada: Lista do espaço utilizado por cada diretório e arquivo.

Casos Especiais ou Tratamento de Erros: Lidar com casos em que não há permissão para acessar determinados diretórios.

Recursos Adicionais: Opções de linha de comando para ajustar o nível de detalhe ou a unidade de medida.

9. Utilitário “ls”:

Objetivo: Lista o conteúdo de um diretório.

Entrada Esperada: Diretório a ser listado.

Processamento Necessário: Obter a lista de arquivos e diretórios no diretório especificado.

Saída Gerada: Lista de nomes de arquivos e diretórios.

Casos Especiais ou Tratamento de Erros: Lidar com casos em que não há permissão para acessar determinados diretórios.

Recursos Adicionais: Opções de linha de comando para ajustar o formato da saída ou incluir informações adicionais.

10. Utilitário “mkdir”:

Objetivo: Cria um diretório.

Entrada Esperada: Nome do diretório a ser criado.

Processamento Necessário: Usar a função “mkdir” para criar um novo diretório.

Saída Gerada: Mensagem indicando se o diretório foi criado com sucesso ou se já existe.

Casos Especiais ou Tratamento de Erros: Lidar com casos em que o diretório não pode ser criado devido a permissões insuficientes ou outros problemas.

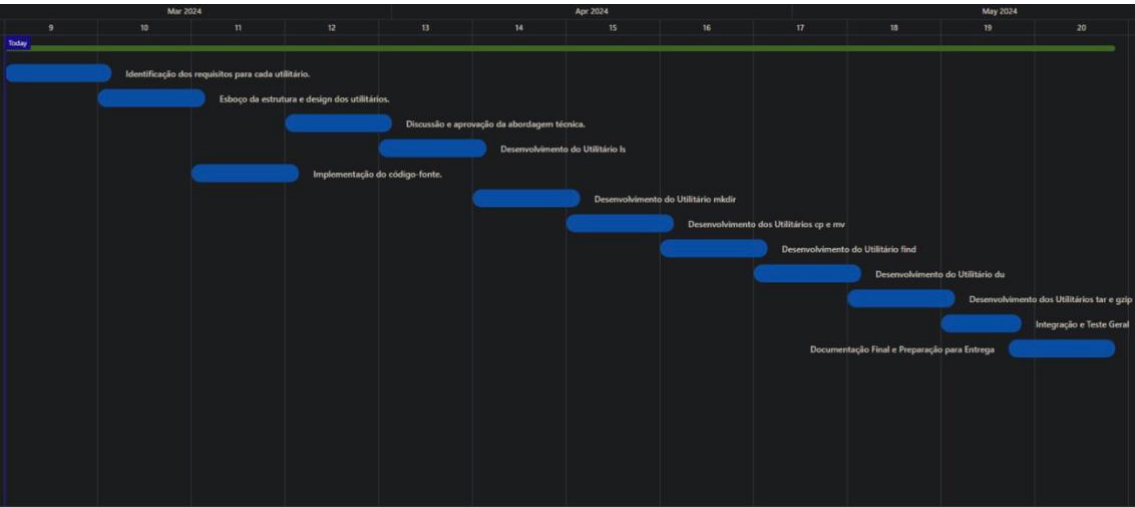
Recursos Adicionais: Opções de linha de comando para especificar caminho ou permissões.

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.

Planeamento:

Name	Assignee	Due date	Priority
<input type="radio"/> Documentação Final e Preparação para Ent...	👤	May 24	🚩
<input type="radio"/> Integração e Teste Geral	👤	May 19	🚩
<input type="radio"/> Desenvolvimento dos Utilitários tar e gzip	👤	Apr 12	🚩 High
<input type="radio"/> Desenvolvimento do Utilitário du	👤	May 5	🚩 Normal
<input type="radio"/> Desenvolvimento do Utilitário find	👤	Apr 28	🚩 High
<input type="radio"/> Desenvolvimento dos Utilitários cp e mv	👤	Apr 14	🚩 Normal
<input type="radio"/> Desenvolvimento do Utilitário mkdir	👤	Apr 7	🚩 Low
<input type="radio"/> Desenvolvimento do Utilitário ls	👤	Mar 31	🚩 Low
<input type="radio"/> Discussão e aprovação da abordagem técni...	👤	Mar 24	🚩
<input type="radio"/> Implementação do código-fonte.	👤	Mar 17	🚩
<input type="radio"/> Esboço da estrutura e design dos utilitários.	👤	Mar 17	🚩
<input type="radio"/> Identificação dos requisitos para cada utilit...	👤	Mar 10	🚩



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.