



CREATING
CREATORS

Sistemas Operativos

Linux Operating System Utilities

https://github.com/alanocb/Linux_Operating_System_Utilities

Alano Baptista - 20190818

Luquenía Galiano – 20210451

Descrição do Problema:

O trabalho consistiu em explorar e entender a funcionalidade de diversos utilitários do sistema operacional Linux, comumente utilizados em ambientes de desenvolvimento, administração de sistemas e operações diárias

Motivação:

O uso eficiente desses utilitários é essencial para melhorar a produtividade e facilitar tarefas comuns no desenvolvimento, administração de sistemas e manipulação de arquivos. Permitindo realizar operações como manipulação de texto, ordenação, filtragem, busca e gerenciamento de espaço em disco de forma rápida e eficiente, sem a necessidade de escrever programas personalizados para cada tarefa.

Ao compreender e dominar esses utilitários, os usuários podem:

- Automatizar tarefas repetitivas e complexas.
- Realizar análises rápidas de dados.
- Gerenciar e otimizar o uso de armazenamento em disco.
- Realizar buscas eficientes em arquivos de registo ou código-fonte.

Os utilitários do sistema operativo Linux oferecem uma variedade de ferramentas poderosas para lidar com diversas tarefas comuns. Compreender o seu funcionamento e aprender a utilizá-los de forma eficiente é fundamental para aumentar a produtividade e a eficácia no ambiente de trabalho, promovendo também uma melhor compreensão do sistema operativo Linux em geral.

É importante salientar que muitos destes utilitários são desenvolvidos e mantidos pela comunidade de software livre, com contribuições significativas da Free Software Foundation (FSF). A FSF é uma organização sem fins lucrativos com uma missão global de promover a liberdade dos utilizadores de computador. Defende os direitos de todos os utilizadores de software, promovendo a ideia de software livre e os princípios de colaboração e partilha.

Ao explorar e experimentar estes utilitários, os utilizadores não só melhoram a sua produtividade, como também se envolvem numa comunidade que valoriza a liberdade do utilizador e a partilha de conhecimento.

echo - Descrição da Solução:

O comando echo é utilizado para imprimir mensagens na saída padrão. Nesta implementação específica, o echo foi estendido para incluir funcionalidades adicionais, como simular a propagação do som através de diferentes materiais e distâncias.

O programa echo desenvolvido recebe três argumentos: o tipo de material, a distância em metros e a mensagem a ser impressa. Dependendo do tipo de material especificado, o programa calcula o tempo de atraso necessário para que a mensagem seja propagada

através desse material até a distância especificada. Em seguida, a mensagem é impressa na saída padrão após o tempo de atraso calculado.

Funcionalidades Adicionais:

- **Opção --help:** Se o utilizador fornecer a opção --help, será exibida uma mensagem de utilização que descreve como utilizar o programa e os materiais disponíveis.
- **Simulação de Propagação do Som:** O programa simula a propagação do som através de diferentes materiais (ar, água, vidro e aço) e calcula o tempo de atraso necessário para a mensagem atingir a distância especificada.

sort - Descrição da Solução:

O programa de ordenação desenvolvido permite ordenar um conjunto de números armazenados em um arquivo de texto utilizando diferentes algoritmos de ordenação. Os algoritmos de ordenação implementados são: Bubble Sort, Selection Sort, Insertion Sort, Merge Sort e Bogo Sort.

Ao executar o programa, o utilizador especifica o algoritmo de ordenação a ser utilizado e o arquivo que contém os números a serem ordenados. O programa então lê os números do arquivo, executa o algoritmo de ordenação escolhido e exibe os números ordenados na saída padrão, juntamente com o tempo de execução do algoritmo.

Funcionalidades Adicionais:

1. **Opção --help:**
 - Se o utilizador fornecer a opção --help, será exibida uma mensagem de utilização que descreve como utilizar o programa e os algoritmos de ordenação disponíveis.
2. **Análise de Complexidade de Tempo:**
 - Para cada algoritmo de ordenação, o programa exibe a complexidade de tempo no melhor caso e no pior caso.
3. **Implementação de Diferentes Algoritmos de Ordenação:**
 - O programa implementa diferentes algoritmos de ordenação, incluindo Bubble Sort, Selection Sort, Insertion Sort, Merge Sort e Bogo Sort, oferecendo ao utilizador opções diversificadas para ordenação de números.

head / tail - Descrição da Solução:

O comando head é utilizado para exibir as primeiras linhas de um ou mais arquivos, enquanto o comando tail é utilizado para exibir as últimas linhas. Estas implementações permitem ao utilizador especificar o número de linhas ou bytes a serem exibidos, além de oferecer opções para controlar a exibição de cabeçalhos e a leitura de arquivos da entrada padrão.

Funcionalidades Adicionais:

1. Opções de Número de Linhas ou Bytes:

- O utilizador pode especificar o número de linhas ou bytes a serem exibidos utilizando as opções -n e -c, respectivamente.

2. Opções de Cabeçalhos:

- As opções -q e -v controlam a exibição de cabeçalhos com os nomes dos arquivos. A opção -q suprime a exibição dos cabeçalhos, enquanto a opção -v garante que os cabeçalhos sejam sempre exibidos.

du - Descrição da Solução:

O programa percorre recursivamente os diretórios e calcula o tamanho total do uso em disco dos arquivos e subdiretórios encontrados. Ele aceita opções como -h para exibir os tamanhos em formato legível para humanos (KB, MB, GB, etc), -s para mostrar apenas o total de um diretório, -a para incluir todos os arquivos na saída, e --time para exibir o tempo de execução do comando.

grep - Descrição da Solução:

O programa grep é uma ferramenta de linha de comando utilizada para pesquisar padrões em arquivos ou entrada padrão. Ele permite que os utilizadores localizem linhas de texto que correspondam a um determinado padrão especificado. O programa aceita vários argumentos, incluindo opções para personalizar a pesquisa, como ignorar maiúsculas e minúsculas, exibir números de linha, contar o número de ocorrências e inverter a correspondência.

Funcionalidades Adicionais:

- Opção -i: Ao usar esta opção, o grep ignora as diferenças entre maiúsculas e minúsculas durante a pesquisa, tornando-a insensível a maiúsculas e minúsculas.
- Opção -n: Quando esta opção é fornecida, o grep exibe o número de linha para cada ocorrência encontrada, permitindo aos usuários localizar facilmente as ocorrências em um contexto de linha específico.
- Opção -c: Com esta opção, o grep conta o número total de ocorrências do padrão em cada arquivo, útil para fins estatísticos ou para determinar a frequência de um padrão.
- Opção -v: Esta opção inverte a correspondência, fazendo com que o grep exiba apenas as linhas que não contêm o padrão especificado.

shred - Descrição da Solução:

O shred é uma ferramenta de linha de comando projetada para sobrescrever o conteúdo de arquivos, dificultando a recuperação de dados. Esta implementação específica permite que os utilizadores especifiquem o número de vezes que desejam sobrescrever o arquivo, adicionem uma sobrescrita final com zeros e optem por remover o arquivo após a operação de sobrescrita.

O shred abre o arquivo especificado em modo de escrita e realiza as sobrescritas conforme necessário. Ele gera dados aleatórios para sobrescrever o conteúdo original do

arquivo, garantindo que os dados anteriores não sejam facilmente recuperáveis. Após as sobrescritas, o programa fecha o arquivo e, se a opção de remoção estiver ativada, remove o arquivo do sistema de arquivos.

Funcionalidades Adicionais:

- Opção -n num: Permite aos utilizadores especificar o número de vezes que desejam sobrescrever o arquivo, proporcionando maior segurança na eliminação dos dados.
- Opção -z: Adiciona uma sobrescrita final com zeros, garantindo que não haja informações remanescentes no arquivo após o processo de sobrescrita.
- Opção -u: Remove o arquivo após a operação de sobrescrita, oferecendo uma maneira conveniente de garantir a eliminação completa dos dados.

wc- Descrição da Solução:

O wc é um programa de linha de comando desenvolvido para contar o número de linhas, palavras e caracteres em um arquivo especificado. Com esta implementação, os utilizadores podem optar por contar apenas as linhas, apenas as palavras, apenas os caracteres ou uma combinação desses três.

Quando o wc é executado, os utilizadores podem fornecer opções como -l para contar o número de linhas, -w para contar o número de palavras e -c para contar o número de caracteres. Além disso, o programa fornece a opção --help para exibir uma mensagem de ajuda que descreve como usar o programa e suas opções disponíveis.

Funcionalidades Adicionais:

- Opção -l: Permite contar apenas o número de linhas no arquivo.
- Opção -w: Permite contar apenas o número de palavras no arquivo.
- Opção -c: Permite contar apenas o número de caracteres no arquivo.
- Opção --help: Exibe uma mensagem de ajuda detalhada sobre como usar o programa e suas opções disponíveis.

Enquadramento nas Áreas da Unidade Curricular (Sistemas Operativos):

Esta solução está diretamente relacionada com a Unidade Curricular de Sistemas Operativos, pois envolve o desenvolvimento e a compreensão de utilitários de linha de comando essenciais para operar e administrar sistemas Linux. Os utilitários abordados são amplamente utilizados na linha de comando do sistema operativo Linux e são fundamentais para administradores de sistemas, desenvolvedores de software e usuários avançados.

Requisitos Técnicos para Desenvolvimento do Projeto:

- Conhecimento em linguagem de programação C para implementar os utilitários.
- Familiaridade com o sistema operativo Linux e seu ambiente de linha de comando.
- Capacidade de compreender e implementar algoritmos eficientes para as funcionalidades dos utilitários.

- Habilidade para testar e depurar os utilitários para garantir seu funcionamento correto.

Arquitetura da Solução:

A arquitetura da solução consistirá em um conjunto de programas independentes, cada um implementando um utilitário específico. Cada utilitário será projetado para receber entrada de arquivos ou da entrada padrão, processá-la conforme sua funcionalidade e produzir a saída desejada na saída padrão ou em um arquivo de saída, conforme especificado pelo utilizador.

Tecnologias a Utilizar:

- Linguagem de programação C para implementação dos utilitários.
- Ferramentas de compilação de código C, como GCC, para compilar os utilitários.
- Ambiente de desenvolvimento Linux para testar e depurar os utilitários.

Bibliotecas utilizadas:

stdio.h: Fornece funções para entrada e saída padrão.

stdlib.h: Contém funções para alocação de memória, conversão de tipos e outras utilidades.

string.h: Oferece funções para manipulação de strings.

unistd.h: Fornece acesso ao sistema de chamadas do sistema operacional POSIX.

getopt.h: Utilizada para processar opções de linha de comando.

fcntl.h: Oferece funcionalidades de controle de arquivos, como abrir, fechar e manipular descritores de arquivo.

errno.h: Define as variáveis globais `errno`, que indicam erros reportados por chamadas de sistema.

sys/stat.h: Contém definições de estruturas e funções para obtenção de informações sobre arquivos.

dirent.h: Fornece funções para manipulação de diretórios e leitura de entradas de diretórios.

time.h: Oferece funções para manipulação de tempo e datas.

ctype.h: Fornece funções para manipulação de caracteres.

Resultados

No geral, conseguimos atingir nosso objetivo de implementar 8 comandos diferentes em C, cada um com suas funcionalidades específicas e opções de linha de comando. Os pontos fortes da implementação incluem a variedade de opções disponíveis para cada comando, o que permite uma flexibilidade maior na execução das tarefas desejadas. Além disso, todos os comandos são funcionais e podem ser utilizados para realizar operações úteis em arquivos e diretórios.

Destaca-se que os comandos `echo` e `sort` foram particularmente criativos em comparação com os outros, apresentando funcionalidades adicionais e opções que os

tornam mais versáteis em diferentes situações. No entanto, reconhecemos que os demais comandos poderiam ter sido mais explorados em termos de criatividade, possibilitando a adição de opções e funcionalidades adicionais para torná-los mais robustos e úteis em uma variedade de cenários.

Exemplos de uso:

A pasta source contém ficheiros txt para facilitar a realização de uso dos comandos realizados.

```
gcc -o <nome> <nome.c>
```

./echo --help

Usage: ./echo <material> <distance_in_meters> <message>

Materials:

-a: air (343 m/s)

-w: water (1480 m/s)

-g: glass (4540 m/s)

-s: steel (5960 m/s)

./echo -w 1000 Hello World

Material: water Speed of sound: 1480.00 m/s Delay: 1.35

Hello World

./sort --help

Usage: ./sort <algorithm> <numbers.txt>

Sorting algorithms:

-b : Bubble Sort

-s : Selection Sort

-i : Insertion Sort

-m : Merge Sort

-l : Bogo Sort

--help : Display this help message

./sort -b test.txt

Time complexity: Best Case - $O(n)$, Worst Case - $O(n^2)$

Sorted numbers:

11 12 23 34 45 56 67 78 89 92

Execution time: 0.000065 seconds

./head --help

Uso: ./head [-arg] [num] [ficheiro.txt]

Opções:

-n num Imprimir as primeiras 'num' linhas em vez das primeiras 10 linhas.

-c num Imprimir os primeiros 'num' bytes de cada ficheiro.

-q Nunca imprimir cabeçalhos com os nomes dos ficheiros.

-v Imprimir cabeçalhos com os nomes dos ficheiros.

--help Mostrar help e sair.

./head -n 5 -q file1.txt

Linha 1 do file1

Linha 2 do file1

Linha 3 do file1

Linha 4 do file1

Linha 5 do file1

./tail --help

Uso: ./tail [-n num] [-c num] [-qv] [ficheiro...]

Opções:

-n num Imprimir as últimas 'num' linhas em vez das últimas 10 linhas.

-c num Imprimir os últimos 'num' bytes de cada ficheiro.

-q Nunca imprimir cabeçalhos com os nomes dos ficheiros.

-v Sempre imprimir cabeçalhos com os nomes dos ficheiros.

--help Mostrar esta ajuda e sair.

./tail -n 5 -v file2.txt

==> file2.txt <==

Linha 11 do file2

Linha 12 do file2

Linha 13 do file2

Linha 14 do file2

Linha 15 do file2

./du --help

Uso: ./du [-h] [-s] [-a] [--time] [diretório]

Opções:

-h Exibe os tamanhos em formato legível para humanos (KB, MB, GB, etc).

-s Mostra apenas o total de um diretório. -a Inclui todos os arquivos na saída, não apenas os diretórios.

--time Exibe o tempo de execução do comando.

--help Exibe esta mensagem de ajuda e sai. Exibe o tamanho de uso do disco de arquivos e diretórios. Se nenhum ARQUIVO for especificado, será utilizado o diretório atual.

./du -h -s --time

175.8KB .

Tempo de execução: 0.00 segundos

./grep --help

Uso: ./grep [opcao] <palavra> <ficheiro>

Opções:

- i Ignorar maiúsculas e minúsculas
- n Mostrar números das linhas
- c Contar o número de ocorrências
- v Inverter a correspondência
- help Mostrar esta ajuda e sair

Procurar por <palavra> em cada <ficheiro> ou entrada <palavra>.

<palavra>: A palavra, número ou letra a ser procurada.

./grep -i justo words.txt

words.txt: Suspendisse in justo eu magna luctus suscipit. Sed lectus. Integer euismod lacus luctus magna. Quisque cursus, metus vitae pharetra auctor, sem massa mattis sem, at interdum magna augue eget diam. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Morbi lacinia molestie dui. Praesent blandit dolor. Sed non quam. In vel mi sit amet augue congue elementum. Morbi in ipsum sit amet pede facilisis laoreet.

./shred --help

Uso: ./shred [-n num] [-z] [-u] [--help] arquivo

Opções:

- n num Número de vezes para sobrescrever o arquivo (padrão é 3).
- z Adiciona uma sobrescrita final com zeros.
- u Remove o arquivo após a sobrescrita.
- help Exibe esta mensagem de ajuda e sai.

./shred -n 3 shred.txt

Verificar ficheiro shred.txt alterado

./wc --help

Uso: ./wc [-l] [-w] [-c] [--help] arquivo

Opções:

- l Contar o número de linhas.
- w Contar o número de palavras.
- c Contar o número de caracteres.
- help Exibe esta mensagem de ajuda e sai.

./wc -l words.txt

Linhas: 17

Bibliografia

<https://github.com/coreutils/coreutils>

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

https://www.youtube.com/watch?v=962SRfhBcQ0&t=4689s&ab_channel=AntsAreEverywhere

<https://www.geeksforgeeks.org/sorting-algorithms/>

<https://pt.wikipedia.org/wiki/GNU>

<https://cs.lmu.edu/~ray/notes/gasexamples/>

<https://chatgpt.com/>