|  | **FACULDADE DE INFORMÁTICA DE PRES. PRUDENTE**  **SISTEMA DE INFORMAÇÃO** |
| --- | --- |

**BRUNO ALISON DA SILVA MELO**

**GABRIEL BARBOSA SALVADOR**

**JOÃO PEDRO NARCIZO FRANÇA**

**JOÃO VITOR DIAS DA SILVA**

**IMPLEMENTAÇÃO DE ARQUIVOS**

**I-NODE**

Presidente Prudente - SP

2023

**Sumário**

[**1. Introdução 3**](#_heading=h.30j0zll)

[**2. Implementação 3**](#_heading=h.1fob9te)

[**3. Estruturas Utilizadas 5**](#_heading=h.bg765rwp5bt3)

[**4. Instruções do Sistema 12**](#_heading=h.zhoh226gje76)

[**5. Relatórios 26**](#_heading=h.2pixsj1qmsu0)

[**6. Conclusão 28**](#_heading=h.mu5gadscv6qg)

# Introdução

Neste presente trabalho, iremos explorar a implementação de arquivos Inode em sistemas Linux/Unix. Os Inodes são componentes fundamentais para o funcionamento desses sistemas, governando a organização e o gerenciamento de arquivos. Além disso, abordaremos a implementação de diretórios, arquivos, links e a criação de um disco simulado(vetor com tamanho escolhido pelo usuário). Os blocos serão monitorados em uma lista de blocos livres. Este trabalho oferece uma visão profunda sobre os sistemas de arquivos Unix/Linux e sua estrutura interna.

# Implementação

A implementação será conduzida em linguagem de programação C++ para representar o sistema de arquivos I-node. Inicialmente, será empregada uma estrutura de dados estática, ou seja, um vetor, para simular a memória secundária, que equivale ao disco. Cada posição nesse vetor terá a capacidade de armazenar informações, como arquivos, diretórios, I-nodes, I-nodes indiretos, links físicos e links simbólicos. Além disso, cada posição poderá representar um setor defeituoso no disco (bad sector), tornando-a inutilizável para o armazenamento de dados. Caso uma posição ocupada no disco se torne defeituosa, os dados nela contidos serão irremediavelmente perdidos.

Os blocos do disco, correspondentes a cada posição do vetor, terão dois estados possíveis: livres ou ocupados. Para gerenciar esse controle, será utilizada uma estrutura denominada lista ligada de blocos livres. Nesta lista, todos os blocos de disco disponíveis serão registrados como livres, e quando um bloco for requisitado, a lista de blocos livres fornecerá a posição que poderá ser utilizada. Se uma posição previamente ocupada for liberada, ela será reintegrada à lista de blocos livres, tornando-se disponível para futuras alocações. Caso o bloco retirado da lista de blocos livres se encontre no estado “bad”, será imediatamente descartado.

Os arquivos regulares serão simulados ocupando posições de blocos no disco, variando de acordo com o tamanho dos arquivos. Esse tamanho poderá ser definido pelo usuário ou, em alguns casos, será determinado pelo sistema. O tamanho do arquivo será representado ocupando posições no vetor de disco. Por sua vez, os arquivos de diretório ocuparão inicialmente um único bloco no disco, correspondendo a uma única posição no vetor. Um diretório possui entradas de diretório que é representado pelo nome do arquivo e o endereço do Inode relacionado àquele arquivo, a princípio essa entrada de diretório possui dez posições fixas, caso a entrada de diretório seja toda preenchida, será ocupado um outro espaço no disco.

As informações sobre a localização das posições de armazenamento dos arquivos no vetor de disco serão registradas na estrutura do I-node, especialmente nas seções dedicadas aos blocos diretos e indiretos, quando necessário. Dessa forma, o sistema será capaz de rastrear eficientemente a alocação de espaço para arquivos regulares e diretórios, garantindo que a estrutura do disco virtual seja gerenciada com eficiência e precisão

A estrutura do I-node é fundamental, uma vez que representa um conjunto de dados que define um sistema de arquivos. Esses dados desempenham um papel crucial na obtenção de informações essenciais sobre arquivos ou diretórios no sistema de

arquivos. A estrutura do I-node abrange uma gama significativa de informações, incluindo:

Proteção: Este atributo define qual é a estrutura que está sendo referenciado, podendo ser um diretório, um link ou um arquivo regular, as permissões de acesso, determinando quais permissões são concedidas a usuários, grupos e outros. As permissões podem incluir leitura, gravação, execução ou pode estar desabilitado.

Contador de Links Físicos: Esse valor indica o número de entradas no diretório que apontam para o mesmo Inode. Ele é utilizado para controlar as ligações físicas entre os arquivos e diretórios.

Proprietário: Este atributo especifica o usuário que é o proprietário do arquivo ou diretório.

Grupo: Indica o grupo responsável pelo acesso ao qual o arquivo ou diretório pertence.

Data de Criação, Último Acesso e Última Alteração: Esses registros fornecem informações importantes sobre a história do arquivo, incluindo quando ele foi criado, acessado pela última vez e modificado pela última vez.

Tamanho Original do Arquivo em Bytes: Este atributo determina o tamanho inicial do arquivo, medido em bytes.

Após esses atributos, a estrutura do I-node inclui os chamados 'blocos diretos'. São cinco posições que apontam para os blocos correspondentes no disco (representados no vetor). No entanto, caso esses blocos diretos não sejam suficientes para armazenar o arquivo ou diretório em questão, utilizamos uma estrutura adicional chamada 'bloco indireto'. Existem três tipos de blocos indiretos:

Bloco Indireto Simples: Adiciona oito posições extras para armazenamento de blocos no disco.

Bloco Indireto Duplo: Estende a capacidade de armazenamento ao adicionar níveis que permite o acesso a mais blocos.

Bloco Indireto Triplo: Caso as duas estruturas anteriores não sejam suficientes, o bloco indireto triplo oferece uma extensão ainda maior ao sistema, fornecendo acesso a uma quantidade significativa de blocos no disco. Caso essa estrutura ainda não atenda todos os blocos que representam o arquivo, é criada uma referência para um outro Inode.

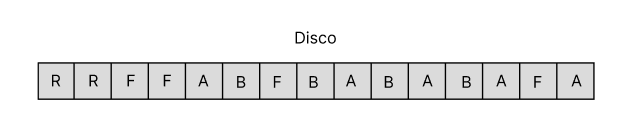
No sistema de arquivos em questão, oferecemos suporte à utilização de links para arquivos e diretórios, permitindo que os usuários tirem proveito de duas variantes de links:

Link Físico: Essa modalidade de link estabelece uma referência direta ao bloco I-node do arquivo. Em outras palavras, os blocos são associados a uma pequena estrutura de dados conectada ao arquivo, enquanto os diretórios apontam apenas para essa estrutura de dados. Isso significa que vários usuários podem compartilhar o mesmo arquivo através dessa ligação direta, proporcionando um mecanismo eficaz de compartilhamento de recursos.

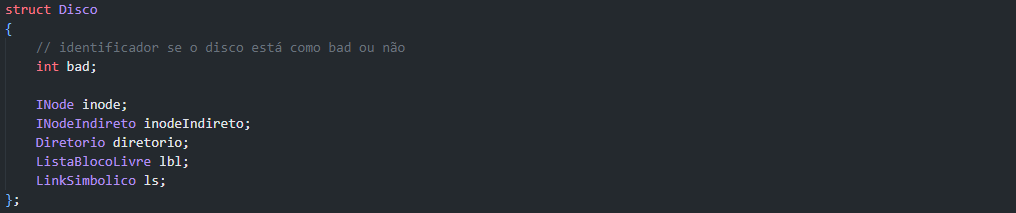
Link Simbólico: Por outro lado, os links simbólicos oferecem um atalho direto para um arquivo ou diretório, representando uma maneira conveniente de acessar rapidamente recursos no sistema. Eles são, na verdade, caminhos simples que apontam para o local real do arquivo ou diretório, fornecendo uma forma prática de navegação no sistema de arquivos.

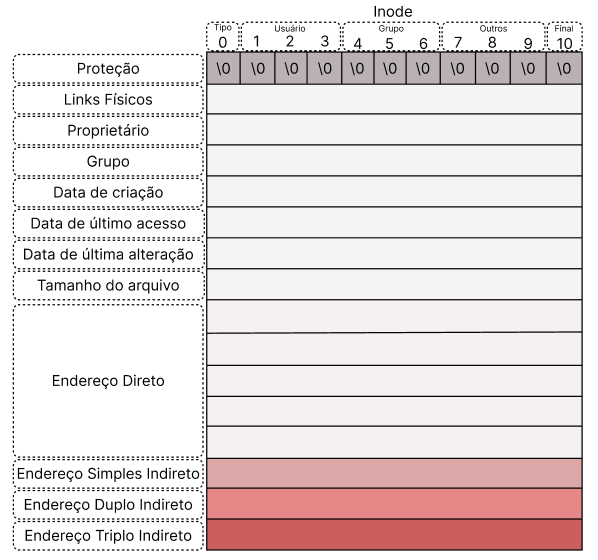
# Estruturas Utilizadas

Para representar o sistema de arquivos, adotamos uma abordagem baseada em uma estrutura estática em forma de vetor. Essa estrutura será responsável por representar todos os blocos que compõem um disco virtual. O tamanho dessa estrutura, ou seja, a quantidade de blocos disponíveis, será escolhido pelo usuário, permitindo flexibilidade na definição do espaço de armazenamento disponível.



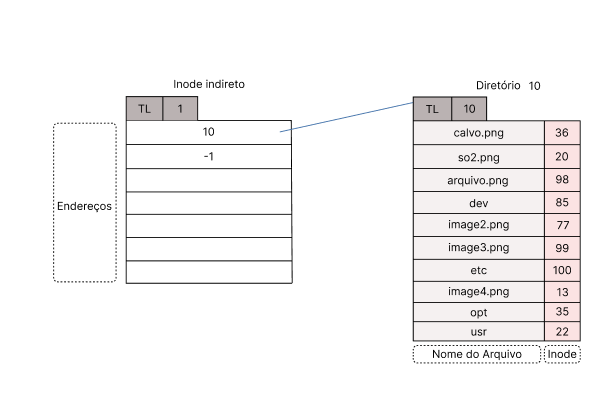
Cada posição no disco pode conter as seguintes estruturas: Inode, Inode Indireto, Diretório, Lista de Blocos Livres, Link Simbólico e um inteiro representando um setor defeituoso (bad)

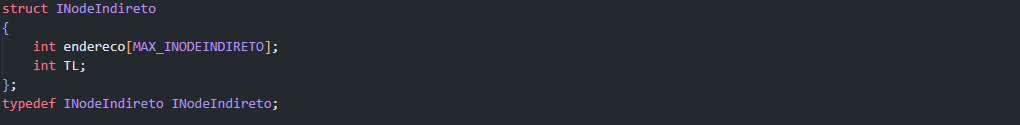
A estrutura do I-node desempenha um papel fundamental, representando um conjunto de dados que define integralmente o sistema de arquivos. Esta estrutura abriga atributos essenciais de um arquivo, incluindo permissões (Proteção), contador de links físicos, proprietário, grupo, registros de data (Criação, Último Acesso e Última Alteração) e tamanho original do arquivo em bytes. Além disso, o I-node também registra a localização física dos arquivos no disco, utilizando bloco direto e uma estrutura conhecida como I-node Indireto (com modalidades Simples, Duplo e Triplo), permitindo um gerenciamento eficiente do armazenamento dos dados



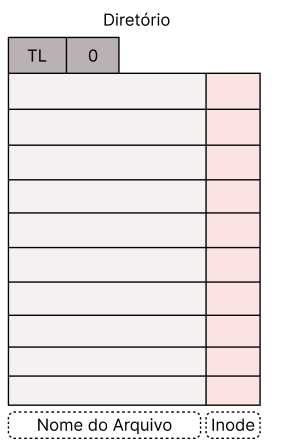
# 

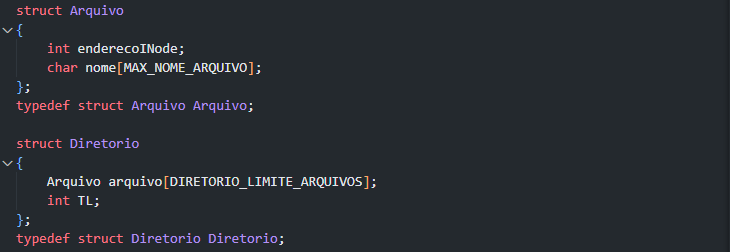
O I-node Indireto desempenha um papel crucial como uma extensão que possibilita o registro da localização física dos arquivos no disco. Essa estrutura oferece espaço adicional ao I-node, com a capacidade de armazenar até oito endereços de blocos do disco correspondentes ao arquivo.





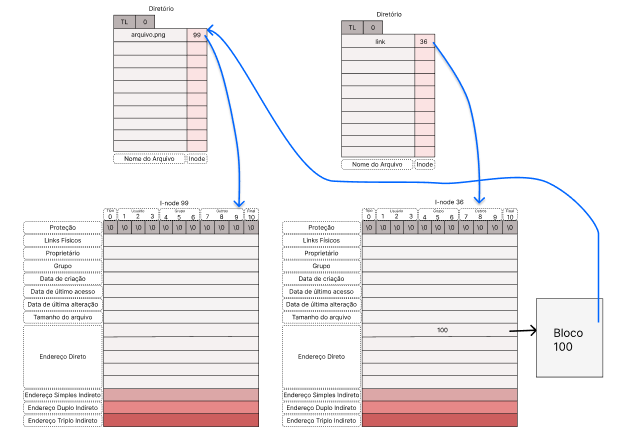
A estrutura de um diretório é composta por entradas de diretório, cada uma com um tamanho de até dez entradas. Essas entradas podem armazenar informações sobre arquivos regulares ou diretórios, incluindo o nome do arquivo e o número do I-node que o representa. Quando as dez posições do diretório são preenchidas, o diretório alocará espaço adicional em outro bloco direto, e em caso de esgotamento dos blocos diretos, recorrerá aos blocos indiretos para acomodar mais entradas.

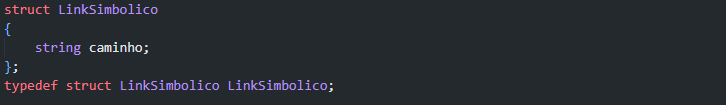




Link Simbólico (Soft Link)

Os links simbólicos oferecem um atalho direto para um arquivo ou diretório, representando uma maneira mais rápida de acessar os recursos no sistema. Eles são, na verdade, caminhos simples que apontam para o local real do arquivo ou diretório, fornecendo uma forma prática de navegação no sistema de arquivos. A estrutura do link simbólico conterá uma string que representa o caminho de origem que será salvo deste link.





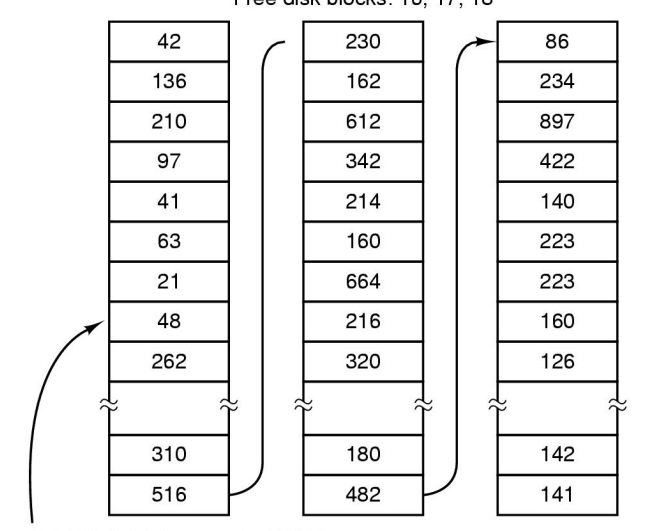
Link Físico

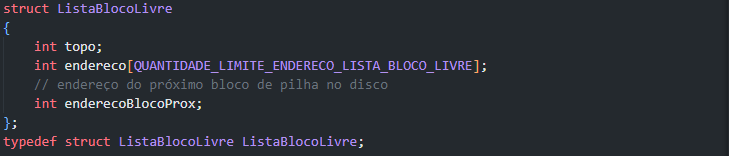
Essa modalidade de link estabelece uma referência direta ao bloco I-node do arquivo. Essa estrutura permite que em uma entrada de diretório do link de destino possa conter o mesmo número do Inode do arquivo de origem em que esta ação foi realizada.



Lista Ligada de Blocos Livres.

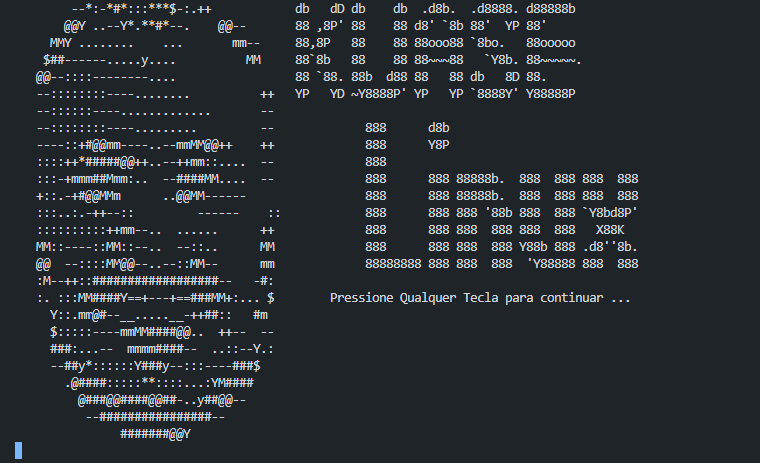
Para monitorar os blocos livres do disco, será utilizada uma estrutura de lista encadeada de blocos livres. Todos os blocos de disco disponíveis serão empilhados na lista, e quando um bloco for requisitado, a lista de blocos livres fornecerá a posição do topo que poderá ser utilizada. Se uma posição previamente ocupada for liberada, ela será empilhada novamente à lista de blocos livres, tornando-se disponível para futuras alocações.





# Instruções do Sistema

**Inicialização do Sistema:**

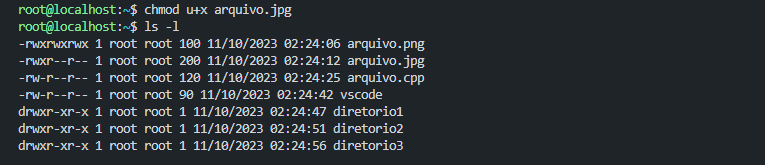
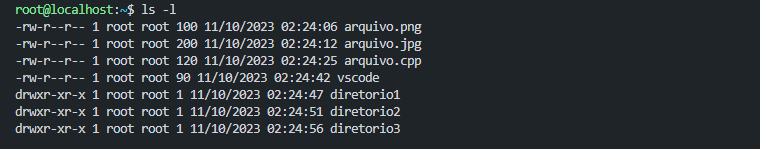
****

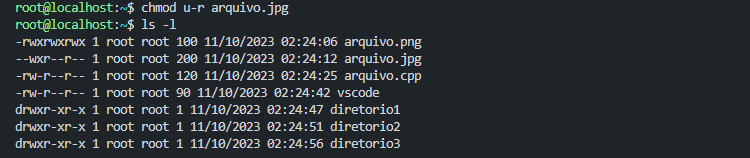
**Informando a Quantidade de Blocos do Disco:**

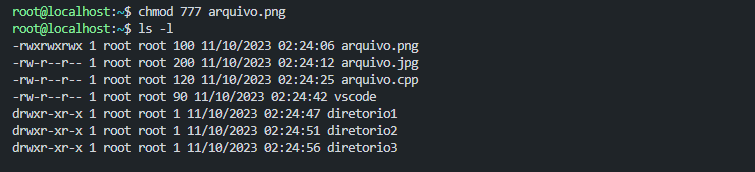
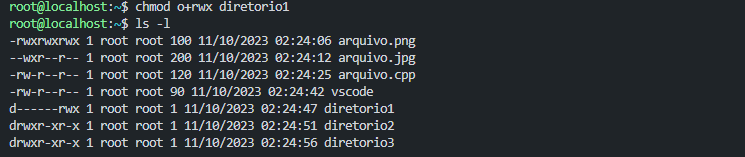
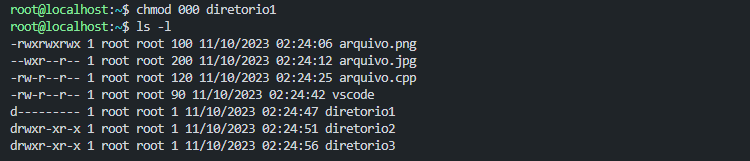
Nesta fase, você deverá especificar o tamanho do disco em termos de quantidade de blocos. Caso não forneça um tamanho, o sistema utilizará o valor padrão, que corresponde a mil blocos.

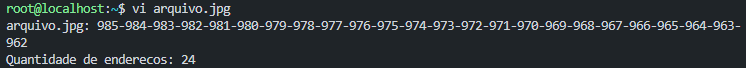
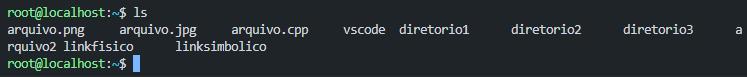


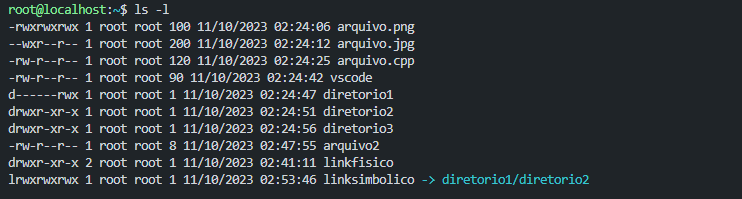
**Comandos Básicos:**

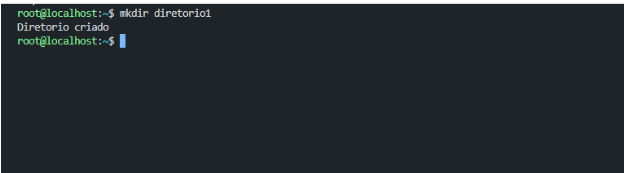
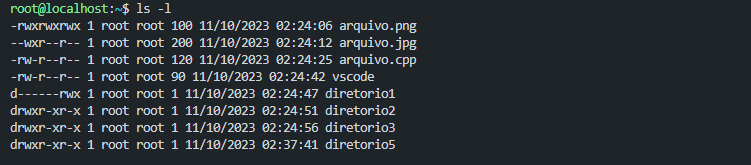
* 1. **chmod (+)(-)ugo RWX:** alterar as permissões de acesso a arquivos e diretórios

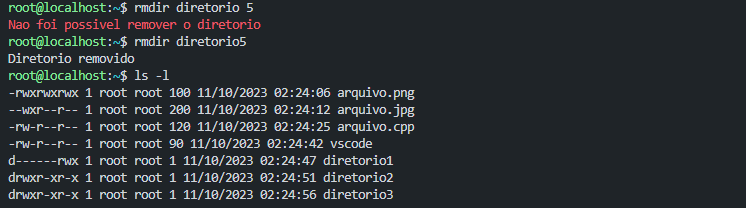


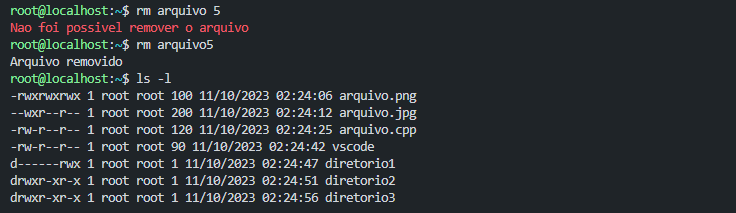
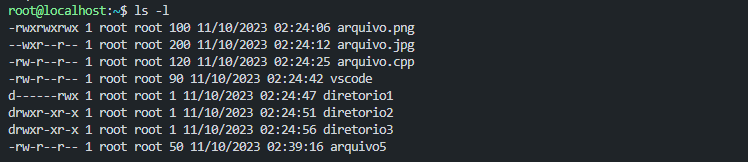
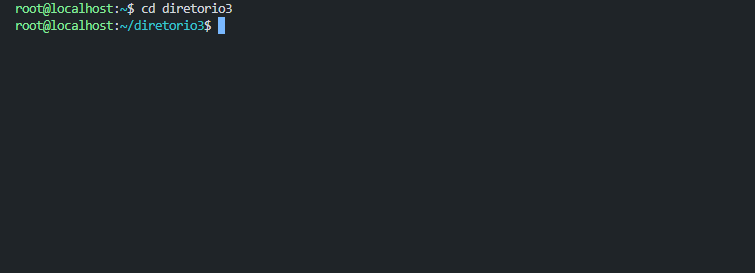
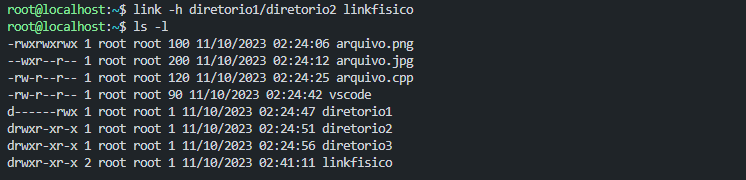


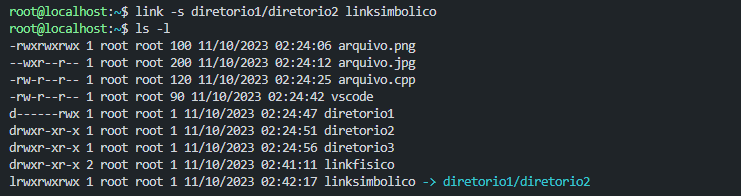
* 1. **vi:** nomeArquivo visualizar um arquivo regular
  2. **ls:** listar os nomes dos arquivos no diretório
  3. **ls -l:** listas os nomes dos arquivos com seus atributos



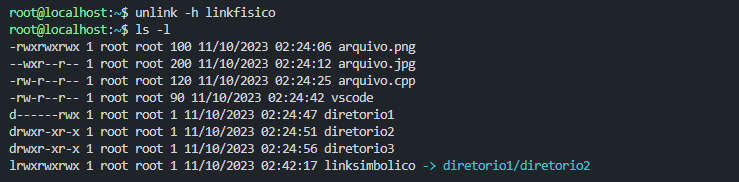
* 1. **mkdir NomeDir:** criar diretórios
  2. **rmdir NomeDir:** deletar direitórios que estejam vazios

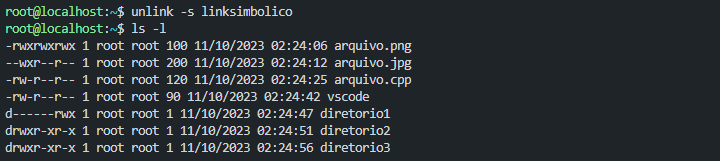


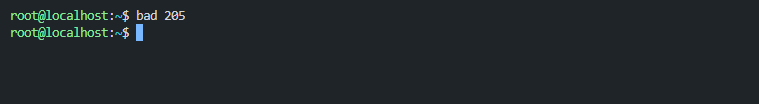
* 1. **rm NomeArq:** deletar aquivos
  2. **cd NomeDir ou . ou .. :** navegar nos diretórios
  3. **link –h nomeArquivoOrigem NomeArquivoDestino:** criar link físico
  4. **link –s nomeArquivoOrigem NomeArquivoDestino:** criar link simbólico

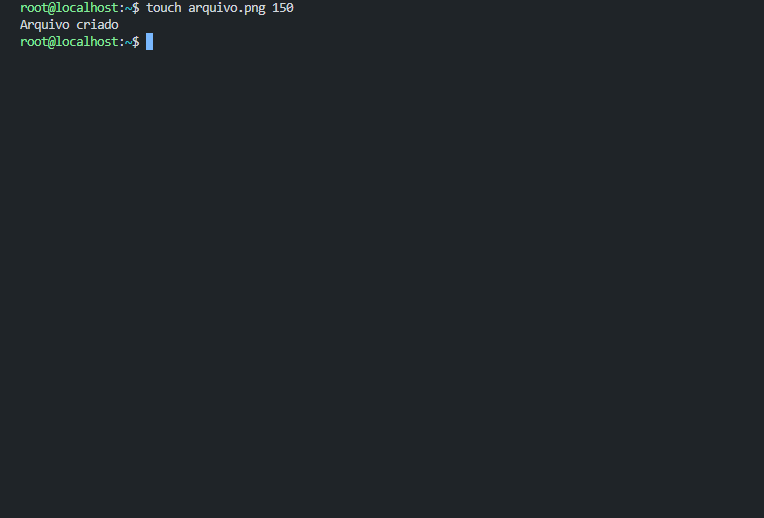
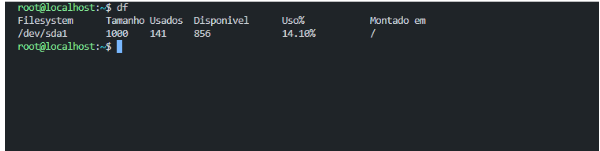
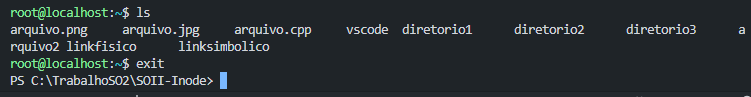
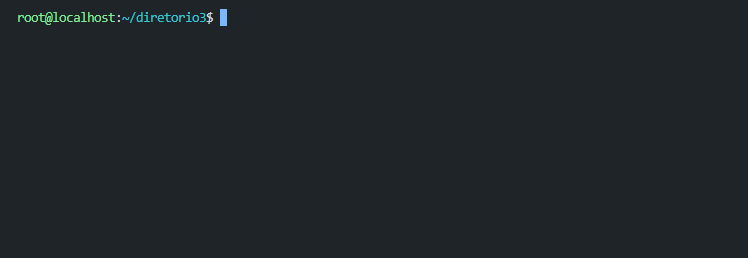
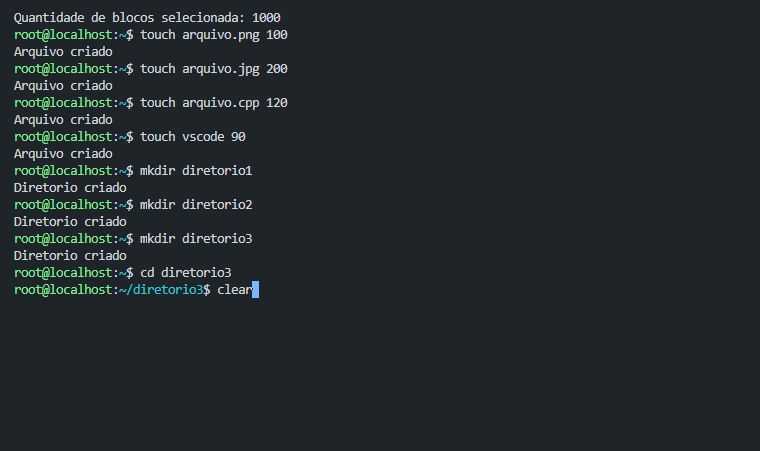


* 1. **unlink –h:** remover link físico

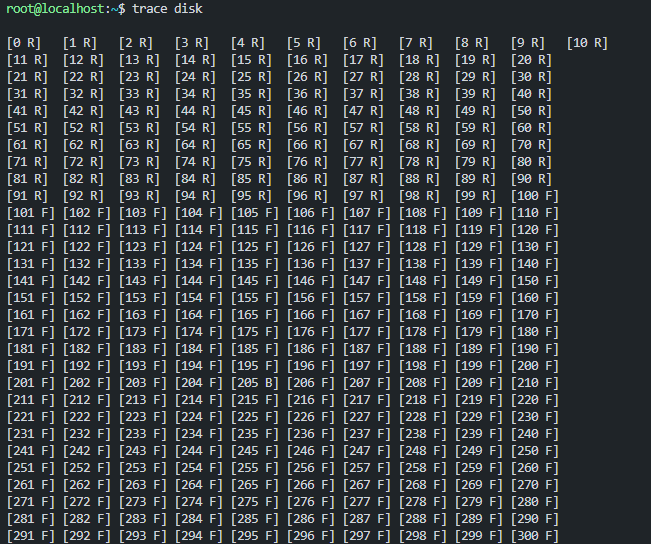


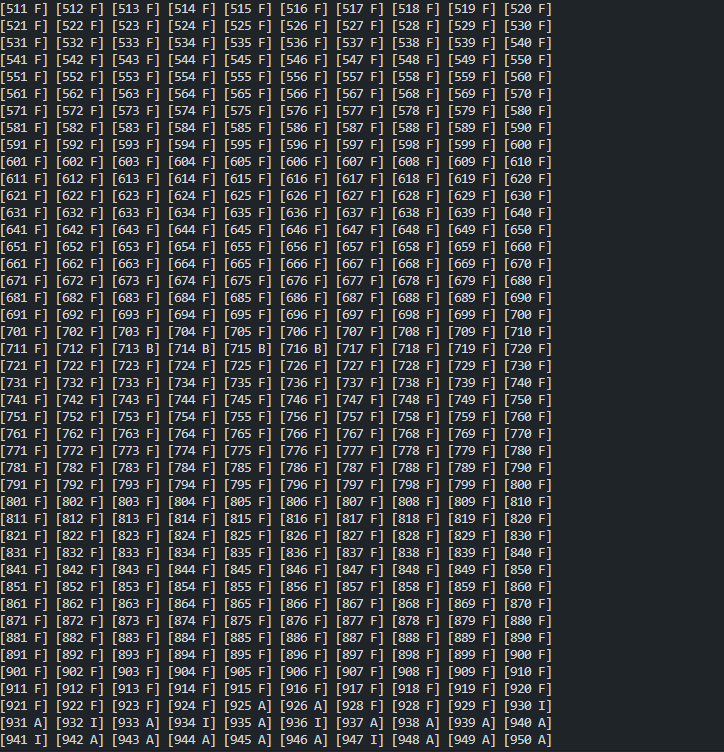
* 1. **unlink –s:** remover link simbólico
  2. **bad numeroBloco:** transformar um bloco em Bad

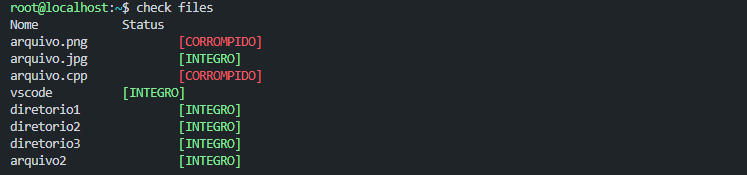
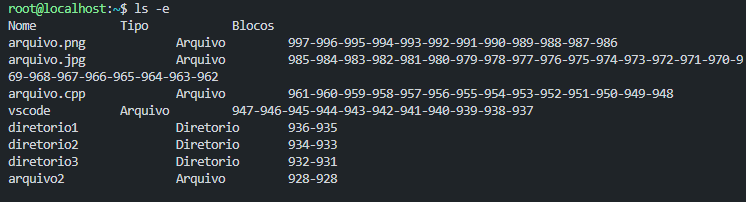
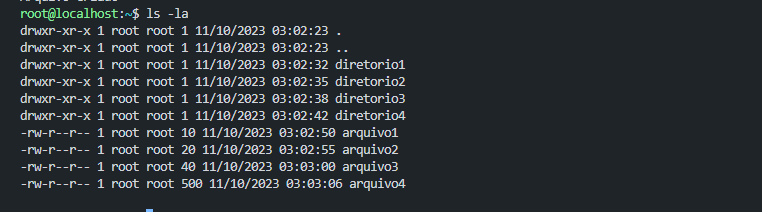
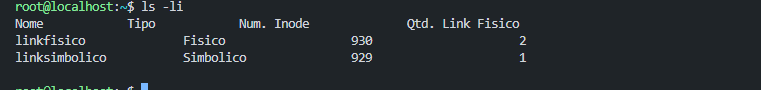


* 1. **touch NomeArquivo TamanhoBytes:** Criar um arquivo Regular
  2. **df:** apresentar em bytes espaço livre e ocupados do disco
  3. **exit:** Finaliza a execução do sistema
  4. **clear:** Limpa a tela

**Comandos de Relatórios:**

* 1. **trace disk:** Apresenta o estado atual de todos os blocos do disco



* 1. **check files:** Realiza uma varredura e exibe o estado do arquivo do disco, se encontra-se corrompido ou íntegro
  2. **lost block:** Apresenta a quantidade de blocos que foram não foram usados (perdidos), são os blocos que não apresentam-se como livres e nem como defeituoso
  3. **max file:** Estima o máximo de blocos que poderão ser utilizados pelo arquivo.
  4. **ls -e:** Apresenta todos os arquivos do diretório atual, seu tipo e os blocos utilizados. Similar a uma visão de lista do Windows Explorer
  5. **ls -li:** Apresenta todos os links do diretório atual, seu tipo, o endereço do inode e a quantidade de links físicos.
  6. **ls -la:** Mostra todos arquivos e seu atributos incluindo os arquivos ocultos com o nome iniciando com '.' na listagem.

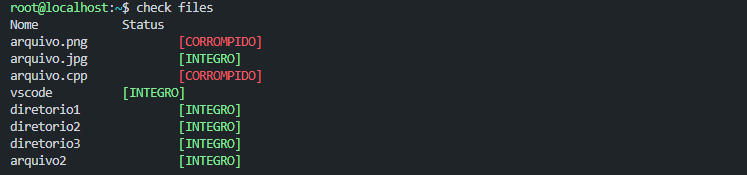
# Relatórios

1. O número de blocos ocupados por um arquivo escolhido por usuário;



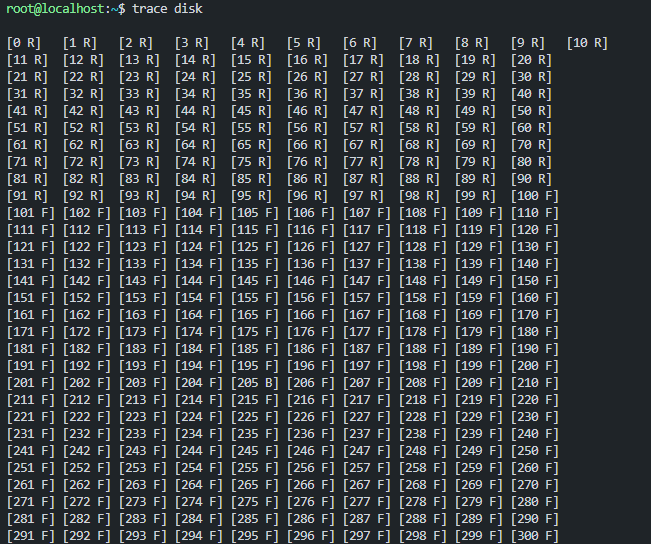
2. O tamanho (em blocos) do maior arquivo que ainda pode ser criado nesse disco; 

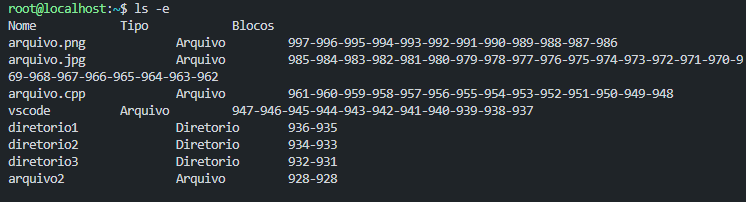
3. Quais arquivos estão íntegros e quais estão corrompidos por blocos defeituosos (badblocks);

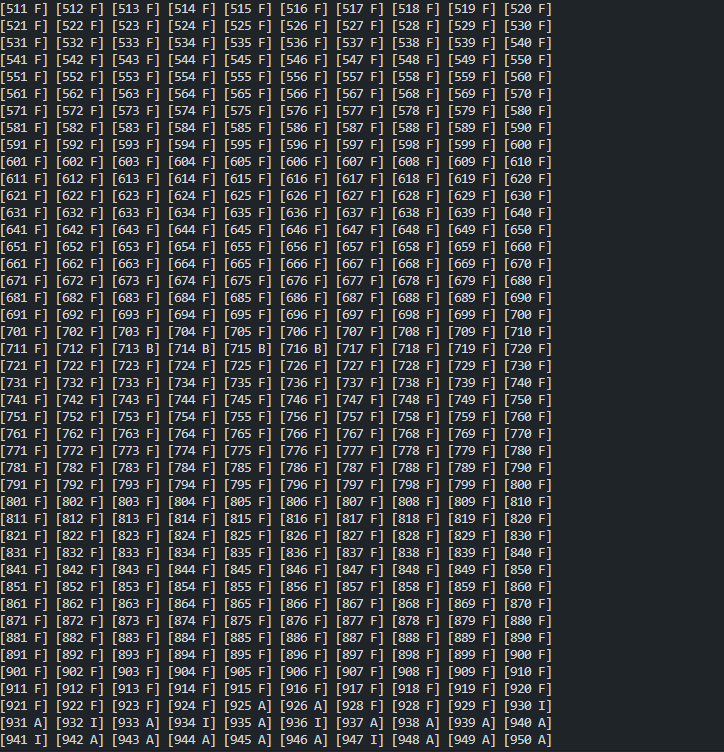


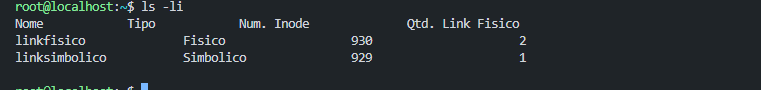
4. Apresentar quantos blocos do disco estão perdidos e quais são eles, ou seja, não são usados por arquivos e nem estão marcados como livres ou defeituosos. Não esqueça de apresentar o espaço de disco perdido em bytes;



5. Imprima todos os blocos em seu estado atual, igual a figura b.

6. Visualize os arquivos e diretórios alocados, apresentando o nome e seus números do bloco correspondentes identificando o tipo. Esta visualização deve ser algo semelhante ao Windows Explorer. Não precisa ser gráfico. 

7. Visualizar a árvore de diretório igual a figura D, apresentado as estruturas com seus atribuídos, nomes e números; 

8. Visualizar os links simbólicos e físicos criados, detalhando as entradas de diretório, inodes e números

# Conclusão

Durante a implementação do trabalho foi possível notar as diferentes peças-chaves na implementação de sistemas de arquivos são essenciais para a organização e a segurança de dados em computadores.

Ao longo de todo o estudo, abordamos o desenvolvimento, configuração e personalização de um ambiente, bem como a importância de compreender seus princípios fundamentais. ficou evidente que os I-Nodes servem como a espinha dorsal de um sistema de arquivos, permitindo a alocação eficiente de espaço em disco, o rastreamento de metadados essenciais, como permissões, timestamps e links rígidos, além de possibilitar a recuperação rápida de dados. Compreender a estrutura e o funcionamento dos I-Nodes é crucial para qualquer profissional de TI que lida com administração de sistemas ou desenvolvimento de software.

Em suma, este trabalho ressalta a relevância dos I-Nodes na implementação de arquivos e sistemas de armazenamento, enfatizando sua organização, segurança e o desempenho de sistemas de computação.