

VINAc – Arquivador com Suporte à Compressão

Pietro Comin
GRR20241955
pietro.comin@ufpr.br

Número de linhas do código: 1989

Descrição

O **VINAc** é um arquivador com suporte à compressão baseado em linha de comando, semelhante a ferramentas como **tar**, **zip** ou **rar**. Seu objetivo é permitir o arquivamento de múltiplos arquivos em um único container (**.vc**), com a opção de compressão individual utilizando o algoritmo LZ.

A estrutura do arquivo **.vc** inclui uma área de diretório responsável por armazenar todos os metadados necessários para manipulação dos membros arquivados.

Funcionalidades

O programa **vinac** oferece suporte às seguintes opções de execução:

- **-ip** ou **-p**: insere membros sem compressão
- **-ic** ou **-i**: insere membros com compressão (LZ)
- **-m membro**: move um membro dentro do arquivo
- **-x**: extrai membros (ou todos)
- **-r**: remove membros do arquivo
- **-c**: lista o conteúdo do arquivo

Estrutura do Projeto

```
A1/
├── include/
│   ├── aux.h
│   ├── lz/
│   │   └── lz.h
│   ├── options.h
│   ├── types.h
│   ├── utils.h
│   └── vina.h
├── src/
│   ├── aux.c
│   ├── main.c
│   ├── options.c
│   ├── utils.c
│   ├── vina.c
│   └── lz/
│       └── lz.c
├── build/
│   └── objects/ (gerado durante a compilação)
├── login/
│   └── vinac (executável gerado)
├── makefile
├── A1 - O Arquivador VINAc.pdf (enunciado do trabalho)
├── texto.txt
├── texto2.txt
├── texto3.txt
└── texto4.txt
```

Compilação

Para compilar o projeto, utilize:

```
make
```

Para limpar os arquivos gerados:

```
make clean
```

Estruturas de Dados e Algoritmos

Estruturas de Dados

- `struct arquivo`: armazena os metadados de cada membro do arquivo:
 - nome
 - UID
 - tamanho original
 - tamanho comprimido
 - ordem
 - offset no arquivo
 - data de modificação
- `struct diretorio`: vetor de ponteiros para `arquivo`, gerencia todos os membros do container.

Funções Relevantes

A manipulação de arquivos variáveis em tamanho e posição foi um dos principais desafios. As seguintes funções foram criadas para garantir consistência:

- `int move(...)`: move um bloco dentro do arquivo
- `int move_sequencial(...)`: move membros em sequência, evitando sobrescrita
- `int insere_membro_arq(...)`: insere um novo membro
- `int comprime_arquivo(...)`: comprime com LZ e atualiza metadados
- `int descomprime_arquivo(...)`: descomprime e restaura o conteúdo

Decisões de Implementação

- A compressão LZ é usada apenas se houver ganho no tamanho.
- A área de diretório é mantida em RAM para facilitar o controle.
- Os dados dos membros são manipulados diretamente no disco.
- O código é modularizado para facilitar testes e manutenção.

Dificuldades Encontradas

- **Deslocamentos binários:** o gerenciamento de offsets exigiu lógica cuidadosa e criação de funções específicas, já que no máximo um membro pode ser deslocado por vez, evitando uso de buffers muito grandes.
- **Compressão seletiva:** inserir membros apenas quando a compressão for vantajosa envolveu lógica condicional.
- **Atualização de metadados:** todas as operações exigem atualização precisa de offsets e ordem.
- **Atomicidade:** foi necessário evitar o uso de arquivos temporários, otimizando a leitura e escrita direta em disco.

Bugs Conhecidos

- **Warning no Valgrind:** alertas sobre bytes não inicializados. Após várias sessões de depuração, acredita-se que seja inofensivo.
- **Testes limitados:** ainda não há testes com arquivos muito grandes ou com nomes inválidos.

Contato

Para dúvidas ou sugestões, entre em contato: pietro.comin@ufpr.br