

# Arquivo README do arquivador Vina++

**Autor:** Gabriel Miyazato Vizeu Ferreira (GRR 20211770)

**Nota:** Para funcionamento do programa, deve-se adicionar os arquivos archiver.c e archiver.h na pasta ./src/ do projeto.

**Resumo:** O programa é um arquivador de arquivos com as seguintes funções: Insert (-i), Insert\_a (-a), Move (-m target), Extract (-x), Remove (-r), List (-c) e Help (-h)

## Bibliotecas

O programa utiliza-se de 4 diferentes bibliotecas, sendo elas: "archiver.h", "arguments.h", "vina\_options.h" e "vina\_aux\_functions.h".

**archiver.h** é uma biblioteca de manipulação de bytes, com funções que inserem, removem, movem e extraem bytes de um arquivo.

**arguments.h** é uma biblioteca que checa se os argumentos da chamada do programa vina++ estão corretos.

### estruturas de dados:

```
typedef enum { INSERT, INSERT_A, MOVE, EXTRACT, LIST, HELP } option_t
```

tipo enumerado com a opção do programa

```
typedef struct { option_t option; char *archive, **files, *target; int file_count } arguments_t
```

struct dos argumentos

### funções:

```
arguments_t *check_arguments(int argc, char **argv)
```

função que recebe argc e argv, e retorna um ponteiro de arguments\_t com os dados de entrada do programa.

```
void destroy_arguments(arguments_t *arg)
```

função que recebe um ponteiro de arguments\_t e libera a memória alocada para ele.

**vina\_options.h** é a biblioteca com a lógica de cada opção do programa vina++, com 7 funções, uma para cada opção do programa.

### funções:

```
int option_insert(arguments_t *arg)
```

função que recebe um ponteiro de arguments\_t com os dados de entrada do programa, insere os arquivos selecionados no arquivador e se o arquivo já esteja lá, deve ser substituído. Retorna 1 caso seja bem sucedido, e 0 caso não seja.

int option\_insert\_a(arguments\_t \*arg)

função que funciona da mesma forma que a option\_insert, mas a substituição dos arquivos ocorre somente se os arquivos passados como parâmetros sejam mais recentes que os arquivados.

int option\_move(arguments\_t \*arg)

função que recebe um ponteiro de arguments\_t com os dados de entrada do programa, move o arquivo selecionado para imediatamente depois do arquivo target (indicado na execução do programa) no arquivador. Retorna 1 caso seja bem sucedido, e 0 caso não seja.

int option\_extract(arguments\_t \*arg)

função que recebe um ponteiro de arguments\_t com os dados de entrada do programa, extrai os arquivos selecionados do arquivador e se o arquivo não esteja lá, é indicado em stderr. Retorna 1 caso seja bem sucedido, e 0 caso não seja

int option\_remove(arguments\_t \*arg)

função que recebe um ponteiro de arguments\_t com os dados de entrada do programa e remove os arquivos selecionados do arquivador. Retorna 1 caso seja bem sucedido, e 0 caso não seja

int option\_list(arguments\_t \*arg)

função que recebe um ponteiro de arguments\_t com os dados de entrada do programa e lista os arquivos dentro do arquivador. Retorna 1 caso seja bem sucedido, e 0 caso não seja.

int option\_help(arguments\_t \*arg)

função que exibe na tela instruções de como utilizar o programa.

**vina\_aux\_functions.h** é uma biblioteca com funções auxiliares e estruturas de dados necessárias para as funções da biblioteca “vina\_options.h”.

#### **estruturas de dados:**

```
typedef struct { unsigned int name_size; char *name; uid_t user_id; mode_t permissions;
                off_t size; time_t modification_date; unsigned int archive_order;
                unsigned long int position; } member_data_t
struct com as informações dos arquivos-membros.
```

```
typedef struct { unsigned int directory_size; int file_count; member_data_t **members }
archive_data_t
struct com informações sobre diretório do arquivador.
```

#### **funções:**

void initialize\_archive(FILE \*archive)

função que inicializa um arquivador vazio com informações iniciais do diretório

archive\_data\_t \*get\_archive\_data(FILE \*archive)

função que pega as informações do diretório de um arquivador

```
member_data_t *get_member_data(FILE *member, char *member_name, unsigned int  
                                archive_order, unsigned int position)
```

função que pega as informações de um arquivo-membro, e retorna um ponteiro de member\_data\_t com essas informações.

## **Funcionamento do Programa**

Ao executar o programa, é chamada a função check\_entries(arguments\_t \*arg), da biblioteca “arguments.h”, que serve apenas para checar se o programa foi executado corretamente, verificando se os argumentos estão corretos de acordo com a opção selecionada.

Logo em seguida, há um switch (case) que dependendo da opção selecionada do programa, o ponteiro de função int (\*option\_function)(arguments\_t\*) aponta para a respectiva função da opção selecionada.

Após esse switch (case) essa função é chamada, e caso a função da opção retorne 0, a função main do programa retorna 2. Caso contrário, retorna 0.

## **Lógica das funções principais**