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 member_data_size, 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_position, directory_size; int file_count; member_data_t **members } archive_data_t
struct com insformações sobre diretório do arquivador.

funções:

void initialize_archive(FILE *archive);
inicializa um arquivador vazio

archive_data_t *get_archive_data(char *archive_name, FILE *archive);
pega as informações sobre o diretório do arquivador

member_data_t *get_member_data(FILE *member, char *member_name, unsigned int archive_order, unsigned int position);
insere as informações sobre um membro indicado nos parâmetros na estrutura de dados member_data_t

member_data_t *get_member_data_from_archive(FILE *archive);
pega as informações sobre um membro que está dentro do arquivador

void put_member_data(member_data_t *member_data, FILE *archive);
adiciona ao diretório do arquivador as informações sobre um membro inserido

void destroy_archive_data(archive_data_t *archive_data);
libera a memória alocada por archive_data

int file_is_in_archive(char *filename, archive_data_t *archive_data);
verifica se o arquivo indicado por filename está no diretório

void extract_file(FILE *archive, FILE *file, unsigned int total_bytes, unsigned int position);
extrai os bytes de um membro de archive e insere em file

void update_directory(FILE *archive, archive_data_t *archive_data, unsigned int old_position, unsigned int new_position, unsigned int old_size, unsigned int new_size, int new_file_count);
atualiza o diretório do arquivador archive

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