Diogo Cruz Alpendre (2191747)

Declaro sobre compromisso de honra que o presente trabalho (código, relatório e afins) foi integralmente realizado por mim, sendo que as contribuições externas se encontram claramente e inequivocamente identificadas no próprio código. Mais se declara que o estudante acima identificado não disponibilizou o código ou partes dele a terceiros.

Estado de funcionamento

Modo 1 / Sem modo

- Verificar se ficheiro existe: funcional

Foi usada a função strtok da biblioteca <string.h> do C, em que é posível dividir uma string em sub-strings, usando um delimitador para separar as mesmas (no caso do projeto uma vírgula). Usando um ciclo while, separou-se a string de listagem de ficheiros e chamou-se uma função dentro do ciclo para processar cada ficheiro listado (verificar e contar bytes).

- Processamento de ficheiros (-f/--file): funcional

Criou-se um vetor com 256 elementos (de 0 a 255) para guardar as ocurrências dos bytes. Dentro da leitura do conteúdo do ficheiro, usou-se um for para comparar o caracter a ser lido, e caso o caracter correspondece a uma posição do for, adiciona-se uma ocurrência. Depois imprime as contagens de acordo ao parâmetro inserido (normal ou compacto).

- Processamento de diretorias (-d/--dir): funcional, mas com problemas (stack aborted quando imprimo os ficheiros na diretorias)

- Imprime ficheiros listados com a opção -f após processamento:

- Normal: funcional / Com tempo de execução: funcional

  Valores passados para a função de “print” por argumentos na função, e usando um ciclo for, imprimiu-se a contagem.

- Compacto: funcional / Com tempo de execução: funcional

  Igual à opção normal, mas não usando quebras de linha.

- Discreto: funcional, mas com problemas (só consegue ler um ficheiro apesar de serem listados vários);

- Output para ficheiro de ficheiros:

- Normal: funcional / Com tempo de execução: funcional

Cria-se um ficheiro (FILE *fptr usando API de alto nível), caso o mesmo não exista, e escreve-se o conteúdo em formato normal para o ficheiro.

- Compacto: funcional / Com tempo de execução: funcional

  O mesmo da opção normal, mas sem quebras de linha.

- Discreto: não funcional devido ao problema de leitura encontrado na opção discrete sem output.

- Output para ficheiro de ficheiros em diretorias:

- Normal: não funcional / Com tempo de execução: não funcional
- Compacto: não funcional / Com tempo de execução: não funcional
- Discreto: não funcional devido ao problema de leitura encontrado na opção discrete sem output.

Para a contagem do tempo de execução, foi utilizada uma struct clock_t da biblioteca <time.h> que permitiu criar uma variável para iniciar o contador no inicio do programa (no ficheiro main.c antes da verificação de argumentos) e parar nas funções corretas, usando outra variável para parar o relógio.

Modo 2

- Verificar se ficheiro existe: funcional

Foi usada a função strtok da biblioteca <string.h> do C, em que é posível dividir uma string em sub-strings, usando um delimitador para separar as mesmas (no caso do projeto uma vírgula). Usando um ciclo while, separou-se a string de listagem de ficheiros e chamou-se uma função dentro do ciclo para processar cada ficheiro listado (verificar e contar bytes).

- Processamento de ficheiros (-f/--file): funcional

Criou-se um vetor com 65536 elementos (de 0 a 65535) para guardar as ocurrências dos bytes. Dentro da leitura do conteúdo do ficheiro, usou-se um for para comparar o caracter a ser lido, e caso o caracter correspondece a uma posição do for, adiciona-se uma ocurrência. Depois imprime as contagens de acordo ao parâmetro inserido (normal ou compacto).

- Processamento de diretorias (-d/--dir): o mesmo problema encontrado com o modo 1.

- Imprime ficheiros listados com a opção -f após processamento:

- Normal: funcional / Com tempo de execução: funcional

  Valores passados para a função de “print” por argumentos na função, e usando um ciclo for, imprimiu-se a contagem.

- Compacto: funcional / Com tempo de execução: funcional

  Igual à opção normal, mas não usando quebras de linha.

- Discreto: não funcional com problemas, de vido ao problema listado no modo 1.

- Output para ficheiro de ficheiros:

- Normal: funcional / Com tempo de execução: funcional

Cria-se um ficheiro (FILE *fptr usando API de alto nível), caso o mesmo não exista, e escreve-se o conteúdo em formato normal para o ficheiro.

- Compacto: funcional / Com tempo de execução: funcional

  O mesmo da opção normal, mas sem quebras de linha.

- Discreto: não funcional devido ao problema listado no modo 1.

- Output para ficheiro de ficheiros em diretorias:

- Normal: não funcional / Com tempo de execução: não funcional
- Compacto: não funcional / Com tempo de execução: não funcional
- Discreto: não funcional devido ao problema listado no modo 1.

A contagem do tempo de execução do programa foi feita da mesma maneira que no modo 1.

Modo 4: não implementado

Outros requisitos

- Menu de ajuda: funcional

A opção –help/-h vem predifinada no gengetopt, portanto coloquei uma descrição pormenorizada do funcionamento dos argumentos.

- Pesquisa padrão: não funcional

Notas:

- Foram usadas as ferramentas Git e Github no projeto para gestão de versões.