Sprawozdanie

Stanislau Yarmats i Ivan Rusinovich

11.04.2023

Cel projektu

Celem projektu jest zrobienie programu do kompresji plików (binarnych) za pomocą algorytmu Huffmana w języku C. Algorytm Huffmana bezstratnie kompresuje pliki za pomocą kodów prefiksowych.

Specyfikacja

W przypadku kompresora potrzebny jest plik, który chcemy skompresować. W przypadku dekompresora potrzebne są plik skompresowany wraz z tabelą kodów (w postaci pliku). Plik z tabelą kodów ma nazwę „kody”. W tabeli kodów znajdują się: liczba nadmiernych bitów, liczba kodowanych symboli, stopień kompresji, symboli oraz ich kody. Są 2 stopnie kompresji: 8 bit (opcja 1) oraz 16 bit (opcja 2). W przypadku kodowania plików tekstowych za pomocą drugiej opcji plik musi być w postaci LF.

Struktura pliku kody:

<liczba nadmiernych bitów> <liczba kodowanych symboli> <stopień kompresji>

<symbol> <kod>

…

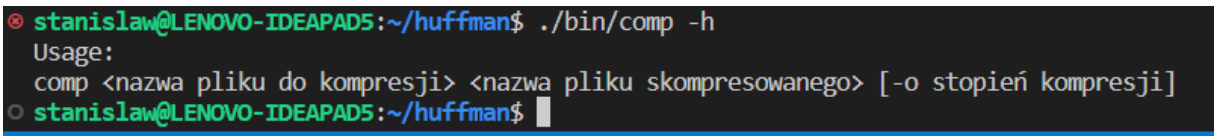
<symbol> <kod>

W pliku minheap.c znajdują się funkcje potrzebne do zrobienia drzewa Huffmana. W pliku huffman.c znajdują się funkcje potrzebne do kompresji. W pliku decomp.c znajdują się funkcje potrzebne do dekompresji. Plik comp.c korzysta z funkcji z plików minheap.c oraz huffman.c.

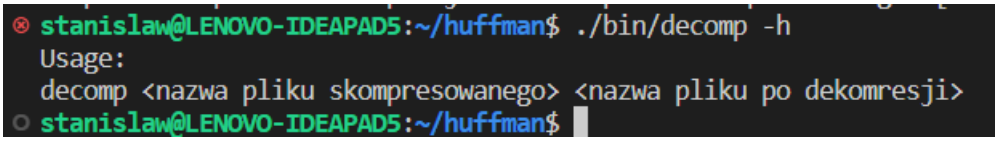
Uruchomienie programu

Żeby skompilować program należy wpisać make. Pliki wykonywalne będą znajdować się w folderze bin.

Menu help kompresora:

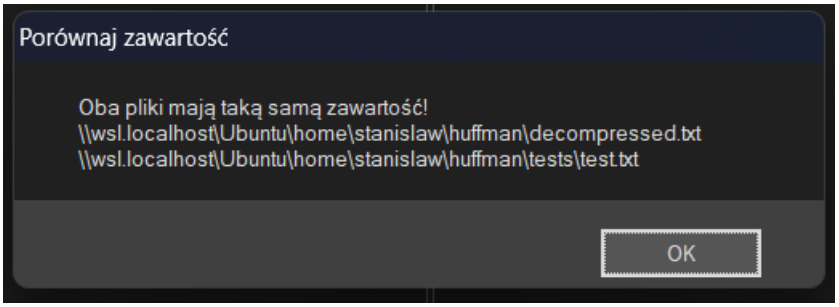


Menu help dekompresora:



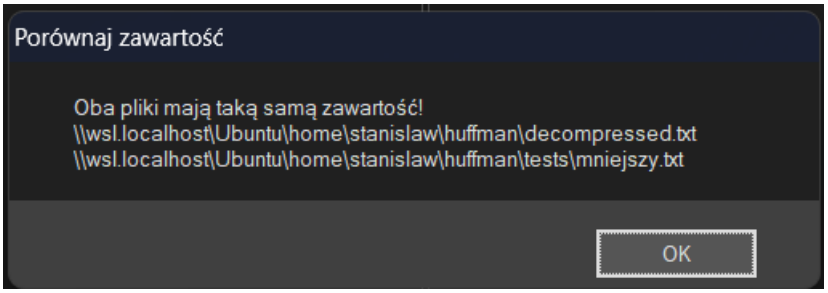
Przykłady działania programu:

```
stanislaw@LENOVO-IDEAPAD5:~/huffman$ cat tests/test.txt
aaaabbcd0
stanislaw@LENOVO-IDEAPAD5:~/huffman$ ./bin/comp tests/test.txt compressed.bin -o 1
stanislaw@LENOVO-IDEAPAD5:~/huffman$ ./bin/decomp compressed.bin decompressed.txt
stanislaw@LENOVO-IDEAPAD5:~/huffman$
```



Inny plik, inny stopień kompresji:

```
stanislaw@LENOVO-IDEAPAD5:~/huffman$ ./bin/comp tests/mniejszy.txt compressed.bin -o 2
stanislaw@LENOVO-IDEAPAD5:~/huffman$ ./bin/decomp compressed.bin decompressed.txt
stanislaw@LENOVO-IDEAPAD5:~/huffman$
```



Wynik działania valgrind:

```
stanislaw@LENOVO-IDEAPAD5:~/huffman$ valgrind --leak-check=full --show-leak-kinds=all -s ./bin/comp tests/mniejszy.txt compressed.bin -o 2
==18366== Memcheck, a memory error detector
==18366== Copyright (C) 2002-2017, and GNU GPL'd, by Julian Seward et al.
==18366== Using Valgrind-3.18.1 and LibVEX; rerun with -h for copyright info
==18366== Command: ./bin/comp tests/mniejszy.txt compressed.bin -o 2
==18366==
==18366==
==18366== HEAP SUMMARY:
==18366==     in use at exit: 0 bytes in 0 blocks
==18366==   total heap usage: 986 allocs, 986 frees, 32,038,146 bytes allocated
==18366==
==18366== All heap blocks were freed -- no leaks are possible
==18366==
==18366== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)
stanislaw@LENOVO-IDEAPAD5:~/huffman$ valgrind --leak-check=full --show-leak-kinds=all -s ./bin/decomp compressed.bin decompressed.txt
==18477== Memcheck, a memory error detector
==18477== Copyright (C) 2002-2017, and GNU GPL'd, by Julian Seward et al.
==18477== Using Valgrind-3.18.1 and LibVEX; rerun with -h for copyright info
==18477== Command: ./bin/decomp compressed.bin decompressed.txt
==18477==
==18477==
==18477== HEAP SUMMARY:
==18477==     in use at exit: 0 bytes in 0 blocks
==18477==   total heap usage: 1,945 allocs, 1,945 frees, 32,117,151 bytes allocated
==18477==
==18477== All heap blocks were freed -- no leaks are possible
==18477==
==18477== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)
stanislaw@LENOVO-IDEAPAD5:~/huffman$
```

Dekompresor działa wolno, bo głównym celem projektu jest kompresor, nie skupialiśmy na nim dużo uwagi. W przypadku dekompresji dużych plików (np. lotr_ascii.txt) trzeba trochę poczekać.