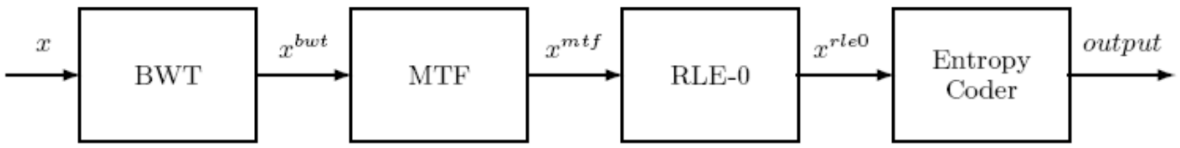
Kódování a komprese dat 2017/2018: Projekt č. 2

Název: Komprese textových souborů s využitím Burrows-Wheelerovy transformace
Datum odevzdání: nejpozději do 06. 05. 2018 prostřednictvím IS FITu.
Forma odevzdání: elektronicky - soubory v archívu ZIP
Počet bodů: max. 30
Poznámka: součástí zadání jsou testovací soubory v adresáři „test“
Dotazy: simekv@fit.vutbr.cz

Zadání:
V jazyce C/C implementujte knihovnu a aplikaci pro kompresi a dekompresi textových souborů s využitím BWT transformace (Burrows-Wheeler). Aplikace bude mít následující strukturu:



Jednotlivé kroky na tomto obrázku představují dílčí fáze zpracování vstupního bloku dat - BWT transformací, překódování řetězce znaků operací Move-To-Front (MTF), efektivní reprezentaci delších sledů čísel metodou RLE a konečně entropické zakódování.

Pro poslední krok, tedy entropické zakódování, lze zvolit statický či adaptivní Huffmanův kód nebo aritmetický kodér. Funkce implementovaného řešení bude ověřena na přiloženém souboru.

Podrobné pokyny k vypracování:

a) implementace knihovny:
V rozhraní knihovny budou povinně definovány dva prototypy funkcí pro kódování a dekódování. Dále bude definován typ struktury udávající velikost kódovaného a nekódovaného souboru dat v bytech.

Datový typ záznamu o (de)kódování bude mít tvar:

```
typedef struct{
        int64_t uncodedSize;
        int64_t codedSize;
} tBWTED;
```

Prototyp funkce pro kódování bude mít tvar:

```
/* bwted – záznam o kódování
inputFile – vstupní soubor (nekódovaný)
outputFile – výstupní soubor (kódovaný)
návratová hodnota – 0 kódování proběhlo v pořádku,-1 při kódování
nastala chyba */
int BWTEncoding(tBWTED *bwted, FILE *inputFile, FILE *outputFile);
```

Prototyp funkce pro dekódování bude mít tvar:

/ bwted – záznam o dekódování*
inputFile – vstupní soubor (kódovaný)
outputFile – výstupní soubor (dekódovaný)
návratová hodnota – 0 dekódování proběhlo v pořádku,-1 při dekódování
*nastala chyba */*
*int BWTDecoding(tBWTED *ahed, FILE *inputFile, FILE *outputFile);*

b) implementace aplikace:
Aplikace se bude jmenovat *bwted*, přičemž bude využívat implementovaných knihoven pro kompresi a dekompresi textových souborů s využitím BWT. Podporované parametry příkazového řádku jsou následující:

-i <ifile> název vstupního souboru <ifile>. Pokud parametr nebude zadán, bude se za vstupní soubor považovat standardní vstup (stdin).
-o <ofile> název výstupního souboru <ofile>. Pokud parametr nebude zadán, bude se za vstupní soubor považovat standardní výstup (stdout).
-l <logfile> název souboru výstupní zprávy <logfile>. Pokud parametr nebude zadán, tak výpis zprávy bude ignorován.
-c aplikace bude vstupní soubor kódovat.
-x aplikace bude vstupní soubor dekódovat..
-h vypíše nápovědu na standardní výstup a ukončí se.

Výstupní zpráva bude obsahovat login, původní velikost vstupního souboru (v bytech) a novou velikost komprimovaného souboru (v bytech), a to ve formátu dle přiloženého příkladu. Příklad zprávy aplikace autora Pepi Zdepa vypadá následovně:

```
login = xzdepa97
uncodedSize = 16384
codedSize = 14937
```

c) komentáře:
Zdrojový kód komentujte tak, aby nebylo potřeba dodatečně projekt obhajovat. Komentujte zejména parametry a činnost funkcí, datové typy, lokalní proměnné (pokud to nebude z názvu přímo vyplývat).
Všechny zdrojové soubory budou obsahovat hlavičku, ve které bude jméno, příjmení a login autora. Dále bude v hlavičce datum vytvoření, název a stručný popis souboru.

d) kompilátor a prostředí:
Knihovnu a výslednou musí být možné přeložit na serveru merlin, a to ve zde dostupné aktuální verzi kompilátoru *GCC*. Průběh překladu aplikace a jejího sestavování proběhne v prostředí *GNU Make*. K tomuto účelu bude sloužit soubor *Makefile*, k jehož interpretaci dojde zadáním příkazu *make* bez parametrů. Testování funkčnosti aplikace bude probíhat zásadně na serveru merlin.

e) dokumentace:
Pro knihovnu a aplikaci napište stručnou dokumentaci. Součástí dokumentace bude stručný popis algoritmu. Rozsah dokumentace by neměl přesáhnout max. 4 strany formátu. Formát dokumentace bude PDF.

f) odevzdává se:
Archív formátu ZIP, který se bude jmenovat *kko.proj2.”login”.zip* (např. pro Pepu Zdepa kko.proj2.xzdepa97.zip), bude obsahovat adresář *kko.proj2.”login”* (např. kko.proj1.xzdepa97) s následujícím obsahem:

- *bwted.h* – rozhraní knihovny
- *bwted.c* – implementace knihovny
- *main.c* – aplikace
- *Makefile* – definice způsobu kompilace programem make
- *bwted.pdf* – dokumentace

Pokud to bude z pohledu řešení účelné (lepší strukturovaní zdrojových kódů, modulárnost aplikace), je samozřejmě možné vytvořit i další soubory (různé pomocné knihovny, hlavičkové soubory, atd) kromě těchto požadovaných. Nezapoměňte je však umístit do odevzdávaného archívu.

Tipy a rady na závěr:

- Důležitou vlastností je také přenostitelnost. Pro různé platformy může být velikost typů int, long různá. Těmto problémům se vyhneme použitím předdefinovaných typů *int8_t, int16_t, int32_t, u_int_8_t* apod., které jsou definovány v knihovně sys/types.h.
- Pro ladění potíží s pamětí (Segmentation fault apod.) doporučuji použít nástroj *valgrind*, který se používá na aplikaci zkompilovanou pomocí *make debug* (příp. *gcc -ggdb3*). Doporučují použít i když na první pohled je vše v pořádku, chybná práce s pamětí se může projevit výpadkem paměti až třeba u jiné platformy.
- V licenci GPL existuje také pro účely ladění pohodlný debugger *DDD* (DataDisplayDebugger), který je grafickou nadstavbou nad *gdb*. Opět je vhodné kompilovat kód pomocí *make debug*.
- Pro zpracování parametru příkazové řádky je výhodné využít funkce *getopt* (resp. getopt_long), jejíž rozhraní je definováno v unistd.h (resp. getopt.h).
- Pro implementaci některých datových struktur a operací s nimi může být výhodné použít v aplikaci knihovnu STL (C++ Standard Template Library).

Doporučená literatura:

[1] Salomon, D.: "Data Compression, The Complete Reference," NY, USA, Springer, 2000, ISBN-0-387-95045-1

[2] Učební texty do předmětu Kódování a komprese dat, Brno, FIT VUT, 2006

[3] Sayood, K.: "Losless Compression Handbook", San Diego, CA, USA, Elsevier Press 2003, ISBN 0-12-620861-1

[4] Sayood, K: "Introduction to Data Compression", San Francisco, CA, USA, Elsevier Press 2006, ISBN -0-12-620862-X

[5] "Burrows-Wheeler Transformation". Odkaz na URL: http://www.data-compression.info/Algorithms/BWT/

[6] "Move To Front". Odkaz na URL: http://www.data-compression.info/Algorithms/MTF/index.htm