Projekt 1 - Práce s textem

Popis projektu

Cílem projektu je vytvořit program, který buď binární data formátuje do textové podoby nebo textovou podobu dat převádí do binární podoby. V případě převodu binárních dat na text bude výstupní formát obsahovat adresy vstupních bajtů, hexadecimální kódování a textovou reprezentaci obsahu. V případě převodu textu do binární podoby je na vstupu očekáváné hexadecimální kódování bajtů.

Detailní specifikace

Program implementujte ve zdrojovém souboru *proj1.c*. Vstupní binární data budou čtena ze standardního vstupu (stdin), výstup bude tisknut na standardní výstup (stdout). Chování programu lze upřesnit jeho argumenty (viz spuštění programu).

Překlad a odevzdání zdrojového souboru

Odevzdání: Odevzdejte zdrojový soubor *proj1.c* prostřednictvím informačního systému.

Překlad: Program překládejte s následujícími argumenty

```
$ gcc -std=c99 -Wall -Wextra -Werror proj1.c -o proj1
```

Syntax spuštění

Program se spouští v následující podobě: (./proj1 značí umístění a název programu):

```
./proj1 [-s M] [-n N]
```

nebo

```
./proj1 -x
```

nebo

```
./proj1 -S N
```

nebo

```
./proj1 -r
```

Pokud je program spuštěn s libovolnými dalšími neprázdnými argumenty, vypíše svůj krátký popis (nápovědu) a úspěšně skončí. V opačném případě provádí čtení a zpracování dat ze vstupu.

Implementační detaily

Převod binárního vstupu na text

Pokud je program spuštěn bez argumentů, s argumentem -x (hexa-print) nebo s argumentem -S (strings), převádí vstupní binární data do textové podoby. Argument programu pak definuje výstupní formát:

Spuštění bez povinných argumentů

Výstupní formát se skládá z posloupnosti řádků, kde každý řádek popisuje jednu sérii 16 bajtů ze vstupního souboru. Každý řádek odpovídá formátu:

```
AAAAAAAA  xx xx xx xx xx xx xx xx  xx xx xx xx xx xx xx xx  |bbbbbbbbbbbbbbbb|
```

kde:

- *AAAAAAAA* je adresa/pozice prvního bajtu dané série ve vstupním souboru. Jedná se o hexadecimální číslo z cifer 0-9a-f zarovnané na 8 číslic, doplněné nulami zleva. Adresa prvního bajtu je 00000000.
- *xx* vyjadřuje hexadecimální hodnotu daného bajtu. Pozice *xx* na řádku odpovídá pozici bajtu v dané sérii.
- *b* je tisknutelná podoba daného bajtu. V případě, že daný znak není tisknutelný, vytiskne se znak . (tečka). Nechť tisknutelný znak je ten, který je v ASCII a je definován pomocí funkce isprint.

V případě, že série obsahuje méně než 16 bajtů, namísto chybějících cifer *xx* a odpovídajících *b* se vytiskne výplň pomocí mezer.

Tato varianta spuštění může mít upřesňující argumenty -s M a/nebo -n N.

- Přepínač -s (skip) definuje, na které adrese má výpis začínat (tedy kolik znaků ze vstupního souboru se má ignorovat). Argument M je nezáporné číslo. V případě, že M je větší než velikost vstupního souboru, nevypíše program nic.
- Přepínač -n (number-of-chars) definuje maximální délku vstupních bajtů ke zpracování. N je kladné číslo.

Spuštění s argumentem -x

Veškerá vstupní data budou převedena do hexadecimální podoby na jeden řádek. Každému vstupnímu bajtu odpovídá dvouciferné hexadecimální číslo z číslic 0-9a-f.

Spuštění s argumentem -S

Program bude tisknout pouze takové posloupnosti v binárním vstupu, které vypadají jako textový řetězec. Každý řetězec je vytištěn na jeden řádek. Nechť řetězec je nejdelší posloupnost tisknutelných a prázdných znaků (tj. mezera nebo tabulátor, viz isblank), jejíž délka je větší nebo rovna N znaků. N je druhý argument programu a udává celé číslo v intervalu 0 < N < 200.

Převod textového vstupu na binární

Pokud je program spuštěn s argumentem -r (reverse), očekává na vstupu sekvenci hexadecimálních číslic a tyto převádí do binárního formátu. Bílé znaky na vstupu program ignoruje. Každá dvojice vyjadřuje hodnotu jednoho bajtu (první číslice má větší váhu). V případě, že je počet číslic lichý, poslední čislice vyjadřuje hodnotu bajtu v rozsahu 0-15.

Omezení v projektu

Je zakázané použít následující funkce:

- všechna volání hlavičkového souboru string.h - hlavičkový soubor je v projektu zakázaný,
- volání z rodiny malloc a free - práce s dynamickou pamětí není v tomto projektu zapotřebí,
- volání z rodiny fopen, fclose, fscanf, ... - práce se soubory (dočasnými) není v tomto projektu žádoucí,
- volání scanf a jeho varianty - cílem projektu je převody naprogramovat, nikoliv je používat.
- volání atoi - neověřuje správnost vstupních dat.

Neočekávané chování

Na chyby za běhu programu reagujte obvyklým způsobem: Na neočekávaná vstupní data, formát vstupních dat nebo chyby při volání funkcí reagujte přerušením programu se stručným a výstižným chybovým hlášením na příslušný výstup a odpovídajícím návratovým kódem. Hlášení budou v kódování ASCII česky nebo anglicky.

Příklady vstupů a výstupů

```
$ echo "Hello, world! Ahoj svete!" | ./proj1
00000000  48 65 6c 6c 6f 2c 20 77  6f 72 6c 64 21 20 41 68  |Hello, world! Ah|
00000010  6f 6a 20 73 76 65 74 65  21 0a                    |oj svete!.      |

$ echo "Hello, world! Ahoj svete!" | ./proj1 -s 14 -n 5
0000000e  41 68 6f 6a 20                                    |Ahoj            |

$ echo "Hello" | ./proj1 -x
48656c6c6f0a

$ printf 'Hello, world!\0Ahoj svete!\n\0AP\nABCD\n' | ./proj1 -S 3
Hello, world!
Ahoj svete!
ABCD

$ echo "48 65 6c6c6f a" | ./proj1 -r
Hello
```

Hodnocení

Na výsledném hodnocení mají hlavní vliv následující faktory:

- přeložitelnost zdrojového souboru,
- formát zdrojového souboru (členění, zarovnání, komentáře, vhodně zvolené identifikátory),
- dekompozice problému na podproblémy (vhodné funkce, vhodná délka funkcí a parametry funkcí),
- správná volba datových typů, případně tvorba nových typů,
- správná funkcionalita převodu dat a
- ošetření chybových stavů.

Poznámky

- Textový soubor je takový, který je buď prázdný nebo obsahuje tisknutelné a bílé znaky a jeho poslední znak je znak konce řádku.
- Při vypracování se můžete inspirovat nástroji hexdump(1) s parametrem -C a strings(1).