IZP - Projekt č. 4 - České řazení

Než začnete programovat, přečtěte si pozorně celé zadání.
Poslední změna: 23. září 2010

Motivace

Cílem této úlohy je zopakovat si nebo se naučit

- studovat a chápat netriviální reálné problémy,
- řazení textů s českými specifiky,
- práci s dynamicky alokovanou pamětí,
- načítání a zápis strukturovaných dat ze/do souborů,
- ošetřování problémů, které mohou nastat při práci se soubory
- vše, co bylo potřeba k předchozím úlohám.

Zadání úlohy

Vytvořte program zpracovávající dotazy týkající se výběru buněk zadané tabulky. Tabulka obsahuje textové buňky, text bude obsahovat českou diakritiku. Dotazy se skládají ze tří částí: (i) výběr řádků, které splňují zadané kritérium, (ii) výběr konkrétního sloupce vybraných řádků a (iii) případné seřazení vybraných hodnot.

Kritéria pro výběr řádků mohou být následující (pro jeden běh programu bude vždy vybráno maximálně jedno kritérium):

- kritérium "je před" je splněno na řádku, ve kterém je hodnota zadaného sloupce abecedně před zadanou hodnotou
- kritérium "je za" je splněno na řádku, ve kterém je hodnota zadaného sloupce abecedně za zadanou hodnotou

Z vybraných řádků podle zadaného kritéria je zvolen sloupec, jehož hodnoty mají být vypsány do souboru (řešte možné vstupně-výstupní chyby). Pokud je požadováno, výstupní textové hodnoty jsou před zápisem do souboru seřazeny -- vyberte si algoritmus řazení a implementujte ho. Abecední porovnání podle zadaných kritérií a řazení textových hodnot musí být provedeno podle normy pro české řazení ČSN 97 6030 (výtah z uvedené normy). Funkce ze standardní knihovny, které řeší lokalizované porovnávání českých řetězců můžete použít pouze pro otestování správné implementace.

Formát vstupního souboru (tabulky)

Tabulka se skládá z hlavičky a záznamů (řádků a sloupců tabulky). Hlavička tabulky pojmenovává jednotlivé sloupce textovými řetězci. Tyto řetězce jsou potom používány pro identifikaci sloupce v použitém kritériu a pro identifikaci sloupce určeného pro tisk. Názvy sloupců jsou odděleny jednou nebo více mezerami. Počet ani pořadí sloupců tedy nejsou předem dány, ale program je zjistí po přečtení hlavičky souboru. Za hlavičkou tabulky následují řádky tabulky. Každý řádek se skládá z několika slov oddělených jednou nebo více mezerami. Každé slovo v řádku odpovídá danému sloupci definovanému v hlavičce tabulky.

Kvůli zjednodušení bude obsah souboru v kódování ISO-8859-2. Všechny sloupce považujte za textové. Chce-li uživatel mít případné číselné hodnoty seřazeny správně podle hodnoty, musí je vhodně doplnit nevýznamnými nulami tak, aby textové porovnání produkovalo správné pořadí (např. místo číselných hodnot 9 a 123 budou v souboru hodnoty 009 a 123).

Příklad: soubor tabulka.txt

```
name       surname      place      birth
Jan        Novák        Brno       2000-01-01
Anna       Novotná      Olomouc    1999-12-24
Franta     Odvedle      Praha      1989-11-17
```

Formát parametrů programu

```
$ ./proj4 [parametry] vstupni_soubor vystupni_soubor
```

Program rozumí následujícím parametrům. Můžete předpokládat, že parametry budou zadávány v naznačeném pořadí. Není nutné (ale ani zakázané) očekávat parametry v libovolném pořadí. Označení nepovinný/povinný znamená, že jej uživatel při spouštění programu může nebo musí použít. Implementovat je musíte všechny.

- **Kritérium výběru řádku** -- nepovinný parametr, maximálně lze použít jedno kritérium na jeden běh programu.

  `--before SLOUPEC ŘETĚZEC`
  : Kritérium "je před". Parametr SLOUPEC je textový řetězec identifikující pozici slova na řádku, parametr ŘETĚZEC je textový řetězec, se kterým bude slovo porovnáno.

  `--after SLOUPEC ŘETĚZEC`
  : Kritérium "je za". Parametr SLOUPEC je textový řetězec identifikující pozici slova na řádku, parametr ŘETĚZEC je textový řetězec, se kterým bude slovo porovnáno.

- **Identifikace sloupce pro tisk** -- povinný parametr.

  `--print SLOUPEC`
  : Výběr sloupce pro tisk. Program bude tisknout řetězce ze sloupce SLOUPEC z vybraných řádků (pokud bylo použito nějaké kritérium) nebo ze všech řádků.

- **Seřazení výstupu** -- nepovinný parametr.

  `--sort`
  : Seřadí hodnoty určené pro tisk. Řazení bude vzestupné. Bude se řadit abecedně podle pravidel pro české řazení.

Příklad: Po spuštění programu na uvedeném příkladu

```
$ ./proj4 --after surname Novák --print birth --sort tabulka.txt narozeni.txt
```

bude soubor narozeni.txt obsahovat:

```
1989-11-17
1999-12-24
```

Příklad: Pro otestování českého řazení můžete použít například tato data. Výsledek vašeho algoritmu pak srovnejte s výstupy programů OO Calc nebo MS Excel, oba umí řadit správně podle české normy. Formát souboru v podstatě odpovídá jedné z variant formátu CSV.

```
surname name place birth
Janů Alena Hodonín 1625-02-21
Januš Jan Aš 1589-03-26
Janu Alena Hrčava 0295-04-05
Janůšek Jan Karviná 0860-12-23
Chalupa Jan Brno 1073-02-17
Pinula Jan Jeseník 9210-09-09
Piňos Josef Brno 1530-11-22
Habrda Jiří Praha 1460-12-01
Dobrovolný Jan Kozomín 1460-07-27
Ďas Jan Poděbrady 1945-04-26
Řezáč Jan Orlová 1978-01-17
Rozhon Jan Stonava 1986-07-09
Janula Jan Albrechtice 1285-11-22
Janušová Alena Frýdek-Místek 1796-12-09
Janošíková Alena Bzenec 1386-04-29
Pištora Jiří Březejc 1780-04-05
Pišťáček Jan Dyjice 1038-07-08
Jánošík Jiří Terchová 2406-09-13
Piška Jan Příluky 1760-12-01
Janoščín Jan Schořov 1891-03-05
Píše Jan Štrampouch 1830-11-06
Jánoš Jan Janovice 2089-02-19
Pišanová Alena Kamýk 1383-02-20
Janoš Jan Močidlec 2305-10-06
Píša Jan Hleďsebe 0900-01-07
```

Maximální počet získaných bodů

Za tuto úlohu lze získat až **8 bodů**, přičemž celkové hodnocení zhruba odpovídá následujícímu rozložení: 50% kritérium výběru řádku (25% parametr `--before`, 25% parametr `--after`), 10% výběr sloupce pro tisk (parametr `--print`) a 40% seřazení výstupu (parametr `--sort`). Součástí řešení není dokumentace (ale je vhodné ji zpracovat, zvláště v případě, že se rozhodnete implementovat nějaké rozšíření - stačí textový soubor).

Prémie

Za tuto úlohu lze získat prémiový bod, pokud si zadání vhodným netriviálním způsobem rozšíříte. Například:

- podpora parametru `--usort` (unikátní řazení, tj. seřazený výstup neobsahuje dvě stejné položky)
- podpora výběru více nebo dalších netriviálních kritérií nad různými sloupci v jednom běhu programu.
- Při zadání parametru `--codepage win1250` bude program schopen pracovat ve zvolené znakové sadě (vstup i výstup). V tomto případě musí zvládat minimálně tato kódování: Windows 1250 a ISO-8859-2 (možná i CP852).
- Při zadání parametru `--utf-8`, bude program schopen pracovat se soubory v kódování UTF-8.
- Vzhledem k tomu, že na tento projekt je o něco méně času než na ostatní, mohou být prémií ohodnocena i ta řešení, která splní zadání beze zbytku.

Co se zejména hodnotí

Cílem úlohy je procvičit algoritmy pro řazení, dále pak získání zkušeností s implementací českého řazení a prací s textem v českém kódování. V této úloze se zejména hodnotí:

- Správná implementace českého řazení podle normy.
- Efektivita zvoleného řešení - zejména porovnávání (plus efektivita implementované varianty zvoleného řadícího algoritmu, např. ve smyslu řazení s/bez přesunu položek; pamatujte, že i bubble sort lze implementovat více či méně efektivně).
- Správná práce se soubory (ošetření chybových stavů při otvírání, zavírání, chybný formát vstupního souboru, ...)
- Správná práce s dynamickými datovými strukturami, budou-li použity.

- Správná práce s poli, budou-li použita.

Pomůcka

Pokud pracujete ve Windows a máte problémy s českým kódováním vstupních souborů v ISO-8859-2, můžete pro překódování použít jednu z funkcí souborového manažeru Salamander (je na disku Q:). Další programy pro překódování diakritiky najdete například na Slunečnici.

V Linuxu slouží pro překódování češtiny program `iconv`.

Než začnete programovat, promyslete si, jak bude vaše řešení vypadat. Není důvod vytvářet složitý program. Jednoduché elegantní řešení dá mnohem méně práce s lepšími výsledky, než rozsáhlý kód. Pokud si svým návrhem nejste jistí, přijďte se poradit s některým z vyučujících.

Tip: Postupujte při řešení tak, aby i nekompletní program byl natolik funkční, že to půjde obhájit a alespoň částečně otestovat. Funkčnost potom přidávejte podle toho, kolik vám zbude času. Například můžete postupovat takto: 1. zpracování výběrových kritérií podle anglické abecedy 2. základní řazení podle anglické abecedy, 3. nahrazení anglického porovnání českým, 4. důkladné ošetření všech výjimečných stavů ve vstupním souboru (nad vhodným pořadím se zkuste sami zamyslet).

Návod pro uživatele Linuxu s nastaveným kódováním UTF-8 (pro Ubuntu)

Programování této úlohy v Linuxu s nastaveným kódováním UTF-8 můžete být dost nepohodlné, protože UTF-8 je zcela odlišný způsob kódování znaků, než ISO-8859-2 vycházející z ASCII. V UTF-8 mají různé znaky různý počet bajtů! To může způsobovat problémy třeba při ladění, kdy debugger zobrazuje znaky v kódu ISO-8859-2 nečitelným způsobem. Rovněž se nepokoušejte zpracovávat vaším programem soubory v kódování UTF-8. Nefungovalo by to. Funkce getchar by každý český znak přečetla jako dva nesmyslné znaky!

Abyste mohli pracovat s kódováním ISO-8859-2, musíte si nainstalovat do systému patřičnou lokalizaci. Lokalizaci v konkrétním kódování je nutné ručně vygenerovat, například příkazem locale-gen (pro více informací se podívejte do manuálové stránky nebo hledejte jeho použití na internetu)

```
sudo locale-gen cs_CZ.ISO-8859-2
```

Máte-li vygenerovánu lokalizaci ve správném kódování, je nutné ji zapnout. Předpokládám, že toto kódování budete používat pouze pro práci na projektu. Nejpohodlnější způsob práce potom asi bude tento: Spusťte program Konsole v KDE nebo Gnome Terminál. Tyto terminály umí změnit kódování aktuálního sezení. V programu Konsole se to mění nastavením aktuálního profilu v záložce pro pokročilé nastavení (platí pro KDE4). V Gnome Terminálu je na to přímo položka v menu Terminal.

Kódování můžete také nastavit sami příkazem

```
export LANG=cs_CZ
```

případně příkazem

```
export LANG=cs_CZ.ISO-8859-2
```

Konkrétní varianta závisí na tom, jak jste si lokalizaci vygenerovali. Když nyní z tohoto sezení spustíte jakýkoli program, bude pracovat s tímto zvoleným kódováním. Z této konzole tedy můžete spouštět například Code::Blocks, editor, různé debuggery, a samozřejmě i váš program. Pokud dodržíte zásadu, že váš zdrojový kód (a zejména tabulka s váhou českých znaků) a vstupních soubor budou ve stejném kódování ISO-8859-2, pak by to měl fungovat bez problémů.

Chcete-li mít toto nastavení zapnuté permanentně, musíte je zapsat do startovacích skriptů modifikací souboru `/etc/default/locale`.

Autor: David Martinek. Poslední modifikace: 23. září 2010. Pokud v tomto dokumentu narazíte na chybu, dejte mi prosím vědět.