TDDE54/TDDD87/TDIU08/725G92/9AMA73: Teckenkodning ...

Torbjörn Jonsson <torbjorn.jonsson@liu.se>

Sat 25/09/2021 07:59

To: **TDDD87_2021HT_FP <tddd87_2021ht_fp@student.liu.se>; TDDE54_2021HT_AU <tdde54_2021ht_au@student.liu.se>; TDIU08_2021HT_C9 <tdiu08_2021ht_c9@student.liu.se>; 725G92_2021HT_Z3 <725g92_2021ht_z3@student.liu.se>**

Hejsan.

Ett mail specifikt om teckenkodning och de fel som är "små", men som vi märkt att det ställer till det för er. Tre olika problem finns som vi sett att ni har som är besvärliga att fixa på vår sida (ett stort problem i hela världen tyvärr).

Tips i slutet av mailet som gör att ni kan hjälpa er själva att få till det så att ni slipper fel med teckenkodningen i automaträttningen i möjligaste mån.

Om ni känner att ni inte vill ha reda på vad som är problemen så kan ni direkt hoppa till slutet och ta tipsen, men såklart rekommenderar jag er att skumma igenom detta då det inte är tungt att bära på kunskap. :-) Kanske något som är värt att veta när ni råkar ut för saker i framtiden.

Olika teckenuppsättningsstandarder

Vi kan börja med det som är "lättast att fixa på vår sida", men ändå inte trivialt. Vi pratar här om t.ex. ASCII, ISO-8859-1, Latin-1, UTF-8, UTF-16 och andra versioner av standarder för att bygga upp eller representera teckenuppsättningar i våra datorer.

En gammal variant som är enkel att förstå är den som också finns i boken som kallas ASCII (den bygger på att vi har 256 olika tecken att använda och varje tecken har sin plats. I denna finns även åäö och en del andra "bra att ha"-tecken som "é" m.fl. Denna är det som används generellt sett när man skriver "Character" i Ada och den ryms i "en byte"

Det finns i Ada även två andra varianter av datatyper för tecken för att klara av de nyare varianterna av kodning, men det är inte trivialt att ta upp hur vi kör med dessa så vi skippar detta i denna kurs. Det är dock bra att veta om att de finns då det finns saker som är speciellt som skapar lite strul med t.ex. indexering i fält. Se mer om det i det mail som kommer lite senare om Ada.P3 och vad som är bra att tänka på där.

Den vanligaste uppsättningen idag är UTF-8. Den bygger på att man kan ha olika mycket minne för ett tecken (upp till 4 byte) så att man kan representera en massa (en MASSA!) tecken. Denna variant är den som används i terminalen i ThinLinc och på våra Linux-datorer i salarna på LiU. Den används antagligen i princip på varenda dator som ni stöter på.

UTF-8 har inte tecknen "åäö" och andra sådana tecken på samma kod som t.ex. ASCII. Därav en del strul med det vi kallar teckenkodning även i automaträttningen. Vi omvandlar dock de filer som ni skickar in som vi kan detektera är andra teckenkodningar till UTF-8 så detta tar bort de allra flesta "fel" som har med detta att göra.

Trots detta blir det ibland fel iallafall. Vi går vidare till nästa del i dettta.

Olika versioner av samma teckenuppsättningsstandard

Här kommer det trista. UTF-8 om vi tar det som exempel finns i olika versioner. Det finns UTF-8 för "Unix", "DOS", ...

Dessa olika versioner har givetvis olika koder för att koda t.ex. "åäö". Detta ställer till det mer då det är svårare att upptäcka och det blir galet om man gör om en redan korrekt fil "en gång till" om man skulle vilja försöka chansa. Här kan man råka lite illa ut i automaträttningen om man alltså sitter på en Windows-dator eller kanske en Mac eller annat och sen kopierar över sin kod till ThinLinc för att skicka in saker. Lösning finns längre ner i mailet om ni har strul med detta.

Alltså: Att köra UTF-8 är inte samma sak på olika datorer i värsta fall.

Olika tecken inom samma standard (exempelvis inom samma UTF-8-variant)

Den värsta av dem alla då som slutkläm på "eländet". Den som gör att vi inte kan lösa detta problem på ett smidigt sätt utan ni behöver hjälpa er själva med de tips som kommer nedan. Vad är nu detta?

Jo, inom t.ex. UTF-8 finns det FLERA sätt att skriva in tecken som "ser lika ut" (eller ser nästan lika ut i vissa fall). D.v.s. det finns flera teckenkoder inom UTF-8 som ger likartade tecken. Vi tar tecknet "ä" som exempel då detta dykt upp som problem för några.

Vi har sett dessa två varianter i ren kod som är inom "UTF-8-Unix" (d.v.s. den på våra system, ThinLinc och våra datorer).

"ä" Vanligt inskrivet genom att trycka på "ä" på tangentbordet.
"ä" Inskrivet med kombinationen av "¨" och "a" (i Emacs på min dator).
"ä" Inskrivet med kombinationen av "¨" och "a" (i annat verktyg).

De kanske ser likadana ut för dig som det gör för mig (i mailet), men när jag ser dem i Emacs eller andra vektyg visar det sig att de inte är helt lika i alla stycken. Om jag kör följande program:

```
with Ada.Text_IO;            use Ada.Text_IO;
with Ada.Integer_Text_IO;    use Ada.Integer_Text_IO;

procedure Testing_Utf8 is

   Str1 : String(1 .. 2) := "ä";    -- Det normala "ä" som vi
använder.
   Str2 : String(1 .. 2) := "ä";    -- Kombination av "¨" och
"a" i Emacs.
   Str3 : String(1 .. 3) := "ä";    -- Kombination från annat
verktyg.
                                    -- OBS! 3 tecken (byte) i
den sista strängen!

begin
   Put(Str1 & "  ");
   Put(Character'Pos(Str1(1)));
```

```
      Put(Character'Pos(Str1(2)));
      New_Line;

      Put(Str2 & "  ");
      Put(Character'Pos(Str2(1)));
      Put(Character'Pos(Str2(2)));
      New_Line;

      Put(Str3 & "  ");
      Put(Character'Pos(Str3(1)));
      Put(Character'Pos(Str3(2)));
      Put(Character'Pos(Str3(3)));
      New_Line;
   end Testing_Utf8;
```

På de tecknen ovan visar det sig att jag får ut följande teckenkodningar vid programkörningen:

```
ä          195        164
ä          195        164
ä           97        204        136
```

Jag hade "tur" när jag råkade få samma teckenkodning på de två första verkar det som. D.v.s. vi sitter i en verklighet idag som ställer till det ordentligt för alla som programmerar (och alla andra också i slutänden). Detta är tyvärr något som vi inte kan göra något åt precis nu utan för att ni skall kunna passera t.ex. nålsögat med automaträttningen (som inte kommer att släppa igenom fel av typen "du har skrivit saker felaktigt") har vi därför gjort en liten tabell nedan med de tecken som borde fungera och klara av att passera genom våra filter (givet att ni skrivit era program i UTF-8 så att ni inte blandar olika standarder vilket kan ställa till det ännu mer :-( ).

Teckenkodningen i automaträttningen

För att ni skall komma framåt lite lättare nu så har jag gjort en liten tabell med den UTF-8 som vi använder i vår automaträttning har följande tecken som kan vara bra i denna kurs. Vi kommer att hålla oss ifrån tecken som ställer till det i möjligaste mån, men vi vill verkligen inte ta bort att man skall kunna skriva saker på t.ex. svenska i utskrifter etc.

Om ni får strul med dessa "småsaker" som teckenkodningen ställer till så rekommenderar jag att ni ersätter de tecken som strular med de som finns i denna tabell. Ni märker det snabbt genom att "vår differens" skriver ut "UTF-8-specifika" tecken som lite kryptiska i de svarsmail ni får. Även om detta inte är trevligt är det iallafall en nytta i detta läge att vi kan rädda er lite grann.

Här kommer de tecken som jag direkt ser som "möjligen rimliga" att vi kommer att kunna ha med i våra tester framöver (finns säkert andra tecken som vore roliga, men vi skippar dem så länge). Dessa är inskrivna i Emacs med den teckenkodning som vi brukar använda så de borde rimligen fungera ihop med de program vi använder som "facit" (eller "orakel" som vissa säger).

```
-- åÅ
-- äÄ öÖ ëË ḧḦ ïÏ ẗ üÜ ẅẄ ẍẌ ÿŸ
-- øØ æÆ
-- áÁ ćĆ éÉ ǵǴ íÍ j́J ḱḰ ĺĹ ḿḾ ńŃ óÓ ṕṔ ŕŔ śŚ úÚ ẃẂ ýÝ źŹ
-- ő̋Ő ű̋Ű
```

-- àÀ èÈ ìÌ ùÙ ẁẀ ỳỲ
-- ãÃ ĩĨ ñÑ õÕ ũŨ ṽṼ ỹỸ
-- ~"'`¦¡$¤!#%&/()=?{}[],.;:-_*^\§
-- °¹²³½¼¾⅝
-- ¢£€¥«»©®‘’“”¡¿
-- ←↓→↑
-- <>±÷×·
-- œþßðđŋħłnµþΩŁÞŊĦÆŒĐ

Om ni får teckenkodningsfel som ni ser skiljer sig från det som "facitprogrammet" ger. Testa att byta ut mot ovanstående så kan det rulla igenom. Om inte: Kontakta mig så får jag se på det.

Om ni har inskickade saker som ligger just nu med dessa problem. Byt ut mot ovanstående och skicka in igen så får vi se om det inte löser sig.

Bästa tipset är nog att sitta via ThinLinc för att skriva programmen ni skall skicka in så att ni inte råkar blanda olika varianter och få en massa strul.

Avslutning

Hoppas att detta mail har givit er en inblick i något som vi alla ställs för i olika situationer idag. Tänkte att det (även om det inte direkt ingår i kursen) kan var av intresse för er att vet lite mer om vad som ställer till det för alla våra "kära" programmerare och er också i olika verktyg.

Hoppas att det nu rullar på lite mer och att ni läser de tips som kommer ut om respektive uppgifter lite allteftersom. Hoppas verkligen att detta kan hjälpa er framåt. Vi vill inget annat än att det skall gå bra för er.

Vet inte hur jag skall peppa er på bästa sätt, men jag skulle vilja sitta på era axlar och ge er tips om hur man gör. Tappa inte modet nu och ge inte upp. Det är på gång nu.

M.v.h.
/TJ
--

- -
_/_/_/_/_/_/_/_/_/ ***Torbjörn Jonsson***
_/ _/ ***013-28 24 67***
_/ _/ ***Torbjorn.Jonsson@LiU.SE***
_/ _/ _/ *IDA/SaS/**UPP***
_/ _/_/ ***Institutionen för Datavetenskap***
- - - - - - - - - - ***Linköpings universitet***