

# Re: Unicode Support

---

*Source:* <http://coding.derkeiler.com/Archive/Assembler/alt.lang.asm/2005-04/msg00794.html>

---

- *From:* [websnarf@xxxxxxxxxx](mailto:websnarf@xxxxxxxxxx)
  - *Date:* 20 Apr 2005 15:00:23 -0700
- 

Beth wrote:

> Chewy wrote:  
>> Hi Beth,  
>> Some of the other issues, such as non-english labels and  
>> numbers, can be a pain in the arse. Numbers not so much (AFIAK most  
  
>> of the world recognises the 0..9 numerical set),  
>  
> Well, sort of...Tamil has characters for "ten", "one hundred" and  
> "one thousand", as well as nine digits (no zero!! They've not been  
> watching that Stargate episode! ;)...but, yeah, numbers aren't so  
> bad...

There are many different numbers in different alphabets.

>> but labels. The biggest issue for me, would be combinations or  
>> equivalence. How should those be handled?  
>  
> There are "rules" for "normalising" strings in the UNICODE book...in  
> fact, some UTF-8 sequences are actually deliberately "invalid" so that  
  
> only one way to access a character is allowed and "overlong forms"  
> (such as using an "escape sequence" but accessing an ordinary ASCII  
> character) are considered "invalid"...

These are just code point representation rules. Beyond that, there are many illegal, and unassigned code points. Also there are nonsensical code point expressions like just a singleton "grave" accent (no language accepts that alone as a character in their alphabet).

> Follow these "rules" to put the string into a "normalised" form (that  
  
> is, there is only one valid way to address any particular character  
> :) and then there should be no problems with things like testing for  
> equivalence between two strings...

Uhh ... normalized means something different. Unicode normalization is not the only way a Unicode string should exist, its just a theoretical form from which you can compare strings. There are two main endorsed

## Re: Unicode Support

normalization forms — compatible and canonical. They specify a mechanism for rearranging and mapping of code point sequences to a more unified version.

For example, there are letters which can take multiple accents. And you can often specify them as base-char + accent1 + accent2. The question is, if you flip the order of the accents, does that represent a different character or not? The unicode normalization algorithms say no for some cases, and yes in others.

>> Well, it appears that the answer (for LuxAsm) is don't. Just  
>> have labels/identifiers in the normal 7bit ASCII range as already  
>> defined by most other assemblers/HLLs.  
>  
> Basically, yes...  
>  
> Though, in principle, I don't have a problem with the idea of also  
> applying it to the labels / identifiers too...but, in practice, a  
> lot of effort for what exactly?

A lot of effort for cornering every market outside of english. If programmers from non-english speaking countries could code in their own native tongue, it would probably be very attractive to them.

> In fact, my little "test" which demonstrates that NASM \_already\_  
> deals with UTF-8 comments and strings, proves the point a  
> different way...some tools \_already\_ are "accidentally" supporting  
> UTF-8 to that "basic level", without even knowing it...

Well that's kind of the point of UTF-8. For certain semantics, you get it for free, simply by allowing characters in the range 0x80 – 0xF8 in your tools. But of course, comments, and embedded strings in this case are easy because, comments don't need any checking, and assembly strings are essentially byte sequences which just correspond to raw data (legality or not is not something that needs to be imposed or checked).

> [...] and the fact that no-one has actually realised  
> this (I didn't either until I thought it would be interesting to  
> see what NASM would actually do ;), shows how great the "demand"  
> is...if people were regularly wanting UTF-8 source files passed  
> through NASM, then your post should have had Frank shouting  
> "NASM already does it!!"...

Well hang on. If you can't make identifiers rendered in UTF-8 (or possibly UTF-16) with proper normalization, then I wouldn't call that real unicode support.

--

Paul Hsieh

<http://www.pobox.com/~qed/>

Re: Unicode Support

<http://bstring.sf.net/>

---

- *Follow-Ups:*

- ◆ **Re: Unicode Support**
  - ◇ *From:* Evenbit
- ◆ **Re: Unicode Support**
  - ◇ *From:* Chewy509

- *References:*

- ◆ **Unicode Support**
  - ◇ *From:* Chewy509
- ◆ **Re: Unicode Support**
  - ◇ *From:* Chewy509
- ◆ **Re: Unicode Support**
  - ◇ *From:* Beth

- Prev by Date: **Re: HAY BETOV, when will you fix the RosAsm data structures ?**
- Next by Date: **Re: RosAsm is a broken pile of crap**
- Previous by thread: **Re: Unicode Support**
- Next by thread: **Re: Unicode Support**
- Index(es):
  - ◆ **Date**
  - ◆ **Thread**