

Re: Unicode Support

Source: <http://coding.derkeiler.com/Archive/Assembler/alt.lang.asm/2005-04/msg00984.html>

- *From:* "Beth" <BethStone21@xxxxxxxxxxxxxxxxxxxxxxxxxxxx>
 - *Date:* Fri, 22 Apr 2005 14:38:24 GMT
-

wolfgang kern wrote:

> Hi Beth,

Hi wolfgang :)

> [..]

> | As for taking up more memory and disk space: UTF-8 does not take up a

> | single bit extra from ASCII for any ordinary ASCII characters...

>

> ??

> I may have missed the point of UTF-8, I'm not through the whole book yet,

> how can I display 'mathematical operators' 2200..22FF (page 193) with UTF8

> ??

>

> At the moment I use ASCII 20..7F as they are and 01..1F and 80..FF

> as 'my personal character-set', while 00 is used to mark the end

> followed by one ore more 'format/function/more..'-'bytes.

>

> ie:

> 40 41 42 43

> 00 0a xx yz nn nn nn mm ; print/edit a numeric variable

> ; (type,format,action,group,address)

> will print: "@ABC-33.35 x10³-6⁶"

>

> 00 ... ; all var-types, input, locate, colours, ... just all needs ;)

> 00 E0 xx.. ; conditional call substring incl. format header (menus...)

> 00 FE ; end of a sub-string (menu/buttons/..)

> 00 FF ; end of the print job

Right, UTF-8 is actually fairly easy to understand, so I might as well describe it here...

I know you prefer concise tables, so let's see if I can make a simple table for it:

U-00000000 to U-0000007F: 0xxxxxxx

U-00000080 to U-000007FF: 110xxxxx 10xxxxxx

U-00000800 to U-0000FFFF: 1110xxxx 10xxxxxx 10xxxxxx

Re: Unicode Support

U-00010000 to U-001FFFFF: 11110xxx 10xxxxxx 10xxxxxx 10xxxxxx

*U-00200000 to U-03FFFFFF: 111110xx 10xxxxxx 10xxxxxx 10xxxxxx 10xxxxxx

*U-04000000 to U-7FFFFFFF: 1111110x 10xxxxxx 10xxxxxx 10xxxxxx 10xxxxxx
10xxxxxx

[* Technically, it is intended that no UNICODE character will ever go beyond FFFFh (1024K characters)...so, though these encodings are defined for UTF-8 (up to 2³¹), you shouldn't ever actually see these in any UTF-8 file...]

Examples:

So, for U-0041 (which is plain old ASCII "A" :), the byte is exactly the same as it would be in ASCII:

01000001 (or 41h)

For U-03A0 (which is Greek letter "pi" :), the sequence would be:

11001110 10100000 (or CEh A0h)

For U-2200 (which is the mathematical "for all" character: That is, an upside-down capital A :), the sequence would be:

11100010 10001000 10000000 (or E2h 88h 80h)

....so forth...I think you can work out the rest yourself...pick your UNICODE character then check what "range" it's in with the table above...place its binary digits into the spaces marked with the "x"...

Points to note:

1. 7-bit ASCII characters are encoded in exactly the same way in UTF-8
2. Byte-based (no "endianness" worries; Does not require any "BOM" ("byte order mark") character at the start of the file or anything like that, as Microsoft advise should be in their plain text file UTF-16 encoding...advice, in fact, that they were overstepping the mark to give because UNICODE themselves do not specify this as anything but "optional"...but, well, that's Microsoft for you, eh? ;)...)
3. All non-ASCII characters use a multi-byte sequence
4. Each byte in that multi-byte sequence has the highest bit set (so, it's clear what is ASCII and what is non-ASCII and they shouldn't become confused)...
5. The first byte of the sequence has as many highest bits set as there are bytes in the entire sequence (e.g. "110xxxxx 10xxxxxx" starts with two set bits in the first byte, so the sequence is two bytes long :)...)

Re: Unicode Support

6. After the first byte, all further bytes are of the form "10xxxxxx" (providing another extra 6 bits of "address range" per byte :)...)
7. The bytes 0xFE and 0xFF are never used in UTF-8 at all...
8. The first byte of a non-ASCII character is in the range C0h to FDh, subsequent bytes in a multi-byte sequence are in the range 80h to BFh, ASCII characters are in the range 00h to 7Fh...you can use this for easy resynchronisation (if you start reading in the middle of a multi-byte sequence, you can know that this is the case by what range the byte is in :)...)
9. All UNICODE characters are available (ASCII bytes are still just one byte long, all 16-bit "BMP" characters one to three bytes long, all defined UNICODE characters four bytes...being "variable-length" then size of files dependent on what characters encoded: If all ASCII, no different from ordinary ASCII file...if all "upper range" Chinese ideographs, then UTF-8 encoding is 4 bytes long (though, note that UTF-16 – with 16-bit per character, as Windows uses – is also 4 bytes long, so UTF-8, at worst, can only be as big as UTF-16 but will typically probably be smaller))...

NOTE: You might notice that it is possible to create what are known as "overlong forms" for characters...for example, you could unnecessarily take two bytes to encode an ASCII character:

11000001 10000001 (or C1h 81h)

Instead of the simpler:

01000001 (or 41h :)

These "overlong forms" are INVALID UTF-8...they should be rejected as errors, if encountered...note that these "overlong forms", as the name suggest, are unnecessarily long, so rejecting these also ensures the shortest possible encoding of the character, as well as "normalising" everything so that "comparisons" are easy (i.e. there is only one valid way to encode any particular character :)...)

In order to help detect "overlong forms" and reject them, here's another simple table...if any of these bit patterns are detected, you have an invalid "overlong form" (these are all invalid sequences in UTF-8):

1100000x (10xxxxxx)
11100000 100xxxxx (10xxxxxx)
11110000 1000xxxx (10xxxxxx 10xxxxxx)
11111000 10000xxx (10xxxxxx 10xxxxxx 10xxxxxx)
11111100 100000xx (10xxxxxx 10xxxxxx 10xxxxxx 10xxxxxx)

[Note that you can work out if it's "overlong" from the first or second bytes, irrespective of what the remaining bytes are (which I've placed in

Re: Unicode Support

brackets)...]

As you can see, it's quite a nice "encoding" for UNICODE because it leaves ASCII alone...and can access any defined UNICODE character in up to four bytes (which is no worse in size than UTF-16 for the "worst case" (a file of only "upper range" Chinese ideographs and nothing else...which is probably not any file you or I would ever want to create, eh? ;) but will generally be smaller in most cases...right down to, as noted, being exactly the same size as ASCII, if the file is only 7-bit ASCII characters)...for easy parsing, all non-ASCII multi-byte sequences have their highest bit set for every byte in that sequence...and the first byte of that sequence is in a different "range" to the subsequent bytes, with the number of "high bits" set being equal to the number of bytes in the entire sequence...

> [..support..]
> | Yeah, this is actually a "reverse" point on what's actually true...the
> | whole purpose of UNICODE is that it gets rid of all the things like
"code
> | page incompatibilities" once and for all...one standard character set
for
> | everyone...
>
> Even it will be a very huge pack of patterns for everyony to carry,
> it may make sense to set up an international agreement of tool-makers
> to a limited standard use of UTF character-sets.

Well, yes, but this is confusing the purpose of UNICODE...it's a "standard" for assigning unique numbers to all of these characters...that's what UNICODE is...just a "standard" for all the characters...

If you only want to support a limited set of characters then do so...if you want to STORE your strings in some "private format" and not one of the UTF "encodings" then do so...

The point of UNICODE is just that when you need to send files and data internationally, then the UNICODE standard ensures that there's ONE standard for all characters...so that you never get the "code page" problem, where a character you can see, can't be seen properly by someone else because their "code page" is different to yours...if everyone sends their files in UNICODE format then, you know, each character has a "unique" character code which cannot be confused with any other character...

If you like, UNICODE is about TRANSMITTING data internationally...a "standard" that just makes completely sure that what people see on the other end is exactly what you sent to them, whatever "code page" someone has set...

As for how many characters are in UNICODE? Well, it's Babylon that's to blame for that, yes? They are just trying to put all known characters into one standard...the fact that there are so many characters out there is to do with the human race and it's "Babylon confusion"...

Re: Unicode Support

Re: Unicode Support

Indeed, it should be noted _HOW_ UNICODE came up with the characters:

The majority of UNICODE characters were introduced by _MERGING_ all the other character set standards and "code pages" and such together...so, for instance, many of the "dingbats" come from standard _PostScript_ fonts...the "box drawing characters", of course, come from IBM's character set in the BIOS...and a large part of the UNICODE characters comes from _MERGING_ previous standards together...

Hence, the amount of characters is a consequence of there being too many "standards" out there...but note the reason _WHY_ UNICODE "merged" all these standards: In order to provide what they call "round-trip compatibility"...that UNICODE is a "superset" of all other standards...so that if you converted from some other standard to UNICODE and then back to that standard, there would be _ZERO_ data loss...no "translation errors"...

Indeed, much of the "problems" that exist in UNICODE relate to that...for instance, there is a specific "capital A with umlaut" character..._AND_ it's possible to have "capital A" character with a "combining umlaut" character after it (which puts the umlaut on top of the previous capital A character)...why both forms? Because, for compatibility with other standards (where the specific character "capital A with umlaut" existed), they had to retain the specific character, as well as their new "combining" characters (which are much more "versatile" because, you know, with a separate "combining umlaut" character then you could place an umlaut over _ANY_ character: A "snowman" character followed by a "combining umlaut" becomes "snowman with umlaut" on the screen ;)...)

It is, in fact, exactly because they were trying (too damn hard, if you ask me) to be "backwards compatible" with all these other, older standards that the "exceptions" here and there appear...

But, going back to the point about it being about "transmission", now you might see why UNICODE went to the effort of all this "backwards compatibility"...the _PURPOSE_ of UNICODE is a "standard" that "merges" all these standards...so that if I have data in one "code page" and you have data in a different "code page" and Rene has some "double byte character set" oriental file he wants to send...well, if we all convert to UNICODE, then _every_ single character in all the other standards is given a _UNIQUE_ character code...so, UNICODE prevents any and all "conflicts"...it _guarantees_ that the characters that appear when we all exchange files with each other – no matter what "code page" or other "standard" we're using – will be exactly the same characters seen that were sent...

That was the original and intended purpose of UNICODE...

But OSes and applications started deciding that they would just exclusively use UNICODE everywhere...as, in fact, this is a far easier implementation than trying to support each of the many "code pages" and "double byte character set" stuff that's out there...

Re: Unicode Support

Indeed, if you think UNICODE is problematic, then try implementing "Double byte character set" and a few thousand "code pages" for your OS: That'll be far, far harder to implement and, for MORE effort, you'd get less functionality...because where UNICODE can show the oriental ideographs AND Hebrew AND Cyrillic AND Latin all at the same time, the "code page" system does not allow that because only one "code page" can be active at a time...

Also, there are some "special characters" that require "processing", yes...but the vast majority of characters are just characters...you just store them in strings...so, other than that the storage is bigger, what's the difference if it's 95 characters, 950 characters or 9500 characters? The only people who are going to have problems with this are the font designers, drawing them all...but then, it is allowable to create fonts that only covers certain "ranges"...and programmers can just use fonts from the professional font designers...

Initially, granted, you might be "awe struck" with the amount of characters...but, really, you don't have to become "personally acquainted" with every Thai and Tamil character or anything like that...note: There's many scripts and languages supported from all over the world...the person who holds the world record for "most languages known" only knew 58 languages...NO-ONE comprehends it all...you're not expected to start learning how to speak "Runic" (not least because it's a "historic language" and NO-ONE out there speaks it anymore)...just store the characters in the strings...take note of the "special characters" (and all the first chapters of the UNICODE book explain all the "processing rules" for that)...but, otherwise, just store the characters in the strings...if you're only interested in reading ASCII then skip all the non-ASCII characters...

Note: I tested out that "Japanese Hello World" program as a UTF-8 file – the strings and comments in the file were in Japanese...but, obviously, following NASM's "rules" for identifiers and using the Intel mnemonics, everything else was standard ASCII – and sent it through NASM...

NASM is NOT designed to process UTF-8 files but it WORKED, anyway...this is because NASM just looks for the ASCII characters in the parsing...and the non-ASCII characters just "passed through" (for the strings) or were "ignored" (for the comments)...

So, you see, to an extent, NASM ALREADY supports UTF-8 files...and this was completely BY ACCIDENT...hence, the "worry" that it's complicated to implement? NASM implemented it without even trying!!! It can handle UTF-8 files and it wasn't even supposed to do that!!! So easy to implement, NASM has done so COMPLETELY BY ACCIDENT...

> | But this is the biggest problem, true...but it's a problem of "my software
> | is not up-to-date with UNICODE" ...yes, but isn't the point Chewy's making

Re: Unicode Support

Re: Unicode Support

> | about: "let's all update things to use UNICODE from now on"?
>
> At least, it's worth to think about it.

Well, the basic idea is simple: _IF_ everyone did move to UNICODE, then this should "solve" all these problems, once and for all (mind you, whenever people say that, it's normally "famous last words"...yeah, and the Titanic was "the unsinkable ship" too...but, you know, this is the "theory" behind it and even if there are problems, what's there already tends to be better than trying to support a few thousand "code pages" and "double byte character set" things and all the other stuff ;)...

Indeed, "one standard with too many characters" might be a problem...but consider "too many standards with not enough characters" instead: This is a worse fate ;)...

Part of the "problem" here is that, basically, what a lot of programmers like us have been doing for many years is to write ASCII programs with English (or French or German, which are also Latin-based so there's no big difference, except for a few "accents" and that funny "double s" character, which is officially being "phased out" of German, anyway, I notice with those "spelling changes" introduced on July 1996)...and we haven't given the slightest thought or done anything to do with "Internationalisation" at all, really...BUT, for those that have been doing this, then it was a _LOT_ of work to support it all...and that's where UNICODE comes in...it was all started by a bunch of people who'd been working on things like "international word-processor" software and such, to try to, once and for all, get rid of all the "Babylon confusions" of "code pages" and such...create one "big" character set which includes all of it...there, problem solved...and, okay, it's a little bit more complex than that (well, you know, all those languages people speak around the world just don't all work alike :) but this is much easier than the other "international" options...so, you know, if UNICODE looks "bad" then just imagine what it was like supporting everything else beforehand...

And, yeah, there's a lot there...and ASCII is, indeed, much simpler...but, well, "you get what you pay for"...ASCII is simpler but, well, you "lock out" people that way...ever noticed that we don't seem to have any Japanese coders posting here very often, or Israeli coders and others who use different alphabets...okay, there's Maxim and Alexei – Russian – on the alt.os.development group but, of course, they've _TAUGHT ENGLISH_ as "standard" in Russian schools for a long time (and are very good at it too ;)...but Maxim was telling me about how he actually has a Cyrillic keyboard but has a "shortcut key" which switches back and forth from Cyrillic to Latin (and he just "knows" where the Latin characters are, from practice...mind you, I never look at the keyboard while typing either, so you could mix up the keys and I'd still be able to type...indeed, on my Linux machine, I popped out the keys because I thought I'd try that "Dvorak" layout and pushed the keys back in the different layout...but then, once I installed Linux, I wanted QWERTY back...so popped out all the keys and pushed them back in the QWERTY way...except, oops, I popped "M"

Re: Unicode Support

and "N" back into the wrong places and they've swapped around...nevermind, it makes no difference to me, really, as I never look at the keys while typing...but, oh dear, it's confusing for everyone else! ;)...hence, yeah, for `_US_ASCII` is much simpler...but it's probably absolutely NO COINCIDENCE whatsoever that those who generally show up on this newsgroup and such, all have Latin-based alphabets or are taught compulsory English in schools and so forth...

> ['Babylonian confusion']
> | `_THIS_` is the real problem, of course...
>
> | BUT is this a problem that's the tool's responsibility? For instance,
> | programming a protected mode OS is very confusing too...or using
> "direct
> | access" to hardware...should tools also start telling us we can't use
> "LGDT
> | /LIDT" or "IN / OUT" instructions too? As they might cause "Babylonian
> | confusion"?
>
> My new disassembler will show `PL0/IOPL/PM/64`-only instructions
> as "force `EXC0D/06/00..`" if the mode configuration wont match.
>
> But this aren't language issues, except Intel-manuals are written
> 'that' confusing, it may even native English speakers drive crazy. :)

Oh, don't worry...when it comes to typical "technical English" and what's written in many manuals with lots of jargon and bad descriptions, then, yes, it might as well be a different language...and there's not much advantage to being a native English speaker at all ;)...

But, yeah, back to the main point: This is what I was trying to say...a tool `_shouldn't_` really be "dictating" these kinds of things...it should provide the "facilities" and then the programmer decides how to use those facilities properly (or not to use them at all)...

You could say that non-ASCII characters are "confusing"...but many would say protected mode is "confusing"...so, if a tool can say "we shall have no more non-ASCII characters!", then is this also saying that the tool should say "we shall have no more access to protected mode!"...

This is "nanny" mentality..."tool knows best"...if `_I_` say it's too "confusing" for you then you can't use it...this just isn't right...

I mean, let's say that I think "direct hardware access" is "too confusing"...so, I come along and "for your own good", I remove all the "IN / OUT" instructions from your assembler...what would you feel about that? That I'm "doing things for your own good"...or that I have "no right" to be doing that and I'm a patronising cow, who should "piss off" and leave you to do what you want to do?

This is the "principle" here...this just isn't what a tool is supposed to

Re: Unicode Support

do...it might not support something because, you know, it's supposed to be "an applications assembler" (so, only 32-bit flat mode and no privileged instructions is a consequence of it being an "applications assembler" – fair enough – not because anyone is thinking "this is too confusing for people, so let's remove to make things easy")...but to "ban" or "limit" something because you feel it's "too confusing"? To be "nanny" for people, who, in fact, you've never met and don't know what they do or don't want?

You know, the greatest wisdom is often to know when to SHUT UP...the greatest power is to know when NOT TO ACT...often, the very best thing a government can do is just to piss off and leave the people to do whatever they want...

Well, perhaps one word makes the point clear: "Clippy" ...

Yeah, "Clippy" was meant to make things "easier" by being your "personal nanny" as you work...he was meant to be "helpful"...now, that's what "Clippy" was supposed to be...but what was he ACTUALLY, in practice, not just for "expert" but also "newbie" users? One of the most annoying creatures ever created on the face of planet Earth...universally loathed by literally millions of people...

And, yes, wasn't "Microsoft Bob" a great idea?

Starting to see the "pattern" here yet? If you do, well done, because Microsoft still ain't worked it out yet...

Sometimes, the most "helpful" thing a person can do for someone else is to, in fact, do absolutely nothing...leave them be...you know, wait until you're asked to lend a hand, without rushing in, changing everything and, well, not really knowing what you're doing...

For the Brits reading, it's the "Brittas Empire" syndrome...for everyone else, that was a comedy show – starring Chris Barrie (Lara Croft's Butler in the movie and the "smeghead" hologram in Red Dwarf :) – about this manager called "Brittas"...and, basically, the joke of the show was that this manager was utterly, utterly incompetent...lethally so...everything that he "managed" would crumble, collapse, explode, get killed or otherwise fall apart completely...but – worryingly, very much like an awful lot of "managers" out there – he was convinced that he was the smartest, cleverest and best manager in the whole world...one of those people who thinks they know best and interferes in everything...but the second they come along and interfere, they don't help at all...they actually make things a million times worse...

Well, sometimes when I think of Bill Gates and Microsoft, then this "Brittas" character comes to mind...convinced they "know best"...convinced everyone else is an "idiot", wo doesn't even know how to tie their shoe laces...and then they come in to try to "help"...and make things 7 million times worse...because, in fact, the only "idiots" are them but they are too

Re: Unicode Support

idiotic to even realise this...

- > | Or should the tools `_support_` these things ...
- > | But why should a tool be forcing a Russian programmer, ...
- >
- > Wouldn't it help a lot if the whole globe talks only one language?

No, not in the slightest...I find that a terrible, horrible, awful and repugnant idea...all that history, all that diversity, all those `_DIFFERENT WAYS TO LOOK AT THINGS_` lost, everyone in the world forced into a single "NewSpeak" language? It would be a terrible oppression...spiritual bankruptcy...we should NEVER give in to such "trans-culturalism" that tries to "expunge" peoples, cultures, histories from existence, simply because it makes "mass-production" a little easier and profits larger...no, no, no...never...it's an awful idea...

And it would be an impossible idea too...remember that many languages are "distortions" of the same language as it became "separated" by hills, mountains, rivers, national borders and so forth..."dialects", "accents", "slang" and such...you know, that English `_IS_` Germanic but it "split off"...and then along came French and Norse and this and that...and it "evolved" from there...the "language" that just naturally came from people speaking all those different languages trying to speak to each other (very much part of the reason, for sure, that English is more "simplified", having lost all its "word endings" and things like that...because it was simply impossible to "resolve" German word endings with French word endings and so on and so forth...so, simply, they just got "dropped" because no-one could "agree" on which way it should be done...also, probably why English is a pretty good "Lingua Franca" in that it's not as difficult as many languages are to be learnt...because it literally did start as a "common tongue" between peoples who all spoke different languages...a "common ground"...you know, where Esperanto was "invented", English is its natural counterpart...of course, it's where it is because of the British Empire and America's "superpower" status and so forth...BUT, at the same time, the language itself `_IS_` actually "appropriate" in that it's "simpler" than most languages...indeed, one of the most difficult parts of English is the pronunciation and spelling have no relation to each other...but, in fact, that `_ALSO_` comes from English being the product of a "clash of languages"...and that, for a long time, it was a "spoken only" language...and when written, there were no "standards" for a long while...it's an infamous point that Shakespeare, in fact, spelt his own name some 20 different ways in different places! :)...

What might be good is if the whole world had a shared `_second language_`, perhaps...and through this language people could always communicate...but the willing destruction of languages and culture and history? Absolutely NOT...

When Tolkien created his languages, he wrote the stories as a "background" to the people who spoke those languages...because he believed that languages and the people who spoke it and their culture were intrinsically

Re: Unicode Support

linked...bound together...and he was exactly RIGHT...

For example, here in Wales, there's the Welsh language which is very ancient and one of the oldest European languages...the English, earlier in history, attempted to deliberately exterminate it...punishing children caught speaking it...burning any books with the language in it...an actual 100% deliberate campaign of language extermination with the goal of its utter extinction...and they came very, very, very close to succeeding...but, in secret, parents would teach their children the "forbidden" language...books would be hidden and concealed...people would speak the language in secret...today, from near extinction, around 25% of the people of Wales are able to speak Welsh...

Which is good because the only actual historical account of the real person who was "King Arthur" is, of course, written in Welsh...the original Merlin legend are written in Welsh too...there are the myths and legends of "the dragons" who fought (and the red dragon is what sits on the Welsh flag from this tale :)...the bardic triads...centuries of history and culture...

And if the English had succeeded in their quest to kill Welsh? It would all be lost...no-one would be able to understand it or read it...an entire culture and, in effect, people lost completely...the English were not the only ones to realise that destroying a language destroys a people, as the Roman Empire before them also came to Wales and headed straight for the Druids – the spiritual leaders and keepers of knowledge in Celtic Britain – and they slaughtered them...every last one...

What is known of the Druids and their ways? In fact, very little...next to nothing...all that exists to document them are what the Romans wrote...about their dress...where they lived...that they were spiritual leaders and the "wise men" who retain all the diverse knowledge of the their people (knowledge was passed orally through stories)...indeed, if you look at the "jedi knights" or the shape of Merlin then they are clearly partly "Druidic" in influence...certainly in the simple robes they wear...part druid, part monk, part knight...

Who knows what stories they had to tell? They were known to have herbal remedies and recipes for medicines...all lost...completely gone...really, next to nothing remains but second-hand Roman accounts of them...everything else the Romans obliterated deliberately...they would, you see, have no "rivals" to their power...no "split loyalties" from the British: They would worship the Glory of Rome and the Emperor...so, the Druids – as local leaders to whom the people may have "loyalty" to – were slaughtered...every last one...every last bit of their knowledge...all of their history...

For example, who built Stonehenge (which is actually older than the Pyramids at Giza, in fact)? What was it for? Who made the large chalk figures on hills nearby? What was before the Romans?

Thanks to their quite deliberate massacre – which was exactly for this one aim – no-one actually knows...almost everything about pre-Roman Britain is

Re: Unicode Support

from archeology alone...if the Druids kept any accounts of this history, they must have been destroyed by the Romans...all the stories that would have been "passed on" in myths and legends and tales of history: DEAD...

Some people in modern times pretend to be "druids" and that they are enacting "druid rituals"...it's all made up in modern times...a bit like the modern "wicca" movement, who claim to be "carrying on ancient pagan traditions"...completely nonsense...it is based on no historical evidence or accounts for these practices...exactly because there are NO evidence or accounts for either of these...or what they really might have done...modern inventions with no basis in any historical reality...

And, note, there was no accident in this...the Romans were utterly deliberate in this goal...what is the saying? "The victors write the history books"...

To willingly surrender the past? The culture? The people? To hand over your Forefathers...to what? Making "mass-production" easier with "trans-culturalism" because it's easier to make a bigger profit when you can just solely print English boxes and English manual "in bulk"? Sorry, but do you really believe that the spread of English through commercialism is entirely "accidental"?

"Those who do not know history are doomed to repeat it"

- > A long way to go, but when I look back 50 years, the world seems
- > to tend to everyone learn to understand written English at least,
- > local US/UK/AUS pronunciation excluded here of course :)

Hail Ceasar! To the Glory of Rome!

Et tu, Brute?

- > [..]
- > | ... if we don't agree on that, then things become "Babylonian
- > | confusion" here very, very quickly, nicht wahr? ;)...
- >
- > That's very true indeed*, isn't it? *)indead for the French :)

Oh, no...I salute the French that, every year, they enter a French language song in the Eurovision...they did, one year, break this and put in an English language song...I was VERY disappointed in that...don't do that ever again, France! It is one of the most beautiful sounding languages on the planet...everything spoken in it sounds like pure poetry...all that history and culture of centuries can be heard in each word...

Let us not forget that Liberty is a French girl...let us NEVER forget that...NEVER...

Mind you, the best Eurovision entry that never won was the bald surrealist Guildo Horn...everyone says that the Germans have no sense of humour...and

Re: Unicode Support

then they enter a completely insane man, who runs around the stage and climbs up the wall, ringing cow bells...it was hilarious...I voted for him...but, no, everyone else couldn't understand that the Germans do have a sense of humour...instead, everyone said it was "terrible that the Germans would insult Eurovision"...oh, come on...it's Eurovision...the campness and crapness is the whole point...that's what so great about it...

Mind you, the British were also told off by other countries taking it all far too seriously...because our commentator, Terry Wogan (who's actually Irish but is a TV presenter in Britain that he does the commentary for us, not Ireland...and he's actually done it every year for some 20 years or something, so, to the Brits, is as much a part of "Eurovision" as anything else :), has this "cheeky" way about him...he pokes fun at this and makes a joke of that...very sarcastic in his comments...it's his style and it can be very funny...but, truth is, he really does like the Eurovision and thinks it's great (as noted, he's been doing a very long time, every single year, and if he really didn't like it, he'd have given up the job long ago :)...but, you know, he just doesn't see it all as "serious" at all...you know, if the Croatian entry walks out wearing a really silly-looking costume, he pokes some fun at it...but, a year or two ago, he was "caught" by some other commentators from other countries he was sitting next to, making his sarcastic comments and having some fun...and they took it all very, very seriously and "reported" him to the Eurovision "authorities" or something, asking that he be "banned" or something...

Although, thinking about it, I've just remembered what our entry is this year...oh dear...oh dear, oh dear...it's page 3 model "Jordan", who seems to want to show that she has "talent" at singing and isn't just a page 3 "bimbo" (of course, she isn't actually talented at all and is just a page 3 "bimbo"...but, nevermind, as long as she's happy, let's not "burst the bubble" for her, eh? ;)...and, well, they put the songs to a "phone vote" to decide who will represent Britain...and she won (I suspect because a lot of men phoned up in order to be sure to see her again on Eurovision ;)...so that's going to be interesting on Eurovision night, when all the other countries are confused as to why Britain seems to have entered a "pole dancer"-like act into the Eurovision...

See? Think about it...if everyone spoke English then we wouldn't have those hilarious things on Eurovision, where they cut over to some over-enthusiastic Swedish presenters who make bad jokes that aren't funny...before getting confused about who's supposed to be talking and then talk over each other...and then they cut over to Moscow, where the presenter is just sitting there, unaware that the camera has come on and everyone can see them...and then the Swedish presenters are saying "hello? Hello Moscow! Hello?"...and it all falls apart in a very "cheap and cheerful" not-very-professional way that's just "Eurovision" through and through...

Now, see...THAT is "culture"...well, kind of...by the way, has anyone worked out how Turkey and Israel are in the EUROvision? Since when was Israel European? They really, really "bend the rules" for some of the

Re: Unicode Support

countries entering, eh? At least Turkey is technically half-on the European plate and is trying to join the EU...but Israel? We might as well throw the doors open and have Canada and America and Japan and everyone joining too!

And, think about it, if everyone spoke English all the time, then we wouldn't have all that "dix points" and "nul points" stuff...that's, in a funny way, part of the British vocabulary now...if you want to say something's crap then it's "nul points" and it's "dix points", if it's great (even though, since all the extra countries have been added to the Eurovision, it's now gone up to 12 points, as the top marks)...

Oh, one last thing: Could countries actually vote for the `_music_` rather than the "block voting" for each other's neighbours in a "political" way? Oh, look, all the Eastern European countries are voting for each other!

Although, that said, when Britain got "nul points" – absolutely no-one voting for us at all (and the song wasn't that bad...and we normally get at least one or two "political" votes, even if we're utter crap ;) – last time, while Blair's following Bush into Iraq...yeah, don't worry...we "got" that one and let's say we don't necessarily "disapprove" of that kind of "political comment" through the voting...goodness knows what reaction "Jordan" is going to have, if Bucks Fizz pulling off their skirts got such a big reaction before...oh dear...oh dear, oh dear ;)...

I reckon, though, that Ireland really should enter "My Lovely Horse", as their "Euro song"...ah, I Love the Eurovision...it's brilliant! ;)

Beth :)

"We're still being challenged in Iraq and the reason why is a free Iraq will be a major defeat in the cause of freedom."

[George Walker Bush Jnr., explaining his Iraq war policy]

.

- *Follow-Ups:*

- ◆ **Re: Unicode Support**

- ◇ *From:* wolfgang kern

- *References:*

- ◆ **Unicode Support**

- ◇ *From:* Chewy509

- ◆ **Re: Unicode Support**

- ◇ *From:* wolfgang kern

- ◆ **Re: Unicode Support**

- ◇ *From:* Chewy509

Re: Unicode Support

◆ **Re: Unicode Support**

◇ *From:* wolfgang kern

- Prev by Date: **Re: Awe diddums, fearless leader has gone into hiding.**
- Next by Date: **Re: Is Betov trying to rip off the FASM source ?**
- Previous by thread: **Re: Unicode Support**
- Next by thread: **Re: Unicode Support**
- Index(es):
 - ◆ **Date**
 - ◆ **Thread**