Tamil in Unicode

V Krishnamoorthy
(Former Professor of Anna University)
Inforeed (Information Research And Education)
30(4 C) Second Main Road, Gandhi Nagar, Adayar, Chennai 600 020 India

---

ABSTRACT
Some of the shortcomings of the present scheme for Tamil in Unicode is pointed out first. Two alternative schemes are presented here for Tamil for possible adoption in the Unicode. One uses 384 locations. The other uses the same 128 locations provided for Tamil, and extends the existing scheme by including the pure consonants. The advantages of these schemes are as follows. In the first scheme, the space requirement is just around 70% of what is needed in the current Unicode scheme. The manipulations needed for the NLP applications are enormously simplified and speeded up, as the vowel and consonant parts are available as blocks of bits in the code itself. The second scheme facilitates the splitting of a letter as consonant and a vowel in the right way. It will reduce the memory required by about 10%. No new space is required, and it is backward compatible with the current code. But this will not be as efficient as the first scheme.

1. Some shortcoming of the Tamil coding in the Unicode

2. The Tamil scholars have very aptly defined consonants and vowels. Except for the aaytham, the other Tamil letters are a combination of a consonant and a vowel. Unfortunately, the consonants do not find a place in this scheme. Instead consonants with the vowel a is included. This leads to the unnatural way of representing a pure consonant as a combination of one letter with another symbol pulli. It has been pointed out by others that this leads to problems while doing processing, due to its incorrect representation. Should we forget what was discovered thousands of years ago?

3. There is a character called ow length marker. Note that to represent the vowel ow, there is already a matra present in the scheme. What is introduced here as ow length marker is nothing but a glyph. But Unicode is supposed to code only characters and not glyphs! It should be noted that the reason behind this inclusion is differentiating this glyph from the glyph for the letter la. But one cannot differentiate between these two, as in today's writing, both the glyphs are identical. Even if we change them into two differently looking glyphs, a glyph should not find a place in the Unicode.

4. The explanations given for the creation of the glyphs for koo etc. are not worded properly. It gives the meaning as if the character koo is equivalent to kee and the thunai ezhthu.

5. The rendering of old type of letters for lai, raa etc. are provided. One has to recall that many years ago, the government of Tamil Nadu, by Government notification has changed all these. This old type rendering should not find a place in a current document.

6. There is nothing called anushwar in Tamil. This is used only in some other Indian languages, and not in Tamil. The Tamil code starts with this wrong symbol.

7. The aaythm is shown as a vowel modifier, by including a dotted circle, which is not correct. It is an independent letter.

2. Scheme 1

To make the use of Tamil very effective in terms of memory, it has been already proposed by a few that all the Tamil letters should be given individual positions in the Unicode scheme. The scheme proposed below is an extension of that thought. Whereas in such schemes of others, the letters are put one after other, in a continuos sequence, here we use a different approach. This leads to enormous simplicity in programming and processing.

The design of this scheme is influenced by the way the ASCII code is designed. In the ASCII code the lower case English alphabets are not put immediately following the upper case alphabets. It is put in such a way that the corresponding upper and lower case letters differ exactly in 32 positions. This will make the conversion of one from the other just by bit manipulation, which is faster than the table look up.

In the case of Tamil, this principle of 'use bit manipulations to speed up processing' can be used more effectively, as given below.

In Tamil, each consonant combines with 12 vowels. Including the pure consonant, there are 13 letters for each consonant. The aaytham and the 12 vowels also form 13 letters. There are 18 pure Tamil consonants and 5 grandha consonants. The grandha letter sri stands alone. So, there are 24 blocks of 13 letters each, and one single letter. Apart from these, the symbols for the numerals, and day, month etc. are to be accommodated. They number about 20. The arrangement of these letters and symbols, each block of 13 letters in one block of 16 places, is the basic idea in this scheme. Since there are 24 blocks exactly 128 x 3 = 384 positions are enough for the Tamil letters. In case we can get 128 x 4 = 512 places, the symbols can be accommodated after all the Tamil letters, and this will be the ideal scheme. In case we have to settle for 384 places, then, as in the case of ASCII, the remaining 3 positions in each block can be used for symbols,. The scheme is given below.

The code for a Tamil letter is given as follows. Assume that abcd efg0 0000 0000 is the starting location for the block provided for Tamil. Consider the position given by the 16 bit binary number abcd efgh ijkl mnop. Consider the Tamil letter, say kaa. This has the first consonant and the second vowel in it. Then efg0 0001 0010 gives the position of kaa. Here the binary number mnop (between 0 and 12) gives the vowel present in the letter. Also the binary number hijkl, which is between 1 and 23, gives the consonant present in that letter. If one of these numbers is zero then it represents a letter which is a pure vowel or a pure consonant. mnop = 13 and hijkl = 0 gives the position of the aaytham. mnop = 13 and hijkl = 23 gives the position of the the letter sree. The symbols for rupee, number, merpadi, date, month, year, debit, credit and Tamil numerals can be kept in the last row with mnop = 15.

In this scheme, the vowels come first. Then the aaythm comes. Then the uirmei letters of a particular mei follow that mei. this happens for the 18 + 5 consonants. then the letter sree comes. As in the case of the ASCII code for English, symbols come inbetween. Leaving these symbols, the Tamil letters come in the correct order. Note that since the uirmei letters depend on the mei letters, it is natural that the mei comes before the uirmeis.

The following is in the tabular form. The first 7 bits abcd efg are not shown here.

hijkl mnop	0	1	2	3	4	5	6	7	8	9	10	11
0		க்	ங்	ச்	ஞ்	ட்	ண்	த்	ந்	ப்	ம்	ய்
1	அ	க	ங	ச	ஞ	ட	ண	த	ந	ப	ம	ய
2	ஆ	கா	ஙா	சா	ஞா	டா	ணா	தா	நா	பா	மா	யா
3	இ	கி	ஙி	சி	ஞி	(r)	ணி	தி	நி	பி	மி	யி
4	ஈ	கீ	ஙீ	சீ	ஞீ	ட	ணீ	தீ	நீ	பீ	மீ	யீ
5	உ	கு	ங	சு	ஞு	டு	ணு	து	நு	பு	மு	யு
6	ஊ	கூ	ஙூ	சூ	ஞூ	டீ	ணூ	தூ	நூ	பூ	மூ	யூ
7	எ	கெ	ஙெ	செ	ஞெ	டெ	ணெ	தெ	நெ	பெ	மெ	யெ
8	ஏ	கே	ஙே	சே	ஞே	டே	ணே	தே	நே	பே	மே	யே
9	ஐ	கை	ஙை	சை	ஞை	டை	ணை	தை	நை	பை	மை	யை
10	ஒ	கொ	ஙொ	சொ	ஞொ	டொ	ணொ	தொ	நொ	பொ	மொ	யொ
11	ஓ	கோ	ஙோ	சோ	ஞோ	டோ	ணோ	தோ	நோ	போ	மோ	யோ
12	ஔ	கௌ	ஙௌ	சௌ	ஞௌ	டௌ	ணௌ	தௌ	நௌ	பௌ	மௌ	யௌ
13	ஃ											
14												
15	Tamil symbols and Tamil numerals come in this row											

hijkl mnop	12	13	14	15	16	17	18	19	20	21	22	23
0	ர்	ல்	வ்	ழ்	ள்	ற்	ன்	ஸ்	ஷ்	ஜ்	ஹ்	க்ஷ்
1	ர	ல	வ	ழ	ள	ற	ன	ஸ	ஷ	ஜ	ஹ	க்ஷ
2	ரா	லா	வா	ழா	ளா	றா	னா	ஸா	ஷா	ஜா	ஹா	க்ஷா
3	ரி	லி	வி	ழி	ளி	றி	னி	ஸி	ஷி	ஜி	ஹி	க்ஷி
4	ரீ	லீ	வீ	ழீ	ளீ	றீ	னீ	ஸீ	ஷீ	ஜீ	ஹீ	க்ஷீ
5	ரு	லு	லூ	ழு	ளு	று	னு	ஸு	ஷு	ஜு	ஹு	க்ஷு
6	ரூ	லூ	வூ	ழூ	ளூ	றூ	னூ	ஸூ	ஷூ	ஜூ	ஹூ	க்ஷூ
7	ரெ	லெ	வெ	ழெ	ளெ	றெ	னெ	ஸெ	ஷெ	ஜெ	ஹெ	க்ஷெ
8	ரே	லே	வே	ழே	ளே	றே	னே	ஸே	ஷே	ஜே	ஹே	க்ஷே
9	ரை	லை	வை	ழை	ளை	றை	னை	ஸை	ஷை	ஜை	ஹை	க்ஷை
10	ரொ	லொ	வொ	ழொ	ளொ	றொ	னொ	ஸொ	ஷொ	ஜொ	ஹொ	க்ஷொ
11	ரோ	லோ	வோ	ழோ	ளோ	றோ	னோ	ஸோ	ஷோ	ஜோ	ஹோ	க்ஷோ
12	ரௌ	லௌ	வௌ	ழௌ	ளௌ	றௌ	னௌ	ஸௌ	ஷௌ	ஜௌ	ஹௌ	க்ஷௌ
13							ஶ்ரீ					
14												

15 Tamil Symbols and Tamil Numerals come in this row

The advantages of this 384 place scheme are enumerated below.

1. In Tamil, a letter can be a pure vowel, a pure consonant, or it can be formed by combining a consonant and a vowel. Also, a word is formed by combining many parts, sometimes as much as eight parts. In natural language processing, while splitting a word to get its components, the splitting rules, which are the reverse of the combining rules, often needs information of the following type, about a letter.
1. Whether a letter is a pure vowel?
2. Whether a letter is a pure consonant?
3.Whether a letter is a ukaram?
4. Whether a letter has a particular consonant?
5. Whether a letter has a hard consonant?
6. Whether a letter is the soft consonant pair of a given hard consonant?

In all these cases, the answers can be got just by checking some bit positions, since both the vowel number and the consonant number are directly available as blocks of bits in the code itself. This will lead to enormous simplification in the programming and also in the Natural language processing. Note that NLP is going to be the hot topic in the coming years. And this may effectively seal the fate of many languages. Note that in the case of table look up it may take more time for all processing.

2. In NLP, many times it is necessary to combine a vowel and a consonant, or to split a letter into its component vowel and consonant. This can also be achieved by bit manipulations.

3. The memory needed to store any text will be just around 60% of what will be needed if the existing Unicode is used. Obviously the communication time required also becomes just half.

4. If 512 locations are used, no special algorithm is needed to sort in the Tamil order. In the case of 384 places, if the special Tamil symbols are not used in the text, any normal sorting algorithm will work without any modification. In case the special symbols are also used, then they may have to be separated first, and then the text and the symbols have to be sorted using any commonly available algorithm, and then they have to be combined. Since in most of the cases, the symbols are not used, any sorting algorithm will work. This will be a boon since any data base software can be used even for Tamil sorting, without any modification.

1. It is shown that the proposed scheme scores over all the other schemes, in terms of memory, communication speed, and processing speed. These three are very important factors which may be crucial in determining whether a language is going to be a language used effectively in the future.

3. An alternative scheme

We have already noted that the ow length marker at position 0BD7 is not necessary. It is not at all a character, and hence can be safely removed from the code. This gives 25 continuous vacant places from 0BCE to 0BE6. The pure consonants 18 + 5 + 23 in number can be accommodated in this slot,

say, starting from position 0BCE. The Grandha letter sree is treated as a single character in Tamil. As such one position has to be given for that, after these 23 places.

This will avoid the problem in processing. Also the both the pure consonants and the half consonants (those with a) can be represented by a single codes. This may save about 10% of the space. Backward compatibility will be maintained. Though this is obviously not the best choice, this is also being proposed purely because of its backward compatibility.

4. Conclusion

Some of the drawbacks of the preset Unicode scheme for Tamil has been pointed out. One scheme proposed uses 383 places and is shown to be best suited when we consider the future of Tamil in a competitive world. Another scheme is presented which is an extension of the current scheme. Though this is not the best, it offers backward compatibility, and solves one important problem. The Tamil community should take a decision after carefully considering the future requirement of Tamil processing in a highly competitive global scenario.

96 empty

அஸ்கி மற்றும் யூனிகோடு தமிழ்க் குறிமுறைகளின் சார்பு செயல்திறன் மதிப்பீடு

சு.சீனிவாசன்,
கணிப்பொறிக் கோட்டம், இந்திராகாந்தி அணுவாராய்ச்சி மையம்,
கல்பாக்கம்-603102, காஞ்சிபுரம் மாவட்டம், தமிழ்நாடு

முன்னுரை

அண்மையில் (ஜூன்,1999) தமிழக அரசின் பெருமுயற்சியால் கணிப்பொறிக்கான தமிழ் விசைப்பலகை அமைப்பும் உள்ளீடு செய்யப்படும் எழுத்துக்களுக்கான குறியீடுகளும் தரப்படுத்தப்பட்டுள்ளன. இதன் பயனாக, உலகத் தமிழர்கள் தரப்படுத்திய குறிமுறைகொண்டு தமிழில் தகவல்களைப் பரிமாறிக் கொள்ள வழி பிறந்திருக்கிறது.

தமிழில் உயிர், மெய், உயிர்மெய், ஆய்தம் மற்றும் வடவெழுத்துக்களின் தேவைக்கென மொத்தம் 313 எழுத்துக்கள் கையாளப்படுகின்றன. கணிப்பொறிகொண்டு இவற்றை அகர வரிசைப்படுத்தி ஆய்வதற்கு தனித்த குறியீடுகள் அமைப்பது அவசியமாகும். கணிப்பொறியில் இவற்றுக்கு குறியீடு அமைப்பதென்றால் குறைந்தபட்சம் 9 பிட்டுக்கள் தேவை. ஒன்பது பிட்டுகளைக் கொண்டு 512 குறியீடுகளை அமைக்க வழி உண்டு. யூனிகோடு முறையில் தமிழ் எழுத்துக்களுக்கு இட ஒதுக்கீடு செய்ய செயல்திறன் (efficient) மிக்க குறியீடு அமைப்பது அவசியமாகின்றது.

அஸ்கி குறிமுறை

கணிப்பொறிப் பயன்பாட்டுக்கென எண்களையும் கூட்டல், கழித்தல், பெருக்கல், வகுத்தல் முதலிய சிறப்புக் குறியீடுகளையும் சிற்றின, பேரின ரோமன் வரிவடிவுகளையும் கொண்டு உருவாக்கப்பட்ட குறிமுறை அஸ்கி (ASCII) குறிமுறையாகும். அஸ்கி குறிமுறையில் 8 பிட்டுக்களைக் கொண்டு குறியீடுகள் அமைக்கப்பட்டிருக்கின்றன. இதன் மூலம் 256 தனித்த குறியீடுகளை மட்டும்தாம் உருவாக்க இயலும். அவற்றிலும் முதல் 32 இடங்களைக் கணிப்பொறியின் கட்டுப்பாட்டுக்கு என ஒதுக்கிவிட வேண்டியிருக்கிறது. எனவே எஞ்சி இருப்பதோ 224 குறியிடங்கள் மட்டுமே. இவற்றைத் திறம்படப் பயன்படுத்துவதற்கு தமிழில் இரண்டு வழிமுறைகள் உருவாக்கப்பட்டுள்ளன. இவை சிற்பக் குறிமுறை(glyph encoding) வகையைச் சார்ந்தன. இக்குறிமுறையில் கால், ஒற்றைக்கொம்பு, இரட்டைக் கொம்பு முதலிய துணையெழுத்துக்களுக்கு தனித்த இடங்கள் ஒதுக்கப்பட்டுள்ளன.

ஒருமொழிக் குறிமுறையில் (monolingual coding) 147 குறியீடுகள் பயன்படுகின்றன. இதில் மெய் எழுத்துக்களுக்கும், இகர, ஈகார உயிர்மெய் எழுத்துக்களுக்கும் தனித்த குறியீடுகள் அமைந்துள்ளன. இவை மேசை அச்சுப் பதிப்புப் பணிக்கு (Desk Top Publishing) மிகவும் ஏற்றவை. ஆங்கிலம், தமிழ் ஆகிய இரு மொழிகளையும் ஒருசேர இணையத்தில்(Internet) பயன்படுத்தும் பொருட்டு இருமொழி (bilingual coding)) குறிமுறை ஒன்றும் உருவாக்கப்பட்டுள்ளது. இதில் 83 குறியீடுகள் பயன்படுத்தப்படுகின்றன. இவற்றில் மெய் எழுத்துக்களையும் இகர, ஈகார உயிர்மெய் எழுத்துக்களையும் உருவாக்க இரண்டு குறியீடுகள் தேவைப்படுகின்றன. இம்முறையில் தகவல்களைக் கோப்பில் (file) சேமிப்பதற்கு கூடுதல் நினைவகம் தேவைப்படுகிறது. இருப்பினும் தமிழோடு ஆங்கிலத்தை ஒருசேரப் பயன்படுத்தும் வாய்ப்பு இதில் இருப்பதால், இக்குறிமுறை இணையப் பயன்பாட்டுக்கு மிகவும் ஏற்றதாகிறது.

யூனிகோடு குறிமுறை

யூனிகோடு முறை (Unicode) பன்மொழிப் பயன்பாட்டுக்கு என உருவாக்கப்பட்ட குறிமுறையாகும். வரிவடிவமுடைய உலக மொழிகள் அனைத்திற்கும் இதில் இட ஒதுக்கீடு அளிக்கப்பட்டுள்ளது. இதில் எழுத்துக்களைப் பதிவுசெய்ய 16 பிட்டுக்கள் தேவைப்படுகின்றன. இக் குறிமுறையில் மொத்தம் 65,536 குறியீடுகள் சாத்தியம். இம்முறையில் துணையெழுத்துக்களையும் மற்ற எழுத்துக்ககளையும் இரண்டு பைட்டுக்களில்தாம்(அதாவது 16 பிட்டுக்கள்) பதிவுசெய்ய இயலும். தமிழ் மொழிக்கு 512 இடங்கள் ஒதுக்கீடு செய்யப்பட்டால், ஒவ்வொரு உயிர்மெய் எழுத்தையும் 16 பிட்டுக்களில்(2 பைட்டுகளில்) அடக்கிவிட முடியும். அதாவது கோ எனும் எழுத்தைச் சேமிப்பதற்கு 2 பைட்டுக்கள் போதுமானது. எனவே யூனிகோடு முறையில் தமிழ் எழுத்துக்களுக்கு இட ஒதுக்கீடு செய்ய செயல்திறன் (efficient) மிக்க குறியீடு அமைப்பது அவசியமாகின்றது.

தற்போது, யூனிகோடு முறையில் தமிழுக்கு 128 இடங்கள் (slots) மட்டுமே வழங்கப்பட்டுள்ளன. அவற்றில் 61 இடங்கள் மட்டுமே தமிழ் எழுத்துக்களை உருவாக்கப் பயன்படுத்தப்பட்டுள்ளன. இதில் ISCII குறிமுறை அடிப்படையில் உயிர்மெய் வடிவுகள் உருவாக்கப்பட்டுள்ளன. இம்முறையில் ஒவ்வொரு உயிர்மெய் எழுத்தும் இரண்டு குறியீடுகளாகச் சேமிக்கப்படுகின்றன. அதாவது ஓர் உயிர்மெய் எழுத்துச் சேமிப்புக்கு 4 பைட்டுக்கள் தேவைப்படுகின்றன. காட்டாக, கோ எனும் எழுத்தைச் சேமிப்பதற்கு 4 பைட்டுக்கள் தேவை.

கணிப்பொறியில் தமிழைப் பதிவுசெய்வதற்கு அஸ்கிகோடு முறையில் ஒருமொழி மற்றும் இருமொழி ஆகிய குறிமுறைகளும் யூனிகோடு முறையில் 128 மற்றும் 512 துளையிடங்கள் கொண்ட மேலும் இரண்டு குறிமுறைகளும் இருப்பதாகக் கொள்வோம். இந்த நான்கு குறிமுறைகளில் எது சிறந்தது என அறிய இவற்றின் சேமிப்புத் திறனை மதிப்பிட வேண்டியிருக்கிறது. இதற்கு பின்வரும் புள்ளி விவரம் உதவுகிறது.

அட்டவணை 1. தமிழ் உரையில் புழங்கும் எழுத்துக்களின் பயன்பாடு

எழுத்து வரிசை	பயன்பாடு(%)
1. அனைத்து உயிரெழுத்துக்கள்	7.35
2. அனைத்து மெய்யெழுத்துக்கள்	29.45
3. அகர உயிர்மெய் எழுத்துக்கள்	21.13
4. இகர, ஈகார உயிர்மெய் எழுத்துக்கள்(டி, டீ நீங்கலாக)	10.08
5. டி, டீ ஆகிய உயிர்மெய் எழுத்துக்கள்	1.39
6. உகர, ஊகார உயிர்மெய் எழுத்துக்கள்	12.93
7. ஆ, எ, ஏ, ஐகார உயிர்மெய் எழுத்துக்கள்	14.97
8. ஒகர, ஓகார, ஔகார உயிர்மெய் எழுத்துக்கள்	2.69

இணையத்தின் வழி இறக்குமதி செய்யப்பட்ட ஏறக்குறைய 4 இலட்சம் எழுத்துக்கள் கொண்ட பகுதியிலிருந்து தமிழ் எழுத்துக்களின் புழக்கம் பற்றிய பல அரிய தகவல்களைக் கணிக்க முடிந்தது. அவற்றின் சுருக்கமான விவரம்:

தமிழ் எழுத்துக்களின் பயன்பாட்டில் மெய் எழுத்துக்களும், அகர உயிர்மெய் எழுத்துக்களும் ஏறக்குறைய 50 விழுக்காடு ஆளப்படுகின்றன. மேலும் அட்டவணை 1-லிருந்து அஸ்கி-ஒருமொழி, அஸ்கி-இருமொழி, யூனிகோடு-128, யூனிகோடு-512 ஆகிய தமிழ்க் குறிமுறைகளின் சேமிப்புத்திறன் மதிப்பிடப்பட்டது. அஸ்கி-ஒருமொழியின் சேமிப்புத் திறனை 1 எனக் கொள்ள, மற்ற மூன்று தமிழ்க் குறிமுறைகளின் சேமிப்புத்திறன் முறையே 1.33, 2.85, 1.66 என அறிய முடிகிறது. இருமொழிப் பயன்பாட்டுக்கு உதவும் சிற்பக் குறிமுறையைக் காட்டிலும் பன்மொழி பயன்பாட்டுக்கு உதவும் யூனிகோடு-512 தமிழ்க் குறிமுறை 25 விழுக்காடு கூடுதல் சேமிப்பிடம் எடுத்துக்கொள்கிறது. இதில் அகரவரிசைப்படுத்துவது எளிதாகிறது. சிற்பக் குறிமுறையில் மறைமுகமாகவே அகரவரிசைப்படுத்த இயலுகிறது.

யூனிகோடு-128 தமிழ்க் குறிமுறையில் அகரவரிசைப்படுத்துவது எளிது என்றாலும் இருமொழி சிற்பக் குறிமுறையைக் காட்டிலும் இதில் 115 விழுக்காடு கூடுதல் சேமிப்பிடம் தேவைப்படுகிறது. இதில் மெய்யெழுத்துக்களின் வரிசை க, ங, ச, ஜ, ஞ, ட, ண, த, ந, ன, ப, ம, ய, ர, ற, ல, ள, ழ, வ, ஷ, ஸ, ஹ என அமைகிறது. இங்கு ஜ, ன, ற, ள, ழ ஆகிய மெய்யெழுத்துக்களின் இடநிலை (place) நம் மரபுவழிப்பட்ட நெடுங்கணக்கு முறையிலிருந்து வேறுபடுவதாக இருக்கிறது. அனைத்து இந்திய மொழிகளையும் கருத்தில் கொண்டு யூனிகோடு-128 குறிமுறை வடிவமைக்கப்பட்டிருப்பதால், இங்கு தமிழ் மெய்வரிசை ஒழுங்கு(order) கெடுகிறது. யூனிகோடு 3.0 திட்டத்தில் தமிழ் மொழிப் பயன்பாட்டுக்கு 512 துளையிடங்கள் வழங்கப்பட்டால் அகரவரிசையாக்கப் பணியைக் குறியீட்டு அமைப்பெண் வரிசையிலேயே நேரடியாக மேற்கொள்ளலாம். இதனால் தரவுத் தளமேலாண்மை (Database Management) பணியில் தேடல்-வினவல் வேகம் கூடும்.

யூனிகோடு-512 தமிழ்க் குறிமுறையில் 313 எழுத்துக்களோடு 12 தமிழ் எண்களையும் இடம்பெறச் செய்ய வேண்டும். தமிழ் எண்களில் பூச்சியத்திற்கு வட்ட வடிவுடைய குறியீடு அமைப்பது அவசியமாகிறது. இந்திய-அராபிய எண்களைப்போல 0, 1, 2, 3, 4 ,5, 6, 7, 8, 9 என்ற முறையில் தமிழ் எண்களையும் வரிசைப்படுத்த வேண்டும். தமிழ் எண்களின் வரிசையைத் தொடர்ந்தே உயிரெழுத்துக்கள் குறிமுறையில் இடம்பெற வேண்டும். பத்தினைக் குறிக்கும் ய எனும் தமிழ் எண்ணை க0 என்று எழுதும் போக்கும் பின்பற்றப்படுகிறது. பன்மொழித் தடத்தில் தமிழ் வேரூன்ற இத்தகைய முயற்சிகள் அவசியம் தேவை.

முடிவுரை

அஸ்கி-ஒருமொழியின் சேமிப்புத் திறனை 1 எனக் கொள்ள, மற்ற மூன்று தமிழ்க் குறிமுறைகளின் சேமிப்புத்திறன் முறையே 1.33, 2.85, 1.66 என அறிய முடிகிறது. இருமொழிப் பயன்பாட்டுக்கு உதவும் சிற்பக் குறிமுறையைக் காட்டிலும் பன்மொழி பயன்பாட்டுக்கு உதவும் யூனிகோடு-512 தமிழ்க் குறிமுறை 25 விழுக்காடு கூடுதல் சேமிப்பிடம் எடுத்துக்கொள்கிறது. இதில் அகரவரிசைப்படுத்துவது எளிதாகிறது. சிற்பக் குறிமுறையில் மறைமுகமாகவே அகரவரிசைப்படுத்த இயலுகிறது.

100 empty

Tamil Encoding in Unicode - A Comparative Study

P.Chellappan
Palaniappan Bros., 14, Peters Road, Chennai - 600014
<Email: chellappan@vsnl.com>

INTRODUCTION

During the TamiNet99 Conference, which was held in Chennai, several papers were presented regarding the need to change the present Encoding of the Tamil Script in Unicode that occupies the Unicode locations U+0B80 to U+0BFF. Among the people who wanted a change, there were two schools of thought. There was one group that wanted assignment of unique locations only for the Uyir, Ayutham, Mei and Grantha characters (12+1+18+5+1). The other group wanted allocation of space for all the 313 Tamil characters. The author of this paper had also presented a paper calling for an encoding scheme that has a unique allocation for each of the 313 Tamil characters (Uyir, Mei and Uyir-Mei including the Grantha characters) and also for other Tamil Symbols. The purpose of this paper is to make a comparative study of the three different encoding schemes, so that a decision can be taken at the earliest.

THREE SCHEMES

1) THE PRESENT UNICODE SCHEME

This scheme allocates unique locations for the Uyir, Akaram Eriya Mei, Vowel Modifier and Symbol. Instead of treating an Uyir-Mei character as a combination of a Mei character and an Uyir character, it is treated as a combination of an Akaram Eriya Mei character and a Vowel modifier character. It also treats the Tamil Grantha characters 'ksha' and 'sri' as conjunct consonants. In order to be compatible to the other Indic scripts, the allocation of these characters are not as per the Tamil sort order. 128 locations are sufficient for this scheme.

Advantages:
All the Indic languages are allocated a block of 128 locations each and similar characters occupy the same relative location within the block. This enables easy transliteration possible between the Indic languages. Just a relative shift of locations would be sufficient for transliteration from one Indic language to another.

It helps in Natural Language Processing, Spell Check etc since the Uyir-Meis are already split into its basic components.

Disadvantages:
Since the Uyir-Meis are represented as combinations of an Akaram Eriya Mei and a Vowel Modifier each one of these characters would take up 32 bits (16 bits each for the Akaram Eriya Mei and Vowel Modifier characters). This results in large file sizes and also poor efficiency in processing.

Because of the same reason, there is no 1:1 relationship between characters and glyphs. Hence Glyph substitution will be required for proper display rendering. Tamil cannot be implemented in Level 1 of Unicode like English and the CJK (Chinese, Japanese and Korean) languages.

It ignores the natural sort order of the Tamil script. Hence it requires a separate Weight Table for proper sorting.

Only softwares that are Tamil enabled can be used.

This scheme does not follow the proper Tamil Grammatical rules.

2) PROPOSAL 1

In this scheme unique locations are allotted only for the Uyir, Mei, Grantha and symbol characters. The proper grammatical structure of Tamil is implemented in this scheme. All Uyir-Mei characters are represented as combinations of a Mei and a Uyir character. 128 locations are sufficient.

Advantages:
It helps in Natural Language Processing, Spell Check etc since the Uyir-Meis are already split into its basic components.

It maintains the Grammatical Structure of the Tamil language.

Since the proper sort order is maintained while allocation itself, straightforward sorting, without the need for a separate sort table, is possible.

Disadvantages:
File sizes are large since the Uyir-Meis are treated as combination character of Mei and Uyir character. This in turn leads to poor efficiency.

Since there is no 1:1 relationship between Characters and Glyphs, Level 1 implementation of Unicode is not possible.

Only softwares that are Tamil enabled can be used.

Transliteration to other Indic languages is slightly more difficult than the existing scheme.

3) PROPOSAL 2

This proposal envisages allocation of unique locations for each of the 313 Tamil characters and all the required Tamil Symbols. In this scheme all Uyir, Mei and Uyir-Mei characters including the Grantha characters are represented as single 16 bit characters (Unicode Characters) and not as combinations of Mei and Uyir characters. This proposal will require increase of the number of locations assigned for the Tamil script from 128 to 313+.

Advantages:
Since all the characters are represented only as 16 bit characters, the file sizes are smaller and as a result it is more efficient.

There is a perfect 1:1 relationship between characters and glyphs. Hence Tamil can be implemented even in a software that is Unicode Level 1 compliant. Literally all available softwares can be used for Tamil, without difficulty.

Sorting is easy and there is no need for separate Sort Weight Tables.

Disadvantages:
Since all Uyir-Meis are stored as single characters, one will have to use a mathematical manipulation to split it into its Mei and Uyir component. Hence at a first glance one will be led to believe that it is not suited for Natural Language Processing, Spell Check etc., But since efficiency is lost only in a memory operation as opposed to the loss of efficiency in a storage device read/write operation, this scheme still results in a better performance than the first two schemes.

Transliteration to other Indic languages is slightly more difficult than the existing scheme.

TESTING

The above comparison of the three schemes clearly shows that the all character representation scheme (Proposal 3) is the best. However all the theoretical discussions will have to be verified by proper testing.

Since both the Existing Scheme and Proposal 1 encode the Uyir-Mei characters as combination characters, efficiency of both these schemes would be similar except maybe in Natural Language Processing, Spell checking etc., where Proposal 1 could be better.

Hence as a matter of convenience, testing was done only to compare Proposal 1 and Proposal 2.

METHODOLOGY
As a preliminary testing process, a Pseudo Testing scheme was designed. A sample text of 25 pages was taken from an existing book and it was re-encoded according to Proposal 1 and Proposal 2 as show below.

Encoding:
Proposal 1 : Each Uyir and Mei character was encoded as a series of two bytes (8x2). The first byte would contain the Uyir or Mei character and the second byte would be blank.

e.g. அ = அ_ and க் = க்_

Each Uyir-Mei character was encoded as a series of four bytes (8x4). The first pair of bytes (16 bits) contains the Mei character and the second pair (16 bits) contains the Uyir character.

e.g. கி = க்_இ_ and = ச்_ஒ_

Proposal 2 : Each Uyir and Mei character was encoded exactly as in Proposal 1.

e.g. அ = அ_ and க் = க்_

However each Uyir-Mei character was encoded only as a series of two byte (8x2) characters.

e.g. கி = க்இ and சொ = ச்ஒ

The above two pseudo encoding schemes simulate the real situation fairly well.

The text derived from the above re-encoding process was used for testing various parameters that would affect the efficiency of the two schemes. For this purpose the following tests were carried out:

1. File size
2. Compressed file size using Pkzip
3. File copy using windows copy command (100 times)
4. Database Sorting of words from the text (20 times)
5. Database Indexing of words from the text (20 times)
6. Full word search for 'அவர்' in the complete text
7. Search for characters 'அன்' in any word in the complete text. e.g. in அவன்

TEST RESULTS
The results obtained from the above tests are tabulated below :

Sl.	Test	Proposal 2	Proposal 1	Difference
1.	File Size	116394 bytes	173904 bytes	49.41 %
2.	Compressed	35917 bytes	39467 bytes	9.88 %
3.	File Copy	1540 msecs	2080 msecs	35.06 %
4.	Database Sort	2310 msecs	3020 msecs	30.74 %
5.	Database Indexing	5490 msecs	7910 msecs	44.08 %
6.	Full word search	38450 msecs	58220 msecs	51.42 %
7.	'அன்' search	38010 msecs	57900 msecs	52.32 %

The pseudo test results are very clearly in favour of Proposal 2.

Other Languages: The concept of encoding all characters even if they are syllables, has been utilised by many languages. Primary examples are the Japanese Hiragana and Katakana script and the Korean Hangul Syllable block.

The Hiragana and Katakana Script allocates separate locations for syllable characters. e.g. 'ka', 'ki', 'ku', 'ke', 'ko', 'sa', 'si', 'su', 'se', 'so', and 'ta', 'ti', 'tu', 'te', 'to'.

Similarly, the Hangul Syllable block allocates a different location for each one of its syllables that are either a consonant-vowel-consonant combination or a consonant-vowel combination. In fact there are over 11172 such syllables which are allotted individual locations in Unicode (U+AC00 - U+D7A3). Apart from this the Hangul script also has a separate block called Hangul Jamo Blocl (U+1100 - U+11FF) which encodes the consonants and vowels alone without its combinations.

Another point to be noted is that the Canadian Syllabics have been alloted over 700 locations in Unicode 3.0

CONCLUSION:

Preliminary pseudo test results point clearly towards the All Character Encoding Scheme. But before proceeding further, it is necessary to test it out in the actual Unicode environment. This would require development of fonts and keyboard drivers. For this purpose the Tamils could come to a private understanding and use the End User subarea of the Private Use Area of Unicode (U+E000 - U+F8FF) for encoding all the Tamil characters. Once this testing is done, we would be in a position to take a final decision about how to proceed further. In case the results favour an All Character Encoding scheme, we should press further and get this implemented through the Unicode Consortium.

Author : The author is a partner of M/s Palaniappa Bros., which is one of the leading Tamil book publishing houses in Tamil Nadu. He is a Production Engineer with a Masters degree in Business Administration specialising in Finance and Information Systems. He has been involved in the fields of Font and Software development and DTP for over 15 years.

Contact : Palaniappan Bros.
14, Peters Road, Chennai - 600014, India.
Phone : 91-44-8268035, Fax : 91-44-8284067, eMail : chellappan@vsnl.com

106 empty

யுனிகோடில் வலியுறுத்துங்கள்

மா. ஆண்டோ பீட்டர்
கணித் தமிழ்ச்சங்கச் செயலாளர்
Softview Computers,
40, Nelson Manickam Road, Chennai 600 029, Tamilnadu, India
Email: svc@giasmd01.vsnl.net.in Internet: http//www.tamilcinema.com

---

தமிழ் எழுத்தாக்க பணிகளை மூன்று வகைகளாக பிரிக்கலாம்.

1. ஓலைச்சுவடிகள் காலம்
2. அச்சக அச்சுக்கோப்புகள் காலம்
3. கணிப்பொறி காலம்

சுமார் கி.பி.1517 ஆம் ஆண்டு வரை தமிழ்மொழி ஓலைச் சுவடிகளிலும் கல்வெட்டுகளிலுமே சேமிக்கப்பட்டது. அதன் நாகப்பட்டினம் தரங்கம்பாடியில் போர்ச்சீக்கீசிய பாதிரியார் உதவியுடன் அமைக்கப்பட்ட தமிழ் அச்சகம் மூலமாகவே தமிழ் மொழி காகிதங்களிலும் மறுமலர்ச்சி பெற்றது. ஓலைச் சுவடிகள் சரித்திரம் படைத்தவை. காகிதங்களில் அவை குடியேற காலங்கள் பல ஓடிவிட்டன. இன்றைய நிலையோ கணிப்பொறிக்காலம்.

இன்னமும் பத்து அல்லது பதினைந்து ஆண்டுகளுக்குப் பின் நாம் கையினால் பேனாவைப் பிடித்துத் தமிழ் மொழியை எழுதுவோமா என்ற சந்தேக நிலையே உள்ளது. அந்தளவுக்கு கணிப்பொறி வளர்ச்சி பெரிய அளவில் விஸ்வரூபம் எடுத்துள்ளது. விரலால் விசைப் பலகையில் எழுத்தைத் தட்டுவதே தமிழ் எழுதும் முறையாகக் கூட மாறலாம். மேலும் இன்றைய நிலையில் கணிப்பொறிக்கு ஏற்றபடி எந்த மொழி வளர்ச்சி பெறுகிறதோ அந்த மொழியின் வளர்ச்சி கண்டிப்பாக பிற்காலத்தில் குன்றாமல் இருக்கும். முதன் முதலில் எலெக்ட்ரானிக் சிட்டியை அமைத்த கர்நாடக மாநிலமோ அல்லது கணிப்பொறித் துறையில் முதலிடம் வகிக்கும் ஆந்திரப்பிரதேசமோ கூட நம் தமிழ்நாட்டில் தமிழ் மொழிக்கு மின்னணுவியல் துறையில் அளிக்கும் முக்கியத்துவத்தை அளிக்கவில்லை என்பது குறிப்பிடத்தக்கது. தமிழ் விசைப்பலகை நடுநிலையாக்கம் செய்ததே பெரிய சாதனை எனப் பலரும் எண்ணிக்கொண்டிருக்கிறார்கள் அது தவறு. கணிப்பொறித் தமிழ்மொழி நடுநிலையாக்கம் என்பது நம் தமிழ்மொழிக்கு உலக அளவில் கிடைத்த அங்கீகாரமே ஆகும். தமிழ் விசைப்பலகை அங்கீகாரம் ஒரு ஆரம்ப நிலையேயாகும். காகிதத்தில் அச்சிடப்படாத பல்வேறு ஓலைச்சுவடிகள், இலக்கியங்கள் ஆகியவற்றைமே கணிப்பொறியில் பதிந்தாலே இணையம் மூலமாக பல்வேறு செயலாக்கங்களைமே பலரும் எளிதாகப் பயன்படுத்திக் கொள்ளலாம். இதைத் தவிர அறிவியல், விளையாட்டு, மருத்துவம் மற்றும் தமிழ் உலகிலுள்ள அனைத்துப் பிரிவுகளைமே பொதுவாக கணிப்பொறியியல் இணையம் வாயிலாகப் பதிவு செய்து, கொசப்பேட்டை முதல் கொலம்பியா வரை பார்வையிட்டுக் கொள்ளலாம். எப்படியிருந்தாலும் இந்த நடுநிலையாக்கத்துக்காக வருங்கால சந்ததிகள் தமிழக அரசை கண்டிப்பாக மறக்காது.

கணிப்பொறி வளர்ச்சி நாளுக்கு நாள் மூங்கில் மரம் வளர்வது போலுள்ள வளர்ச்சியாகும். கணிப்பொறியின் எழுத்தமைப்பு முறையை சர்வதேச அளவில் பொது மொழிக் குறியீட்டு முறையாக 'னிகோட்' என்ற திட்டம் இன்னும் சில நாட்களில் உலகையே ஆளப்போகிறது. இந்த யுனிகோட் திட்டத்தில் தமிழுக்காக வலியுறுத்த வேண்டிய பல விஷயங்கள் உள்ளன.

தற்போதுள்ள ஆங்கில எழுத்துக் குறியீடுகள் கணிப்பொறியில் பதிவு செய்யும் போது 256 எழுத்துக் குறியீடுகளாக பதிவு செய்யப்படுகின்றன. இவை தமிழுக்காக பயன்படுத்த மற்றும் பதிவு செய்யும் போது ஆங்கில எழுத்துக்கள் இருக்குமிடத்தில் தமிழை பதித்து பயன்படுத்தப்படுகிறது. இவ்வாறு ஆங்கிலத்தில் மட்டும் பதிவு செய்யும் முறை ஆஸ்கி (ASCII-American Standard Code For Information Interchange) என அழைக்கப்படுகிறது. இந்திய மொழிகளை பதிவு செய்யும் முறை இஸ்கி (ISCII-Indian Standard Code For Informataion Interchange) என அழைக்கப்படுகிறது. இந்த ISCII முறையை மத்திய அரசாங்கத்தை தவிர பொதுமக்கள் யாரும் பயன்படுத்துவதில்லை. இந்த ISCII முறையானது இந்தியை அடிப்படையாகக் கொண்டு வடிவமைக்கப்பட்டது. இந்த ISCII முறையை இப்போது அப்படியே யுனிகோட் திட்டத்தில் விதைக்க நினைப்பதால் நாம் பல சிக்கல்களை தமிழ் மொழியில் எதிர் கொள்ள வேண்டியுள்ளது.

ஆஸ்கி கோட் முறையில் கணிப்பொறியில் பதிவு செய்ய 8 பிட்டுகளும், யுனிகோட் முறையில் கணிப்பொறியில் பதிவு செய்ய 16 பிட்டுகளும் அடிப்படையாக உள்ளது. யுனிகோட் திட்டத்தை நேர்த்தியான முறையில் ஆராய்ச்சி செய்து உலகிலுள்ள அனைத்து மொழிகளையும் உள்ளடக்கிய சர்வதேச பொதுமொழிக் குறியீடு ஒன்றை தயாரித்து அளிப்பதற்காக unicode consrtium என்ற அமைப்பு அமைக்கப்பட்டுள்ளது. உலகிலுள்ள பெரிய கணிப்பொறி நிறுவனங்கள், ஆராய்ச்சி மையங்கள், அரசு நிறுவனங்கள், கல்வியமைப்புகள் இந்த அமைப்பில் உறுப்பினர்களாக உள்ளனர். யுனிகோட் திட்டத்தில் தமிழக அரசும் உறுப்பினராக இருப்பது பாராட்டத்தக்க அம்சமாகும்.

யுனிகோட் மொழி மூலமாக பதிவு செய்யப்பட்ட கணிப்பொறி கோப்புகளை அனைத்து மொழி கணிப்பொறியிலும் பயன்படுத்தலாம். யுமலும் அகரவரிசை, தேடுதல் பணிகள், வரிசைப்படுத்தல், தகவல் சேமிப்பு, தகவல் பரிமாற்றம், குரல் உச்சரிப்பு ஆகியவற்றை அனைத்து மொழிக்கும் யுனிகோட் திட்டமாக ஒரு எளிமையான முறையாக கையாளப்படவுள்ளது. கம்ப்யூட்டர் என்றாலே ஆங்கிலத்தால் மட்டுமே முடியும் என்ற மாயையை யுனிகோட் திட்டம் முறியடித்து அனைத்து மொழிகளுக்கும் கணிப்பொறி பயன்படும்படி வடிவமைக்கப்பட்டுள்ளது. ஆனால்.....

தமிழ் நெடுங்கணக்கில் 30 தமிழ் எழுத்துக்களும், தமிழ் அச்சுக்கோப்பு முறையில் 314(247+67 வட எழுத்துக்களும்) தமிழ் ஆங்கிலக் கலப்பு கணிப்பொறிச் செயலாக்க முறைக்கு 95 எழுத்துக்களும் (Bilingual), தற்போதைய பதிப்பக தமிழ்நெட் 99 (Monolingual), முறையில் தமிழ்மொழிக்கு 213 எழுத்துகளும் ஒதுக்கப்பட்டுள்ளன. ஆனால் யுனிகோட் முறையில் தமிழ் மொழிக்கு அளிக்கப்பட்டுள்ள (பைனரி குறியீட்டு இடங்கள் OB80 முதல் OBFF வரை) இடமோ 128 தான். இதில் பாதிக்கும் மேற்பட்ட இடங்கள் காலியாகவே பயன்படுத்தப்படாமலே உள்ளன. உயிரெழுத்து 12, மெய்யெழுத்து 18, ஆயுத எழுத்து 1, இணைப்பு எழுத்துக்கள் 14, தமிழ் எண்கள் 12, வடமொழி 4 என 61 எழுத்துக்களே பயன்படுத்தப்பட்டுள்ளது. 128 இடத்தில் 67 இடங்கள் காலியாக உள்ளன. ஏன் என்று ஆராய்ந்து பார்த்தால் கணிப்பொறியில் ஹிந்தி மற்றும் மற்ற வடமொழிகளுக்கு ஏற்றப்படி யுனிகோட் திட்டத்தை அமைத்த முறையேயாகும். மேலும் சில தமிழ் (நடைமுறை) எழுத்துகளும் இல்லை. இலக்கணப்படி உயிரெழுத்தும் மெய்யெழுத்தும் சேர்ந்தே ஒரு தமிழ் எழுத்து பிறக்கிறது என்றாலும், தமிழ் மொழியின் ஒவ்வொரு எழுத்துக்கும் நாடி நரம்புகளும் இரத்த ஓட்டமும் உள்ளன. ஆகவே நாம் யுனிகோட் திட்டத்தில் ஒவ்வொரு எழுத்திற்கு தனி இடம் கேட்பது நம் கடமையாகும். யுனிகோட் முறைப்படி உயிரெழுத்துகளும் மெய்யெழுத்துகளுமே தமிழுக்காக அளிக்கப்பட்டுள்ளது. இதனால் நாம் கணிப்பொறியில் டைப் அடிக்கும் நான்கு பக்க தமிழாக்கங்களுக்கே ஒரு பிளாப்பியை டிஸ்க்கை பயன்படுத்த வேண்டிய நிலை வரலாம்.

உதாரணத்திற்கு ஒன்றைக் குறிப்பிடலாம்
'தினமணி' என்ற சொல் இப்போதைய கணிப்பொறி பதிவுப் படி நான்கு குறியீடுகளாகவே பதிவாகும். ஆனால் யுனிகோட் முறைப்படி 'தினமணி' என்ற சொல்லானது "த்+இ+ன்+அ+ம்+அ+ண்+இ" என

எட்டு குறியீடுகளாக பதிவாகும். நாளைய உலகை ஆளப்போகும் கணிப்பொறியில் இந்த முறையில் தமிழ் கையாளப்பட்டால் பதிவு செய்யும் முறையில் சிக்கல், பதிவுக்காக கூடுதல் கட்டணம் செலவிடும் முறை, கோப்பின் அளவு பெரிதாக்கப்படுதல், கணிப்பொறியில் படிக்கும் மற்றும் எழுதும் முறை தாமதமாதல், அகரவரிசைப்படுதலில் நேரம் கூடுதல், தேடுதல் பணிகளுக்கு நேரம் கூடுதல், தகவல் சேமிக்கும் பணிக்கும் நேரஞ்சுமை, வலுவில்லாத குரலமைப்பு, தகவல்களை ஓரிடத்திலிருந்து மற்றொரு இடத்திற்கு அனுப்புவதற்கே பல மணி நேரம் செலவிடும் அரிய நிலை ஏற்படும். இந்த குளறுபடி முறையிலும், ஒரு வேளை யுனிகோட் அயல்நாடுகளில் புகழ் பெற்றுவிட்டால் நம் தமிழ் மொழி புறக்கணிக்கப்பட்டு ஆங்கில மொழியை பயன்படுத்தும் நிலையும் (இப்போது உள்ளது போல்) ஏற்படலாம்.

தமிழ் மொழி Syllabic Language ஆகும். ஆங்கில மொழி Spelling Language ஆகும். ஜப்பானிய மொழி Morphinic Language ஆகும். கணிப்பொறியில் ஆங்கில மொழிக்கு லத்தின் வரிவடிவம் அளிக்கப்பட்டுள்ளது. தமிழ் மொழிக்கு தேவனாகிரி வரிவடிவத்தை யுனிகோட்டில் அளிக்கப்பட்டுள்ளது. ஜப்பானிய மொழிக்கு ஹான் (Han), ஹிரான்கானா (Hirangana), கட்டகனா (Katakana) ஆகிய வரிவடிவங்கள் பயன்படுத்தப்படுகின்றன. இவ்வாறு வடிவங்களும் மொழிகளும் இருவேறு பிரிவுகள் என்பது குறிப்பிடத்தக்கது. யுனிகோட் என்ற திட்டம் 1988 ஆம் ஆண்டு துவக்கப்பட்டு, 1991 ஆம் ஆண்டு யுனிகோட் கன்சார்டியமாக மாபெரும் அமைப்பாக நிறுவப்பட்டுள்ளது. இவ்வமைப்பு ஒவ்வொரு நாடாக ஆராய்ச்சியாளர்கள் ஒவ்வொருவருடனும் தொடர்பு கொண்டு இந்தப் பொது மொழித் திட்டத்தில் அனைத்து மொழிகளையும் புகுத்தி 1991 முதல் ஆண்டிற்கு இரு முறை கருத்தரங்கையும் நடத்தி வருகிறது. இந்தாண்டு ஆகஸ்டு 30 முதல் செப்டம்பர் 2- வரையிலான தேதிகளில் அமெரிக்க கலிபோர்னியாவிலுள்ள சான்யுஜாஸ் என்ற நகரில் யுனிகோடின் சர்வதேச அரங்கில் 'யுனிகோட் -3.0' என்ற மக்கள் பயன்படுத்தும்படியான புதிய திட்டம் அறிமுகப்படுத்தப்பட்டுள்ளது. இந்தத் திட்டத்தில் உலக மொழிகள் அனைத்திற்கும் 65,000 இடங்கள் ஒதுக்கப்பட்டுள்ளன. ஆனால் இந்தக் குறியீட்டு இடத்தில் தமிழுக்கு 128 இடங்கள் அளிக்கப்பட்டுள்ளது. இன்னமும் யுனிகோட் திட்டத்தில் 7,800 குறியீடுகளுக்கான இடங்கள் காலியாக உள்ளன. நாம் தமிழ் மொழிக்கு இப்போதே குரல் கொடுத்தால் அதிகக் குறியீட்டு இடங்களை கண்டிப்பாகப் பெறமுடியும்.

குறியீட்டுக்காக அதிக இடங்களை பெறமுடிமோ என்ற சந்தேகம் இருந்தாலும் கவலைப்பட வேண்டியதில்லை. ஏனெனில் யுனிக்கோட்டின் விதிப்படி ஒரு மொழியின் ஒவ்வொரு எழுத்தின் ஒலிக்கும் வேறுபாடுகள் இருக்க வேண்டும். இந்த அடிப்படையில் பார்த்தால் தமிழ்மொழியில் அ, க், கி, கு, கூ என ஒவ்வொரு எழுத்திற்கும் வெவ்வேறு ஒலியே உள்ளது. மேலும் தமிழ் மொழிக்கு அதிக இடம் கேட்டால் மொழி சிதையுமென பலரும் கருதுகிறார்கள். அதிகக் குறியீட்டு இடமென்பது கணிப்பொறியின் உள்செயல்பாட்டுக்கே. ஆகவே சான்யுஜாஸ் மாநாட்டிற்கு முன்பாகவே நமக்கு அதிகக் குறியீட்டு இடங்கள் தேவையென நாம் குரல் எழுப்ப வேண்டிய அவசர நிலையில் உள்ளோம்.

128 இடங்களைத் தான் யுனிகோட் குறியீட்டில் அளித்துள்ளார்களே, சர்வதேச அரங்கில் இதைவிட அதிகமாக தமிழ் குறியீட்டு இடத்தை அளிப்பார்களா என நினைக்க வேண்டாம். ஏனெனில் கொரிய (ஹாங்குல்) மொழிக்கு இக்கணிப்பொறித் திட்டத்தில் 11217 இடங்கள் ஒதுக்கப்பட்டுள்ளன. இணையத் தமிழ் விரும்பிகளும் தங்கள் உரிமைகளை unicode.org என்ற இணைய தலத்திற்கு மின்னஞ்சல் மூலமாக தங்கள் கருத்துகளை அனுப்பலாம். யுனிகோட் திட்டம் என்பது கணிப்பொறிக்கான வருங்காலப் பொதுதிட்டம். இப்போது விட்டால் பிற்காலத்தில் இட ஒதுக்கீடு கேட்க முடியாது. அதிக குறியீட்டு இடம் கிடைத்தால் நம்முடைய தமிழ் வாரிசுகள்தானே அனுபவிக்கப் போகிறார்கள்.

வாழ்க தமிழ்!