UXStrings performances

UXStrings

Unicode Extended Strings utilities in Ada are provided to use strings as simple as it is in other programming languages :

- Unicode characters,
- Dynamic length,
- A comprehensive string convenient subprograms,
- A comprehensive list of strings convenient subprograms,
- And more...

Four experimental implementations

UXStrings 1

UTF-8 encoding is chosen for internal representation. It is faster when storing and getting external UTF-8 strings of text files or web pages. But string manipulations as concatenation are time consuming.

UXStrings 2

In addition to implementation UXStrings 1, some API have been added to support ASCII 7 encoding based on the observation that most text files (especially source codes) and web pages are in English mostly represented by ASCII 7. It is a subset of UTF-8 thus no change with the internal representation. When a string is full ACSII 7, character manipulations are possible with direct index access. It is faster but figuring out if a string is full ASCII 7 is time consuming.

UXStrings 3

Unbounded_Wide_Wide_Strings Ada standard package is chosen for internal representation. It is an optimized Unicode character container despite a time penalty when translating from or to UTF-8.

UXStrings 4

Ada.Containers.Vectors standard generic package is chosen for internal representation. Characters are stored as Wide_Wide_Characters (equivalent to Unicode). This is the only implementation which support character direct indexed string assignment.

Comparaison setup

The performance tests use AdaEdit program using Gnoga V1 and V2 in order to display six different texts on a web page :

- test_ASCII_7.txt: short text with ASCII 7 characters
- test_UTF_8.txt: short text with UTF-8 characters
- 20210414-UXS1-temps.txt: medium size text with ASCII 7 characters
- English.txt: large text with UTF-8 English characters
- Français.txt: large text with UTF-8 French characters
- Emojis.txt: large text with UTF-8 Unicode characters

Gnoga V1 uses Ada standard Latin 1 strings.
Gnoga V2 uses UXStrings implementations 1 to 4.
The measurements which have been made less than a dozen times are not statistical representative.

Comparaison based on Gnoga V1

	Size (k)	Characters	Lines	Gnoga V1	Gnoga V2-UXS1	Gnoga V2-UXS2	Gnoga V2-UXS3	Gnoga V2-UXS4
test_ASCII_7.txt	0,128	125	4	1	0,95	0,79	0,33	0,97
test_UTF_8.txt	0,149	124	4	1	0,44	1,05	0,50	0,50
20210414-UXS1-temps.txt	4	4103	62	1	80,97	12,07	1,10	7,10
English.txt	315	322599	9688	1	105,29	36,20	5,43	33,72
Français.txt	447	434980	3563	1	83,08	56,98	6,95	25,01
Emojis.txt	439	210672	5545	1	64,53	64,94	1,33	6,19

UXStrings 1 is faster or equal on short texts but the penalty of concatenations is a major time factor (60 to 100 times) on large texts.
UXStrings 2 improves on full ASCII text but stay with a high time factor (30 to 70 times) on large texts.
UXStrings 3 is faster on short texts and with an acceptable time factor (1 to 7 times) on large texts.
UXStrings 4 is faster or equal on short texts and the penalty of concatenations is a moderate time factor (6 to 30 times) on large texts.

Comparaison based on Gnoga V2 - UXString 1

	Size (k)	Characters	Lines	Gnoga V2-UXS1	Gnoga V2-UXS2	Gnoga V2-UXS3	Gnoga V2-UXS4
test_ASCII_7.txt	0,128	125	4	1	0,83	0,35	1,02
test_UTF_8.txt	0,149	124	4	1	2,36	1,14	1,12
20210414-UXS1-temps.txt	4	4103	62	1	0,15	0,01	0,09
English.txt	315	322599	9688	1	0,34	0,05	0,32
Français.txt	447	434980	3563	1	0,69	0,08	0,30
Emojis.txt	439	210672	5545	1	1,01	0,02	0,10

UXStrings 2 shows an improvement with full ASCII texts as expected.
UXStrings 3 is really faster by a factor of 10 at least on large files.
UXStrings 4 is faster by a factor of 3 at least on large files.

Comparaison based on Gnoga V2 - UXString 3

	Size (k)	Characters	Lines	Gnoga V2-UXS3	Gnoga V2-UXS4
test_ASCII_7.txt	0,128	125	4	1	2,94
test_UTF_8.txt	0,149	124	4	1	0,99
20210414-UXS1-temps.txt	4	4103	62	1	6,43
English.txt	315	322599	9688	1	6,21
Français.txt	447	434980	3563	1	3,60
Emojis.txt	439	210672	5545	1	4,66

UXStrings 4 is slower by a factor of 4 to 6 on large files.

Conclusion

In terme of performance, UXStrings 3 is a better choice for application like Gnoga and those who want similar performances than Ada standard string packages.
But only UXStrings 4 support character direct indexed string assignment, so it is good opportunity to replace standard Ada string packages.

Pascal Pignard, October 2024.