Encoding ASCII into cf2022 colorForth tokens

EuroForth 2022 Sep 18

Cakes and Biscuits

Flour plus :

Name	Fat	Egg(s)	Raising Agent	Topping
Pancake	0.5	1	0	Maple Syrup
Biscuit	0.5	0	Soda	Chocolate
Cake	1	2	Soda	Icing
Bread	0	0	Yeast	Butter
Jaffa Cake	0.25	0.5	Soda	Chocolate

Cakes and Biscuits

Computer plus :

Name	File	ASCII, Unicode	Raising Agent	Topping
C Source File	1	1, UTF-8	Pre-processor	Libraries
Word Docx	1	1, *	Templates	Styles
Forth Source	1	1, UTF-8	Forth	Source Database
PDF file	1	1, UTF-8	Forth-like	Zip, Signing
colorForth	0.01	0.01, UTF-8	Forth	Chocolate

Layers

Layers

Choose your ingredients, add toppings.

Layers

Win32Forth	Native Forth	colorForth	
Application	Application	Application	
File System	File System	Blocks	
OS	OS	-	
Hardware	Hardware	Hardware	

Encoding ASCII into cf2022 colorForth tokens   EuroForth 2022 Sep 18

Layers

Win32Forth	Native Forth	colorForth	JaffaCake Forth
Application	Application	Application	Application
File System	File System	Blocks	Blocks
OS	OS	-	OS/VM
Hardware	Hardware	Hardware	Hardware

Encoding ASCII into cf2022 colorForth tokens   EuroForth 2022 Sep 18

```
; ASCII / UTF8 support. If the first Shannon-Fano encoded letter is a 4 bit NULL,
; display the next 24 bits as three ASCII characters.
; $03e3c009 is displayed as '><'
```

; ASCII / UTF8 support. If the first Shannon-Fano encoded letter is a 4 bit NULL,
; display the next 24 bits as three ASCII characters.
; $0<mark>3e3c</mark>009 is displayed as '<mark>><</mark>'

Encoding ASCII into cf2022 colorForth tokens   EuroForth 2022 Sep 18

; ASCII / UTF8 support. If the first Shannon-Fano encoded letter is a 4 bit NULL,
; display the next 24 bits as three ASCII characters.
; $0==3e3c==009 is displayed as '><'

Encoding ASCII into cf2022 colorForth tokens   EuroForth 2022 Sep 18

; ASCII / UTF8 support. If the first Shannon-Fano encoded letter is a 4 bit NULL,
; display the next 24 bits as three ASCII characters.
; $0`3e3c`009 is displayed as '><'

Encoding ASCII into cf2022 colorForth tokens   EuroForth 2022 Sep 18

```
showShannonFano:       ; ( token -- ) \ display the Shannon-Fano encoded token on TOS
    ; ASCII / UTF8 support. If the first Shannon-Fano encoded letter is a 4 bit NULL,
    ; display the next 24 bits as three ASCII characters.
    mov _SCRATCH_, _TOS_              ; save the token value
    and _SCRATCH_, 0xF0000000
    cmp _SCRATCH_, 0x00000000
    jnz .forward
        ; display as three ASCII characters
        mov _SCRATCH_, _TOS_

        mov _TOS_, _SCRATCH_
        shr _TOS_, 20
        and _TOS_, 0x000000FF
        jz .null_terminator
            _DUP_
            call emit_

        mov _TOS_, _SCRATCH_
        shr _TOS_, 12
        and _TOS_, 0x000000FF
        jz .null_terminator
            _DUP_
            call emit_

        mov _TOS_, _SCRATCH_
        shr _TOS_, 4
        and _TOS_, 0x000000FF
        jz .null_terminator
            _DUP_
            call emit_

        ; arrive here if an ASCII character is an ASCII NULL, or if all three have been emitted
        .null_terminator:
        call space_                   ; display a space character at the end of the word
        _DROP_
        ret
```

Encoding ASCII into cf2022 colorForth tokens   EuroForth 2022 Sep 18

```
lowercase:    ; display a white text word in normal lower-case letters
    call white
showSF_EDI_: ; ( -- ) \ display a Shanon-Fano encoded token pointed to by  edi  in the current colour
    _DUP_
    mov _TOS_, [ ( edi * 4 ) - 0x04 ]    ; fetch the next token – drops through to showShannonFano

call showShannonFano

    .forward:

    ; display as Shannon-Fano encoded token name
    and _TOS_, byte -0x10    ; and _TOS_, 0xFFFFFFF0 ignore token colour when displaying the letters

lowercasePrimitive: ; ( token -- ) \ display the given Shanon-Fano encoded word in the current colour
    call unpack
    jz lowercasePrimitiveEnd
    call emitSF_
    jmp lowercasePrimitive
lowercasePrimitiveEnd:
    call space_
    _DROP_
    _DROP_
    ret
```

This simple change to the colorForth pre-parsed source format, and the corresponding editor display words, allows the colorForth system to continue to use 32 bit „colored" tokens, and also support ASCII / UTF-8 characters.

This simple change to the colorForth pre-parsed source format, and the corresponding editor display words, allows the colorForth system to continue to use 32 bit „colored" tokens, and also support ASCII / UTF-8 characters.

ToDo: Add keyboard support to type ASCII / UTF-8 characters !

This simple change to the colorForth pre-parsed source format, and the corresponding editor display words, allows the colorForth system to continue to use 32 bit „colored" tokens, and also support ASCII / UTF-8 characters.

ToDo: Add keyboard support to type ASCII / UTF-8 characters !

Questions?