Subject: Re: A terminal emulator widget for U++
Posted by Oblivion on Wed, 01 Jul 2026 04:39:32 GMT
View Forum Message <> Reply to Message

Hi,

There are many changes, new features and improvements under the hood TerminalCtrl.
I would like to share one of them. This is an optimization.
I've been working hard to further optimize TerminalCtrl's ANSI sequence -lexical- parser (VTInStream), and I think I achieved some remarkable results.
I have vectorized the parser, meaning TerminalCtrl will use SIMD instructions by default.
Final test results are as follows:

Workload	Scalar	SIMD	Gain
ASCII (Ground State)	1384.18 MB/s	4621.07 MB/s	3.34x
CSI (Parameters)	307.05 MB/s	326.28 MB/s	1.06x
OSC (String Payloads)	398.01 MB/s	694.23 MB/s	1.74x
DCS (Passthrough)	319.27 MB/s	360.63 MB/s	1.13x
Randomized / mixed workloads			
Random ASCII	1092.11 MB/s	1906.23 MB/s	1.75x
Random CSI	725.91 MB/s	810.19 MB/s	1.12x
Random OSC	496.36 MB/s	1443.07 MB/s	2.91x
Random DCS	572.19 MB/s	767.82 MB/s	1.34x
Mixed (shell-like)	336.07 MB/s	383.30 MB/s	1.14x
Pathological / edge	872.20 MB/s	2791.99 MB/s	3.20x
4MB Long Randomized Payloads (Speed Benchmarks)			
4MB Random ASCII	1056.48 MB/s	1904.80 MB/s	1.80x
4MB Random CSI	708.55 MB/s	815.38 MB/s	1.15x
4MB Random OSC	494.32 MB/s	1489.28 MB/s	3.01x
4MB Random DCS	571.40 MB/s	755.17 MB/s	1.32x
4MB Mixed (shell-like)	339.28 MB/s	382.84 MB/s	1.13x
4MB Pathological / edge	861.76 MB/s	2511.11 MB/s	2.91x
4MB Raw byte fuzz	105.56 MB/s	100.55 MB/s	0.95x
64MB Single Huge Payload (Passthrough / Extracted)			
64MB Single Huge OSC	650.02 MB/s	3030.16 MB/s	4.66x

64MB Single Huge DCS | 1141.43 MB/s | 3067.49 MB/s | 2.69x

Diagnostic: scalar vs SIMD, full buffer (no chunking at all)

[parity] Mixed (shell-like) | 0/100 full-buffer trials diverged (scalar vs SIMD)
[parity] Random CSI | 0/100 full-buffer trials diverged (scalar vs SIMD)
[parity] Random OSC | 0/100 full-buffer trials diverged (scalar vs SIMD)
[parity] Random DCS | 0/100 full-buffer trials diverged (scalar vs SIMD)
[parity] Pathological / edge | 0/100 full-buffer trials diverged (scalar vs SIMD)
[parity] Raw byte fuzz | 0/100 full-buffer trials diverged (scalar vs SIMD)
[parity] C1 DCS/ST probe | 0/100 full-buffer trials diverged (scalar vs SIMD)

Diagnostic: same engine, whole buffer vs. randomly-chunked feed
---
[scalar] Pathological / edge | 0/100 trials diverged (whole buffer vs. randomly-chunked feed)
[scalar] Raw byte fuzz | 0/100 trials diverged (whole buffer vs. randomly-chunked feed)
[SIMD] Pathological / edge | 0/100 trials diverged (whole buffer vs. randomly-chunked feed)
[SIMD] Raw byte fuzz | 0/100 trials diverged (whole buffer vs. randomly-chunked feed)

Diagnostic: 1MB Long Randomized Payloads (Stress / Parity)

[parity] 1MB Mixed (shell-like) | 0/10 full-buffer trials diverged (scalar vs SIMD)
[parity] 1MB Pathological / edge | 0/10 full-buffer trials diverged (scalar vs SIMD)
[parity] 1MB Raw byte fuzz | 0/10 full-buffer trials diverged (scalar vs SIMD)
[scalar] 1MB Pathological | 0/10 trials diverged (whole buffer vs. randomly-chunked feed)
[SIMD] 1MB Pathological | 0/10 trials diverged (whole buffer vs. randomly-chunked feed)

Diagnostic: minimal repro - first byte offset where a 2-way split breaks it
---
[scalar] Pathological / edge | no single 2-way split reproduced it - needs 3+ chunks
[SIMD] Pathological / edge | no single 2-way split reproduced it - needs 3+ chunks
[scalar] Raw byte fuzz | no single 2-way split reproduced it - needs 3+ chunks
[SIMD] Raw byte fuzz | no single 2-way split reproduced it - needs 3+ chunks
---

Machine Specs: AMD Ryzen 5600g, 32GB DDR4, Linux 7.0, GNOME 50 / Wayland

As you can see there are real gains for plain ASCII parsing and some passthrough operations (OSC/DCS/APC),yet have behavior parity. :)
The optimized parser will be merged within this month.

Note that this optimization is using SSE/NEON api of U++. However, it can be easily modified to use AVX/AVX2 and their counterpart on ARM machies. But that is a future plan.

Best regards,
Oblivion