

```
// farolito: [Spanish, paper lantern, diminutive of faro, lantern, from faro, lighthouse, lantern, from Latin pharus, from Pharus Pharos.]
// Stride 8 = 4x2, HalfStride = (2*sizeof(uint16_t))
uint32_t FNV1A_farolito(const char *str, uint32_t wrdlen)
{
    const uint32_t PRIME = 591798841;
    uint32_t hash32 = 2166136261;
    uint32_t hash32b = 2166136261;
    uint32_t hash32c = 2166136261;
    uint32_t hash32d = 2166136261;
    const char *p = str;
    uint32_t Loop_Counter;
    uint32_t Second_Line_Offset;

    if (wrdlen >= 2*(2*sizeof(uint16_t))) {
        Loop_Counter = (wrdlen>>2);
        Loop_Counter++;
        Second_Line_Offset = wrdlen-(Loop_Counter)*(2*sizeof(uint16_t));
        for(; Loop_Counter; Loop_Counter--, p += (2*sizeof(uint16_t))) {
            hash32 = (hash32 ^ *(uint16_t*)(p+0)) * PRIME;
            hash32b = (hash32b ^ *(uint16_t*)(p+0+Second_Line_Offset)) * PRIME;
            hash32c = (hash32c ^ *(uint16_t*)(p+0+2)) * PRIME;
            hash32d = (hash32d ^ *(uint16_t*)(p+0+Second_Line_Offset+2)) * PRIME;
        }
        hash32 = _rot_KAZE((hash32 ^ _rot_KAZE(hash32b,16)) * PRIME,24);
        hash32 = _rot_KAZE((hash32 ^ _rot_KAZE(hash32c,16)) * PRIME,16);
        hash32 = _rot_KAZE((hash32 ^ _rot_KAZE(hash32d,16)) * PRIME,8);
    } else {
        // Cases: 0,1,2,3,4,5,6,7:
        if (wrdlen & 2*sizeof(uint16_t)) {
            hash32 = (hash32 ^ *(p+0)) * PRIME;
            hash32 = (hash32 ^ *(p+1)) * PRIME;
            hash32 = (hash32 ^ *(p+2)) * PRIME;
            hash32 = (hash32 ^ *(p+3)) * PRIME;
            p += 2*sizeof(uint16_t);
        }
        if (wrdlen & sizeof(uint16_t)) {
            hash32 = (hash32 ^ *(p+0)) * PRIME;
            hash32 = (hash32 ^ *(p+1)) * PRIME;
            p += sizeof(uint16_t);
        }
        if (wrdlen & 1) {
            hash32 = (hash32 ^ *(p+0)) * PRIME;
        }
    }
    return hash32 ^ (hash32 >> 16);
}

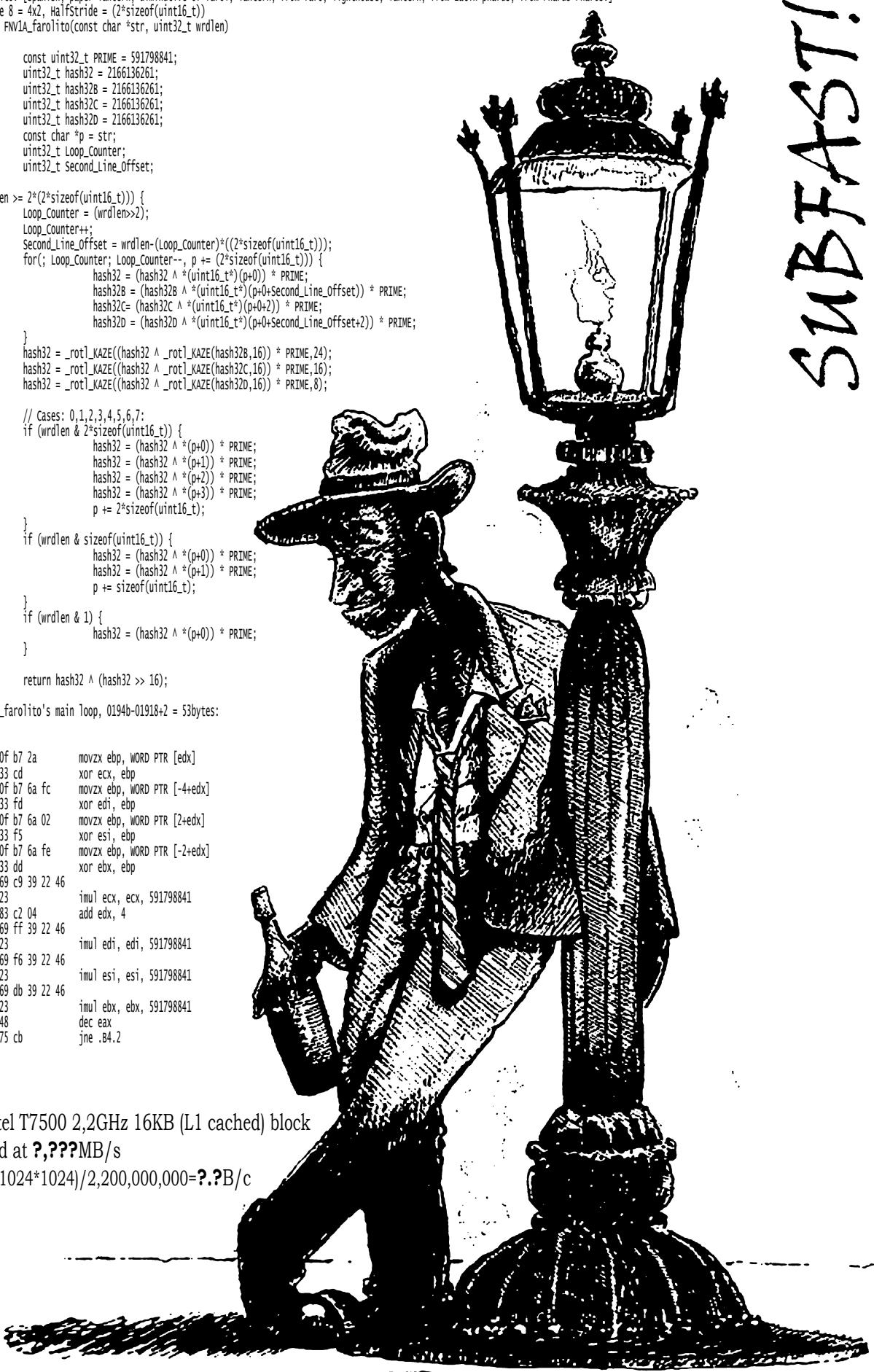
// FNV1A_farolito's main loop, 0194b-01918+2 = 53bytes:
/*
.B4.2:
01918 0f b7 2a    movzx ebp, WORD PTR [edx]
0191b 33 cd        xor ecx, ebp
0191d 0f b7 6a fc  movzx ebp, WORD PTR [-4+edx]
01921 33 fd        xor edi, ebp
01923 0f b7 6a 02  movzx ebp, WORD PTR [2+edx]
01927 33 f5        xor esi, ebp
01929 0f b7 6a fe  movzx ebp, WORD PTR [-2+edx]
0192d 33 dd        xor ebx, ebp
0192f 69 c9 39 22 46
                   23    imul ecx, ecx, 591798841
01935 83 c2 04        add edx, 4
01938 69 ff 39 22 46
                   23    imul edi, edi, 591798841
0193e 69 f6 39 22 46
                   23    imul esi, esi, 591798841
01944 69 db 39 22 46
                   23    imul ebx, ebx, 591798841
0194a 48          dec eax
0194b 75 cb        jne .B4.2
.B4.3:
*/

```

On Intel T7500 2,2GHz 16KB (L1 cached) block
hashed at ????MB/s

(????*1024*1024)/2,200,000,000=??.B/c

Simplifier / Dumbdownificator



FNV1A_farolito – a nifty text hasher

3