

Microprocessors

Exercise 1

Jonas Sticha, Thorsten Fuchs, Florian Reichhold

Approaches

- C
- Lookup Table
- ASM
- Intrinsics

Lookup Table

- Constant

```
const char lut[] = {65,66,67,68,69,70,71,72,73,74,75,76,77,78,79,80,81,82,83,84,85,86,87,88,89,90,91,92,93,94,95,96,97,98,99,100,101,102,103,104,105,106,107,108,109,110,111,112,113,114,115,116,117,118,119,120,121,122,123,124,125,126,127,128,129,130,131,132,133,134,135,136,137,138,139,140,141,142,143,144,145,146,147,148,149,150,151,152,153,154,155,156,157,158,159,160,161,162,163,164,165,166,167,168,169,170,171,172,173,174,175,176,177,178,179,180,181,182,183,184,185,186,187,188,189,190,191,192,193,194,195,196,197,198,199,200,201,202,203,204,205,206,207,208,209,210,211,212,213,214,215,216,217,218,219,220,221,222,223,224,225,226,227,228,229,230,231,232,233,234,235,236,237,238,239,240,241,242,243,244,245,246,247,248,249,250,251,252,253,254,255};
```

- Correction of the Characters

```
for(; *text != '\0'; text++) {  
    *text = lut[*text-65];  
}
```

- Dynamic

```
static char* buildLookupTable() {  
  
    char* table = (char*)malloc(58);  
  
    for(int i = 65; i <= 90; i++)  
        table[i-65] = i;  
    for(int i = 97; i <= 122; i++)  
        table[i-65] = i-0x20;  
  
    return table;  
}
```

ASM

```
static void toupper_asm(char * text) {  
    __asm__ __volatile__ (  
        "movq $0, %%rsi\n\t"           // init rsi with 0 (offset of address)  
        "loop:\n\t"  
        "movb 0(%%rbx, %%rsi, 1), %%al\n\t" // load value at rbx + (rsi * 1) - 0  
        "cmp $0, %%al\n\t"             // check if string end is reached  
        "je end\n\t"  
        "inc %%rsi\n\t"                 // increment offset  
        "cmp $0x5A, %%al\n\t"           // compare letter case  
        "jle loop\n\t"  
        "sub $0x20, %%al\n\t"           // make large case  
        "movb %%al, -1(%%rbx, %%rsi, 1)\n\t" // store value  
        "jmp loop\n\t"  
        "end:\n\t"  
        :                             /* no output registers */  
        : "b" (text)                  /* input registers */  
        : "al", "rsi"                 /* clobbered registers */  
    );  
}
```

Intrinsics - MMX

```
static void toupper_mmx(char * text) {  
    __m64 simddataOld;  
    __m64 simddataNew;  
    __m64 comparator;  
    __m64 compresult;  
    __m64 subtractor;  
    __m64 ones;  
  
    comparator = _mm_set1_pi8(0x5A);  
    subtractor = _mm_set1_pi8(0x20);  
    ones = _mm_set1_pi8(0xFF);  
  
    unsigned int textlen = strlen(text);  
  
    unsigned int iterations = textlen / 8;
```

```
    for(int i = 0; i < iterations; i++)  
    {  
        simddataOld = *(__m64*)text;  
        compresult = _mm_cmpgt_pi8(simddataOld, comparator);  
        simddataNew = _mm_sub_pi8(simddataOld, subtractor);  
        simddataNew = _mm_and_si64(simddataNew, compresult);  
        compresult = _m_pxor(compresult, ones);  
        simddataOld = _mm_and_si64(simddataOld, compresult);  
        simddataNew = _mm_add_pi8(simddataNew, simddataOld);  
        *(__m64*)text = simddataNew;  
        text += 8;  
    }  
  
    toupper_lookup(text);
```

Intrinsics - SSE

```
for(int i = 0; i < iterations; i++)
{
    simddataOld = _mm_load_si128((void*)text);
    compresult = _mm_cmpgt_epi8(simddataOld, comparator);
    simddataNew = _mm_sub_epi8(simddataOld, subtractor);
    simddataNew = _mm_blendv_epi8(simddataOld, simddataNew, compresult);
    _mm_store_si128((void*)text, simddataNew);
    text += 16;
}

toupper_lookup(text);
```

Intrinsics - AVX2

```
for(int i = 0; i < iterations; i++)
{
    simddataOld = _mm256_load_si256((void*)text);
    compresult = _mm256_cmpgt_epi8(simddataOld, comparator);
    simddataNew = _mm256_sub_epi8(simddataOld, subtractor);
    simddataNew = _mm256_blendv_epi8(simddataOld, simddataNew, compresult);
    _mm256_store_si256((void*)text, simddataNew);
    text += 32;
}

toupper_lookup(text);
```

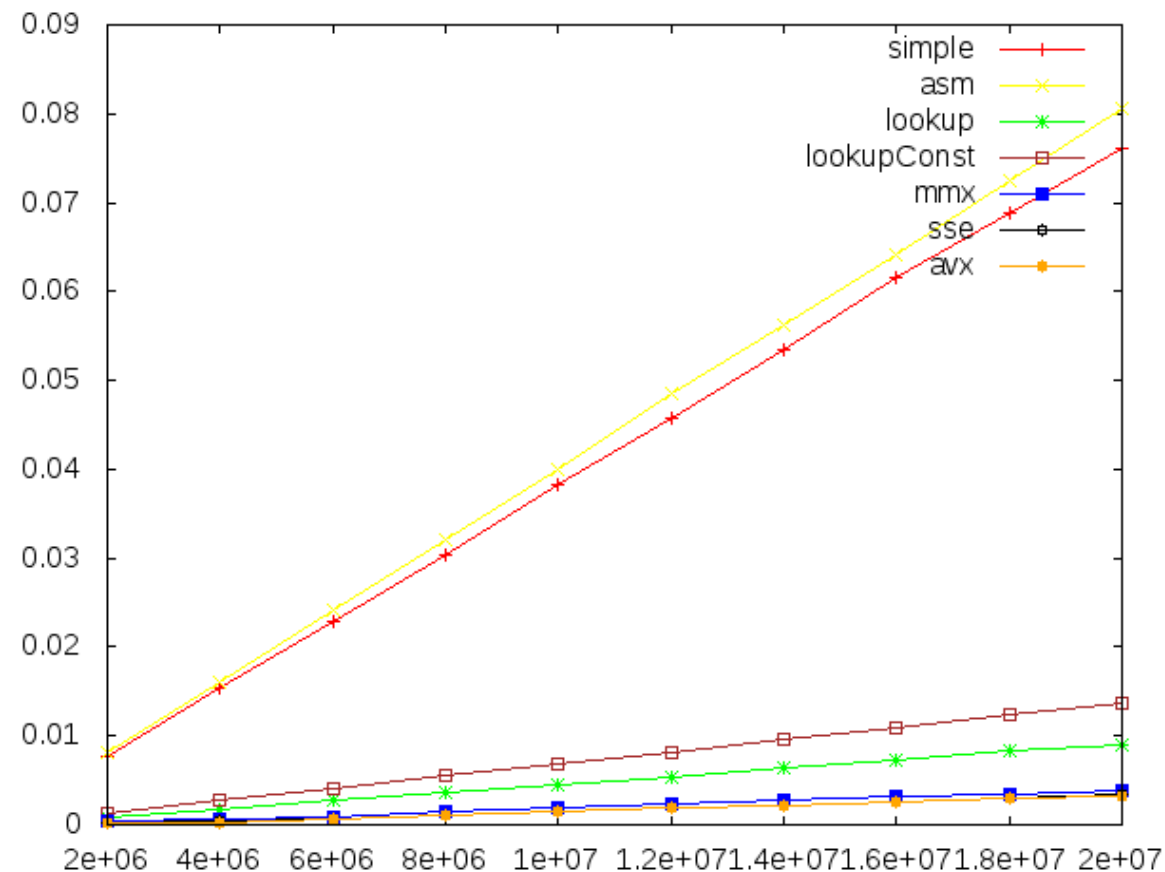
```
static void toupper_avx(char * text) {
    __m256i simddataOld;
    __m256i simddataNew;
    __m256i comparator;
    __m256i compresult;
    __m256i subtractor;

    comparator = _mm256_set1_epi8(0x5A);
    subtractor = _mm256_set1_epi8(0x20);

    unsigned int textlen = strlen(text);

    unsigned int iterations = textlen / 32;
```

- No Optimisation (-O0)



Speedup Comparison

-o0	2 mio	10 mio	20 mio
simple	0.012293	0.061624	0.123405
asm	0.007894	0.039501	0.079033
lookupConst	0.004766	0.023870	0.047762
mmx	0.003004	0.015570	0.031069
sse	0.001071	0.005786	0.011582
avx	0.000598	0.003400	0.006728
speedup (simple/avx)	~ 20x	~ 18x	~ 18x

-o3	2mio	10mio	20mio
simple	0.007703	0.038173	0.076197
asm	0.008024	0.040065	0.080496
lookupConst	0.001360	0.006793	0.013692
mmx	0.000380	0.001887	0.003872
sse	0.000292	0.001580	0.003373
avx	0.000142	0.001590	0.003293
speedup (simple/avx)	~ 54x	~ 24x	~ 23x