

## Listing

```
for (int i=0; i< N; ++i)
{
    a[i] = 2.0 * a[i];
}
```

Implementierung Neon:

```
inti;
// Vector Loop
for (i=0; (i<N-3) && (N&~3); i+=4)
{
    float32x4_t vr= vld1q_f32(&a[i]);
    vr= vmulq_n_f32(vr, 2.0);
    vst1q_f32 (&a[i], vr)
}
// Drain loop
for (; i < N; ++i)
    a[i] = 2.0 * a[i];
```

Implementierung SVE:

```
For (int i = 0 ; i < N; i += svcntw() )
{
    svbool_t Pg= svwhilelt_b32(i, N);
    svfloat32_t vr= svld1(Pg, &a[i]);
    vr= svmul_x(Pg, vr, 2.0);
    svst1(Pg, &a[i], vr);
}
```

Führt zu folgendem Assembler-Code:

```
cmp w0, #1
```

```

    b.lt .Ende
// i = 0
    mov w8, wzr
.Loop:
// Pg= svwhilelt_b32(i, N);
    whilelt p0,w8,w0
    sxtw x9, w8
// vr= svld1(Pg, &a[i]);
    ld1w {z0.s}, p0/z, [x1,x9,ls! #2]
// i += svcntw()
    incw x8
// i < N
    cmp w8, w0
// vr= svmul_x(Pg, vr, 2.0);
    fmul z0.s, p0/m, z0.s, #2.0
// svst1(Pg, &a[i], vr);
    st1w {z0.s}, p0, [x1,x9,ls! #2]
// i < N
    b.lt .Loop
.Ende
    ret

```

Listing: Vektorisierung einer skalaren Schleife mit Neon und SVE.