Optimizing The Idle Loop

Pointless work. Yes, you're making progress, but it won't make any difference to the result.
Is there any situation where it might actually be useful (beyond a make work project)?
It's not difficult to imagine a hard real-time problem where this could apply.

In OperatingSystem programming, the IdleProcess? is used to occupy the CPU in those times when *no* other process is awake. In many systems, it is nothing more than a tight loop around a halt or wait instruction:

```
idle:
    hlt
    jmp short idle
```

When the scheduler queues the idle process, it immediately halts the CPU until an interrupt occurs, at which point the CPU wakes and the scheduler takes over processing from the IdleProcess?. When the IdleProcess? is next queued, it loops back to the halt instruction, waiting for the next interrupt to come. -- JayOsako

---

If the idle loop is using considerable resources (due to a BusyWaiting implementation), optimizing it will free resources to the other threads of the program (and other programs in Windows).
-- AurelianoCalvo

When I was first learning to program (in Atari BASIC, over 20 years ago - I was still in grade school at the time), I quickly learned that one could slow down a program which was too fast (an occurrence that was seldom in Atari BASIC - an interpreted BASIC running on a 1.79Mhz, 8-bit processor) with a delay loop, i.e.

```
10 FOR I = 1 TO 100
 20 NEXT I
```

(I apologize if I got that wrong; it's been decades).
One day, while writing a program, I discovered that it ran too slowly for my taste. Being the GrandMasterProgrammer that I was (heh heh heh), I tried the obvious solution -- inserting a sped-up delay loop, as follows:

```
10 FOR I = 1 TO 1
20 NEXT I
```

Was quite chagrined to find that this subterfuge did not work... soon, it did dawn on me why. :)
-- ScottJohnson

A lot of EmbeddedSystems lack sophisticated timers and/or interrupt facilities. For those, optimizing the idle loop helps the SpinLock become more accurate.
Battery-constrained devices such as GameBoyAdvance? often have several different forms of being idle, from BusyWaiting on down to a DeepSleep? state just above being off. Optimizing the IdleLoop? in this case would refer to choosing the idle state that preserves enough functionality to get your program to run while drawing the least current.
-- DamianYerrick?