Abgabe zur 3. Praktikumsaufgabe im Fach „Embedded Computing“

erstellt von

Markus Schmidt  
Maximilian Gaul

Funktion waste\_msecs:

Unsere Funktion addiert die Summe von 0 bis zu einer Obergrenze.

Die Anzahl der Schleifendurchläufe setzt sich aus einem Faktor, der zu wartenden Millisekunden-Anzahl *msecs* und einer Konstante, in unserem Fall *1.000.000*, zusammen:

**void** **waste\_msecs**(**unsigned** **int** msecs) {

**unsigned** **long** runs = (**unsigned** **long**)(1000000 \* fac \* msecs);

**unsigned** **long** sum = 0;

**unsigned** **long** i = 0;

**for**(i = 0; i < runs; i++) {

sum += i;

}

}

Zu Beginn ist der Faktor fac auf 1.0 gesetzt. Um ihn für zukünftige Messungen richtig einzustellen, führen wir waste\_msecs einmal in der Erwartung aus, 100ms zu verbrauchen.  
Der Faktor ergibt sich dann aus der Division der 100ms und der tatsächlich gewarteten Zeit in ms:

difference\_in\_seconds = end\_time.tv\_sec - start\_time.tv\_sec;

difference\_in\_ns = end\_time.tv\_nsec - start\_time.tv\_nsec;

time = (difference\_in\_seconds \* 1000 \* 1000 \* 1000 + difference\_in\_ns) / (1000.0);

fac = (**double**)msec / (**double**)(time / 1000.0);

Bei der Ausgangspriorität von *10* beträgt der Faktor *0.09058*.  
Danach führen wir eine Reihe von Messungen für 1ms aus, die auch ziemlich genau erreicht werden:

Main thread priority is 10

Setting fac to 0.090580

Elapsed time 1104000us

Elapsed time 1000us

Elapsed time 1000us  
  
Anschließend führen wir das Setzen des Faktors und die Messungen für 1ms erneut auf einem Thread auf, der mit einer sehr hohen Priorität von *253* läuft, aus. Dort liegt der Faktor bei *0.090744*. Der leicht erhöhte Faktor ergibt sich daraus, dass der Ablauf des Programms schneller ist, somit müssen mehr Schleifendurchgänge durchlaufen werden, um auf die gleiche Wartezeit zu kommen.  
Auch die nachfolgenden 1ms Wartezeiten werden sehr genau eingehalten:

--- THREAD TALKING ---

My priority is 253

Setting fac to 0.090744

Elapsed time 1102000us

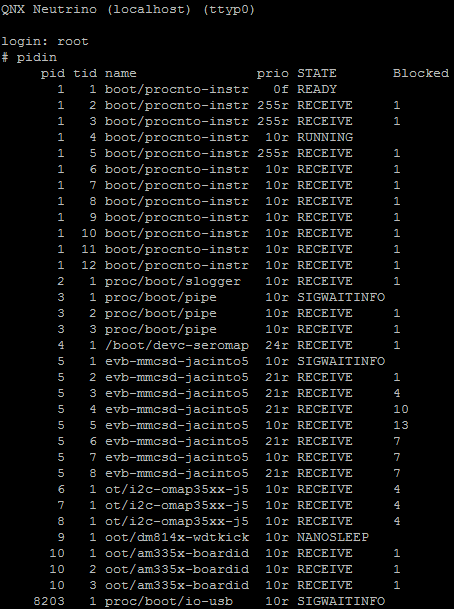
Elapsed time 1000us

Elapsed time 1000us

Telnet-Verbindung:

Wir loggen uns über PuTTY auf die IP *192.168.8.2* via Telnet über den login *root* ein.

Anschließend erhalten wir die Übersicht aller laufenden Prozesse und deren Priorität über den Kommandozeilenbefehl *pidin*:



Einstellen der Priorität eines Threads

Zuerst schalten wir die Vererbung der Thread-Parameter aus:

pthread\_attr\_setinheritsched(&thread\_attr,PTHREAD\_EXPLICIT\_SCHED);

Anschließend verändern wir nur den Parameter sched\_priority auf *253*:

pthread\_attr\_getschedparam(&thread\_attr, &thread\_sched);

und schreiben ihn wieder zurück:

pthread\_attr\_setschedparam(&thread\_attr, &thread\_sched);