Abgabe zur 2. Praktikumsaufgabe im Fach „Embedded Computing“

erstellt von

Markus Schmidt  
Maximilian Gaul

Einstellen der Auflösung der Systemuhr

Die Auflösung der Systemuhr lässt sich über das Setzen des Attributes *nsec* der Struktur *\_clockperiod* und anschließendes Übergeben an *ClockPeriod* einstellen:

**Code:** *struct \_clockperiod set\_period;  
set\_period.nsec = microsecs \* 1000; /\* Setzen auf übergebenen Parameter der Funktion \*/  
set\_period.fract = 0;  
ClockPeriod(CLOCK\_REALTIME, &set\_period, NULL, 0);*

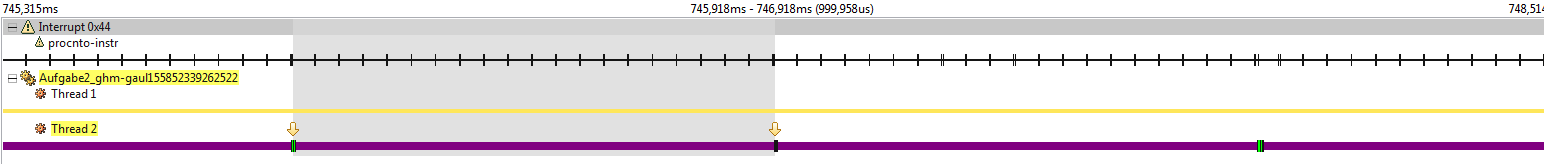
Auslesen der Auflösung wird wie folgt realisiert:

**Code:** *struct \_clockperiod get\_period;*  
*ClockPeriod(CLOCK\_REALTIME, NULL, &get\_period, 0);*

Unseren Messungen zufolge ist die Systemuhr beim Starten auf *1000us* eingestellt. Die Auflösung von 1ms reicht aus, um den Takt einer Millisekunde zu realisieren.

Nachweis der korrekten Takterzeugung von 1ms:

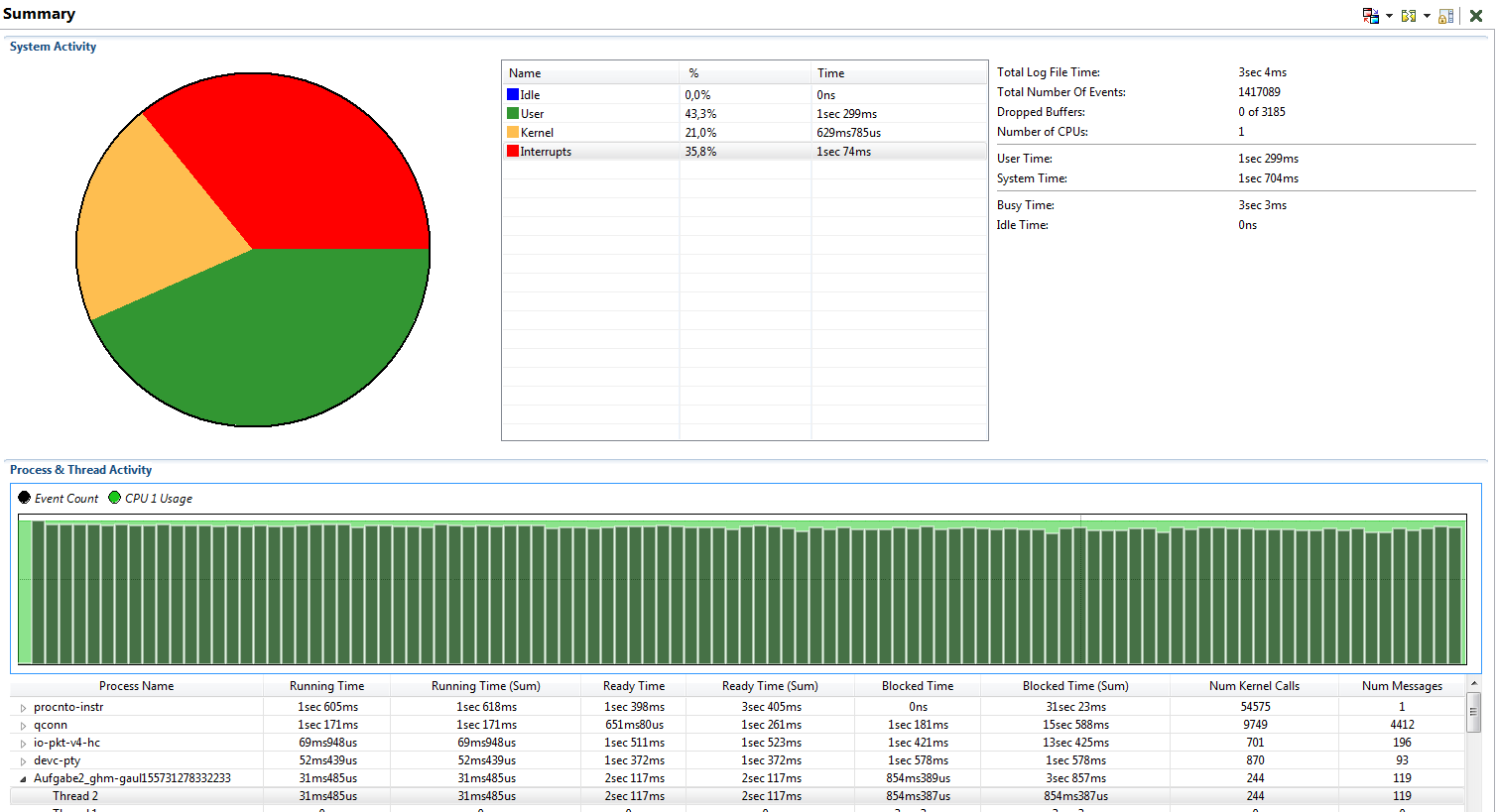
DieQNX-DIE erlaubt im Kernel-Events-Tracer die Interrupts nachzuverfolgen.



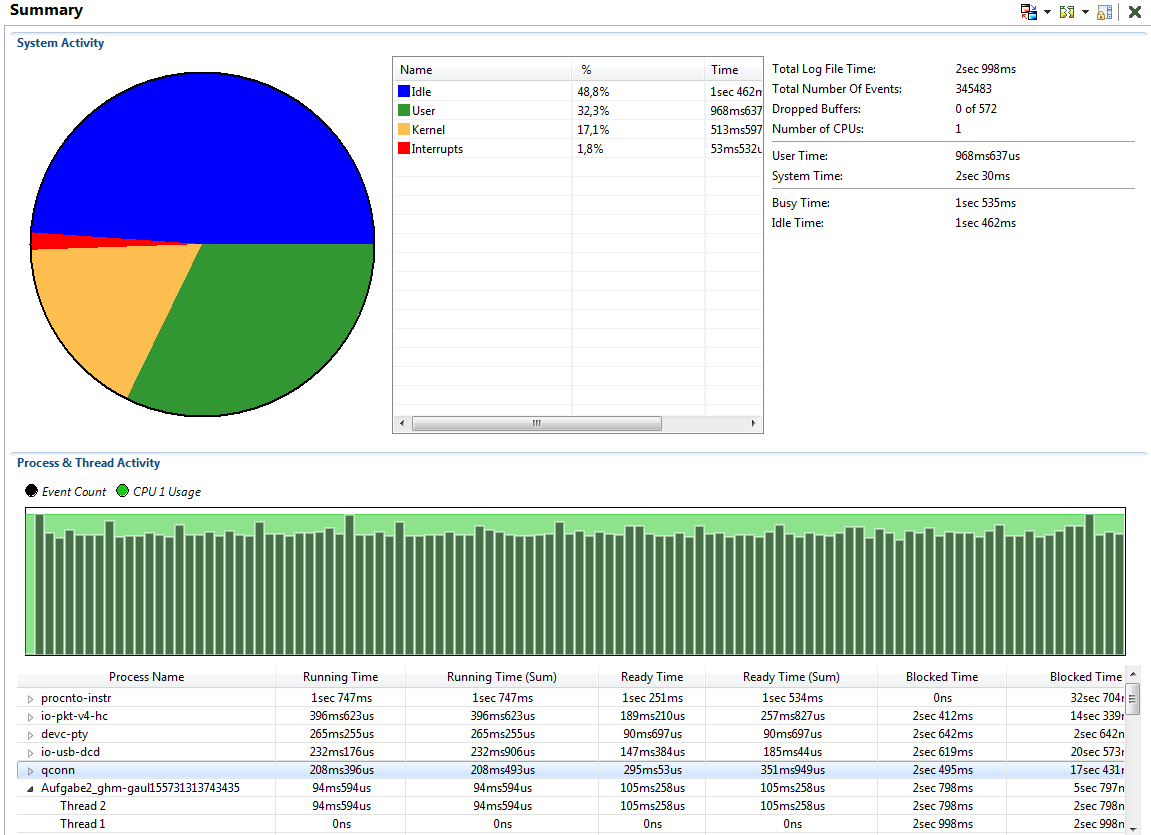
Der Abstand zwischen zwei Aufrufen ist ziemlich genau 1ms.

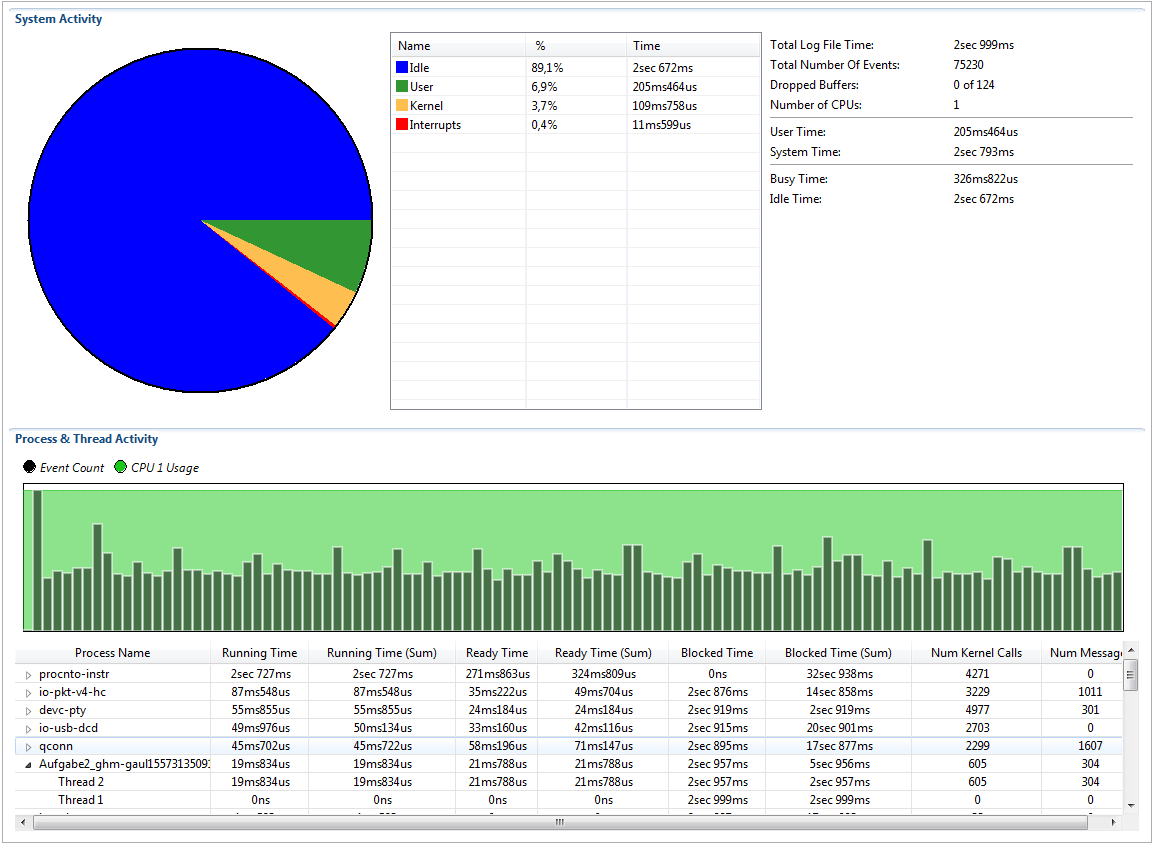
Auswirkungen auf die Systemlast

Je höher die Auflösung der Systemuhr desto höher ist die Systemlast.  
Ein auf *10us* eingestellter Systemtakt erzeugt folgende Messungen des *Kernel Events Tracer*



Wie zu erkennen ist, führt ein Systemtakt von *10us* zur kompletten Auslastung des Systems (*Idle =* *0.0%*) da dadurch sehr häufig Interrupts erzeugt werden, die natürlich alle behandelt werden müssen (ggf. Änderung auf einen Prozess mit höherer Priorität + Kontextwechsel) = *35.8%*.

Wie auf den folgenden Abbildungen zu sehen reduziert sich die Systemauslastung mit Verringerung des Systemtakts (1. Bild = 1000us, 2. Bild = 5000us) systematisch: 



Je niedriger der Systemtakt desto ungenauer ist die Wartezeit aus *a)*.  
Mit *100us* Systemtakt wird die Wartezeit von *1ms* noch ausreichend genau erreicht, gleichzeitig ist die Systemlast nicht zu stark (noch ausreichend Idle-Zeit verfügbar):  
