Add an interface to query resolution of time bases
Jens Gustedt

To cite this version:

HAL Id: hal-02378605
https://inria.hal.science/hal-02378605
Submitted on 25 Nov 2019

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
Add an interface to query resolution of time bases v3
Proposal for C2x

Jens Gustedt
INRIA and ICube, Université de Strasbourg, France

We propose the inclusion of a query function for time resolution that is modelled after ISO 9945’s clock_getres.

History: This is one part of a follow-up of N2402 and N2417, which had been denied adequate treatment in the Ithaca 2019 meeting of WG14.

1. INTRODUCTION

The interfaces in time.h to manipulate time values have grown mostly unattended over the years and present several problems that could be easily avoided with more modern, redesigned interfaces. This paper is concerned with the following problem:

— The function timespec_get has a resolution for which there is no query interface.

2. STRATEGY

C17 has no interface that would allow to query the resolution of the TIME.UTC time. Because of the genericity of timespec_get, the interface to query resolutions should not be a series of macros:

— User functions may have a time base as a parameter, so they cannot decide at compile time which resolution would be to query.
— The resolution may not be part of the platform ABI. E.g. it may be dependent of a particular version of the CPU or operating system.
— The resolution for a specific time base should not change during program execution. Therefore performance critical code can easily cache these values at program startup or thread startup if they need to.

ISO 9945 has a function that is capable to capture resolutions of predefined bases and also of all implementation-defined bases, the clock_getres function. This function has the following properties:

— The resolution of a given time base is dynamic and not guaranteed to be the same for all executions. It is only guaranteed to be stable per POSIX process.
— The resolution is returned via a pointer parameter to a timespec. This allows to model time bases with a resolution that exceeds the second.
— The pointer parameter is allowed to be null. This allows to use that function just as a query interface for the existence of the corresponding time base.

As the C17 function timespec_get has been modeled after clock_gettime, we propose to model such a function, timespec_getres, accordingly after clock_getres.

3. IMPLEMENTATION EXPERIENCE

The clock_getres function is present on all systems that conform to ISO 9945:2009 or newer. For POSIX versions before that it had been present if the “times” option had been supported.

To accommodate the API changes that C11 had inflicted to clock_gettime we just propose a change in the API that goes along the same lines.

© 2019 by the author(s). Distributed under a Creative Commons Attribution 4.0 International License
4. PROPOSED WORDING
Add a new clause to 7.27.2 as follows:

**7.27.2.X The timespec_getres function**

**Synopsis**

```c
#include <time.h>
int timespec_getres(struct timespec *ts, int base);
```

**Description**

If ts is non-null and base is supported by the timespec_get function, the timespec_getres function returns the resolution of the time provided by the timespec_get function for base in the timespec structure pointed to by ts. For each supported base, multiple calls to clock_getres during the same program execution shall have identical results.

**Returns**

If the value base is supported by the timespec_get function, the timespec_getres function returns the nonzero value base; otherwise, it returns zero.