

Suspender la ejecución de un programa durante milisegundos y/o microsegundos

Escrito por adrianvaca
Miércoles, 23 de Marzo de 2011 00:02 -

Si estamos en Windows y queremos suspender la ejecución de un programa durante cierto número de milisegundos, usaremos la función Sleep definida en windows.h, por ejemplo:

```
Sleep ( 5000 ); // Du
```

Ahora si queremos suspender la ejecución durante cierto número de microsegundos, nos encontramos con la sorpresa de que en Windows no hay una función que nos permita lograrlo (en Unix/Linux tenemos usleep), al menos de forma directa, tenemos que contruir nuestra propia función...

La solución se basa en usar contadores de precisión para poder medir el tiempo, en concreto las funciones QueryPerformanceFrequency y QueryPerformanceCounter:

Primero la cabecera:

```
#include <windows.h>
```

Ahora contruimos una clase que nos permita reutilizar el código:

```
// class CDuracion
{
protected:
    LARGE_INTEGER m_Inicio;
    LARGE_INTEGER m_Fin;

    LONGLONG m_Frecuencia;
    LONGLONG m_Correccion;
```

Suspender la ejecución de un programa durante milisegundos y/o microsegundos

Escrito por adrianvaca

Miércoles, 23 de Marzo de 2011 00:02 -

public:

```
    CDuracion(void);

    void Iniciar(void);
    void Detener(void);
    double DevolverDuracion(void) const;
};
```

```
inline CDuracion::CDuracion(void)
```

```
{
    LARGE_INTEGER Frecuencia;
```

```
    QueryPerformanceFrequency(&Frecuencia);
    m_Frecuencia = Frecuencia.QuadPart;
```

```
    Iniciar();
    Detener();
```

```
    m_Correccion = m_Fin.QuadPart-m_Inicio.QuadPart;
}
```

```
inline void CDuracion::Iniciar(void)
```

```
{
    // Asegura que no sera interrumpido por otro proceso durante cierto tiempo
    Sleep(0);
    QueryPerformanceCounter(&m_Inicio);
}
```

```
inline void CDuracion::Detener(void)
```

```
{
    QueryPerformanceCounter(&m_Fin);
}
```

```
inline double CDuracion::DevolverDuracion(void) const
```

```
{
    return (double)(m_Fin.QuadPart-m_Inicio.QuadPart-m_Correccion)*1000000.0 / m_Frecuencia;
}
```

Ahora podemos construir la siguiente función, que nos permite suspender la ejecución durante cierto número de microsegundos:

Suspender la ejecución de un programa durante milisegundos y/o microsegundos

Escrito por adrianvaca

Miércoles, 23 de Marzo de 2011 00:02 -

```
void uSleep          (          double intervalo          )
{
    CDuracion timer    ;

    timer              .          Iniciar          ();

    while( timer        .          DevolverDuracion      () <
    {
        timer          .          Detener          ();
    }
}
```

Y podemos usarla así:

```
uSleep          (          5000000          );          // Du
```

Recordar que:

1 segundo = 1000 milisegundos

1 segundo = 1000000 microsegundos

Otro uso de la clase puede ser hallar el tiempo en microsegundos que tardó en ejecutarse cierto código, por ejemplo:

```
CDuracion timer    ;

timer              .          Iniciar          ();

// Aqui va el código a ejecutar...
Foo                ();          // Funcion a llamar
```

Suspender la ejecución de un programa durante milisegundos y/o microsegundos

Escrito por adrianvaca

Miércoles, 23 de Marzo de 2011 00:02 -

timer	.	Detener	();	
cout	<<	"La funcion tardo: "	<<	timer