International
Carpathian Control
Conference ICCC' 2002
MALENOVICE,
CZECH REPUBLIC
May 27-30, 2002

REAL-TIME APPLICATIONS IN WINDOWS NT

Vladimír VAŠEK, Michal BLIŽŇÁK and Karel PERŮTKA
Institute of Information Technologies
Tomas Bata University in Zlin, Czech Republic

Abstract: For the education in the area of control theory there was built the program system for the interactive work with the real discrete control loop. The program is running in the real time conditions and allows define both the structure and the parameters of the discrete controller, which is connected with the real controlled system. In the first was necessary to solve the real time program running. For this problem was done the analysis of the windows possibilities in this area. Finally the multimedia application timing was used for this problem solving.

Key words: Windows NT, real-time, multimedia timer, hard and soft real time system

1 Introduction

Operating systems with real-time support are generally systems, which allow us simultaneous parallel processing of several tasks, programs, processes. It means that they support technologies as multitasking, multithreading, multi-processors tasks processing, exact timing and so on. MS DOS is the typical example of the operating system without the support of the tasks processing in real-time. This system is based as a system, in which is able to process only one process in the given time. Processing of many processes in real-time is able only with many special, often very complex, modifications and operating system upgrades, for example RTMON for MS DOS and C, C++. MS Windows, Unix are some of the operating systems which enable to use multiprocessing. However, these operating systems have some drawbacks, limitations, which inhibit us to create adequate applications which use tasks processing in real-time. Analysis of these limitations in the operating system MS Windows has been done. It was focused on finding as simple solution as possible of the problems connected with the realization of real-time applications in this system. Operating system MS Windows is generally system, which enables us the independent processing of several applications or processes simultaneously. Multiprocessing (multithreading) is supported by 32bit versions of this system. It includes timers with theoretical value of the period down to 1ms and other elements which are

necessary for creating applications working in real-time. Their quality and possibilities of usage vary very much. It depends on the version the used operating system. Acceptable quality is received at the operating systems based on the platform Win NT and higher.

2 Real time in the conditions of MS Windows

Operating system MS WINDOWS is a system based on the processing of messages. It means that arbitrary system event (input –output event, system requirements, etc.) is a direct effect, outgrowth or cause of the processing of the *system event*. These events are sorted to the events queue a processed by the kernel of the system in accordance with some given rules, as is type and priority of the event. In means it practice that it is not always guaranteed immediate processing of the request, or we do not know, when the event will be processed. By the way, it means that computer time of the processor is not assigned under the exact instants of time to the process, but under the state of the queue of the processes.

OS Windows determines, which events, messages are connected with the processing program and function *GetMessage* is finished, when it is necessary to process the message. When no messages are sent, the processing program is paused and other programs can run. When message arrived, paused program is awaken. Function *TranslateMessage* converts messages, which have included ASCII signs and the function *DispatchMessage* gives the control to the "pump" of the messages. When the service function is finished, the control is given to the MFC code, which, finally, stopped of the *DispatchMessage* function.

2.1 System timers

Service procedures of counters are not also activated on the request for pausing, but the system, again, send a message about reaching the time interval. From this it concludes that system timers cannot be used for exact time control. Moreover, messages from timers are not asynchronous. It means, when a message from the timer appears at the time when the previous timing message was not serviced because of the operating system overcharge, the previous message will be serviced never more. It is because Windows do not save messages from timers to queues, but sets sign about the presence of this message. So we are loosing a summary of the whole number of the incoming messages from timer and hence about time.

Another problem connected with usage of times is a fact that, theoretically, it is possible to program them on generating the messages with the interval down to 1 ms, but, however, it is impossible in practice. The basic frequency of the hardware timer on which the timers of the Windows 9x system are based is in the most workstations 54,9 ms with the tolerance of several microseconds. All other adjustable frequencies are multiples of this 55ms.

It means the fact when we set the timer on the value 1ms, our request for the operation will be generated after the time 55ms after the counter started. It will be the same for the value 50ms. If we set the value of the timer interval to 60ms, timer will send messages every 110ms. Moreover, the messages will not have to be handled immediately, but it is possible that they will be delayed, mainly depended on the active processes of the system. This delay is usually tens of milliseconds, but, however, in some cases it could be a few seconds. The situation is almost the same in the operating system **Windows NT/2000** but

with the slightly difference, the lowest possible period, which could be set, is **10ms** and all other periods are its multiple. The table shown bellow includes a list of the real intervals among the receiving of the each of the messages by WM_TIMER, the 32-bit application of the system Windows 9x. It was programmed for the purpose to send messages every 500ms by the timer:

Table 1. Intervals among the receiving the message by the WM_TIMER application

Hlášení č.	Interval [s]	Hlášení č.	Interval [s]
1	0,542	11	0,604
2	0,557	12	0,550
3	0,541	13	0,549
4	0,503	14	0,549
5	0,549	15	0,550
6	0,549	16	0,508
7	1,936	17	0,550
8	0,261	18	0,549
9	0,550	19	0,549
10	0,549	20	0,550

It is evident from the table that the average time among the arrival of the each of the messages is about the value 550ms and the most of the real intervals is very close to this value. The only one disturbing element is the interval 1.936 between the 6th and 7th message of the WM_TIMER. It appeared because of dragging and dropping the window to the different position on the screen. It is clear from this that Windows do not allow us to form a mob in the queue of the messages. If this had not been true, the application would have received three or four fast messages in sequence by the timer that have had been processed very fast for catching-up the delay of 1.936s.

For easier analysis of the abilities of the timers in dependence on the occupancy of the system and the value of the interval, the application **SysTimer** was created. It was designed for the experimental verification of the fact mentioned above.

Graphs in figure 1 were measured with using the application **SysTimer** and they are showing the time differences, which mean the differences between the set period of the timer and the real response of the system. Simulation was going in accordance with this procedure. In the first half of the experiment, the application was kept in abeyance, while in the second half Windows were strained through starting of several applications, minimizing and maximizing the windows and so on. The influence of timers is clear from the graphs, mainly during the system overloading. Moreover, in difference to the timers in the systems Win9x, they do not have permanent time error because of the bad chosen minimal time resolution of the timers. It concludes from the facts mentioned above that it is much more difficult to have timely response of the application on the external impulse in Windows.

It is the same as for the event of the external character as for the handling of the internal message of the system. And this is the main reason why the system MS Windows is not so suitable for creating the real-time applications working simultaneously with several tasks. They need exact running, mainly during the control of the processes, which dynamic changes could be very fast.

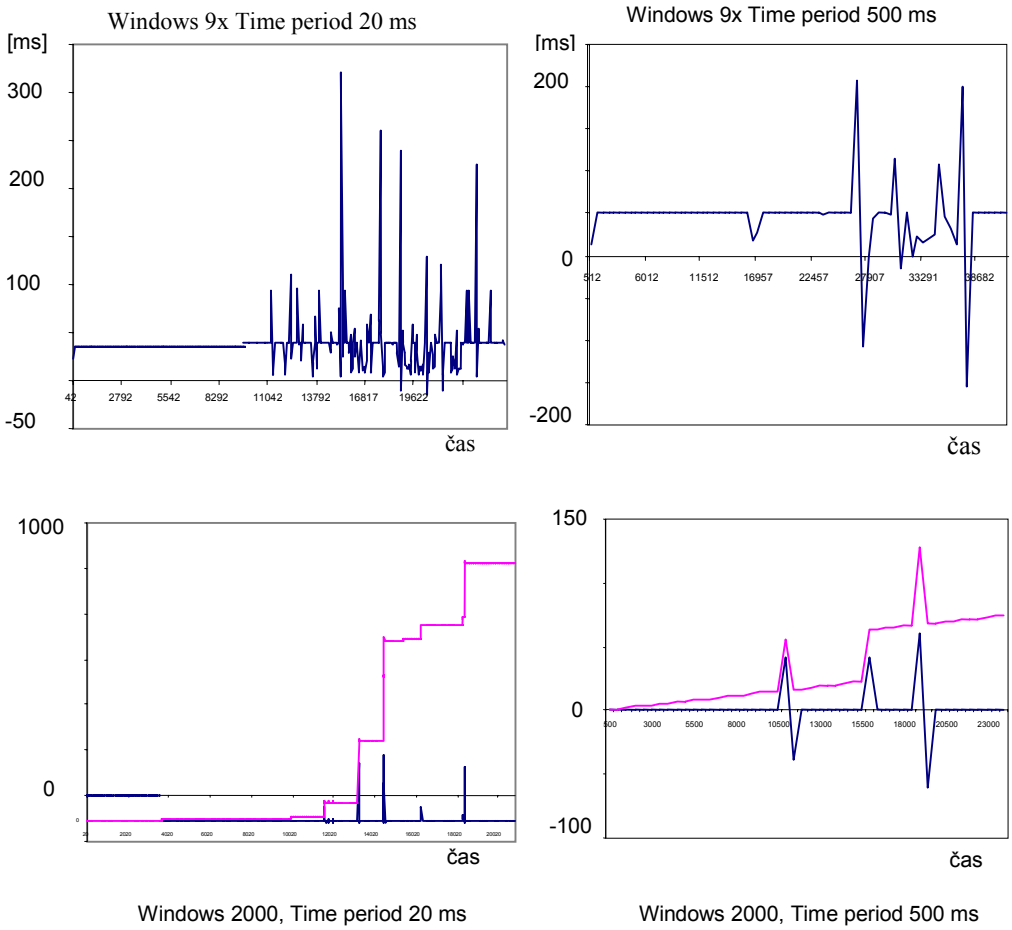

Figure 1 The errors of the multimedia timer in MS Windows 9x and 2000

From the characteristics of the real-time applications, in principle, we distinguish two types of systems:

Hard Real-Time – this type has the highest quality, but it is the most work-intensive, too. It is based on the usage of the hardware. With using the hardware it is possible to generate the hardware breaking, pausing used for the deduction of the exact time base. It would take creating new virtual drivers which would control exact and immediate processing and handling the pieces of information from the technological acquisitions

cards, which is not so easy in Windows NT. There is another question, whether the results would compensate the money and effort of this project.
Soft Real-Time – in case we exclude the usage of the system timers of MS Windows because of the facts given above, we use the choice, which combines both techniques mentioned above. It is based on the usage of timers, so called multimedia timers. The usage of these timers guarantees very exact timing and, through small problems with programming, is quite good way of programming the real-time applications.

2.2 Usage of multimedia timer

These timers are timers used by Windows for exact data processing. They are mainly used for creating the multimedia applications, such as during the timing in the MIDI sequencers. These timers have guaranteed the accuracy down to 1ms and handling the events is not through the messages of Windows, but as a handling of pausing.

The higher requirements on the system resources, the dependence on the hardware of the whole PC and the problems during programming are their drawbacks. As is clear from the graphs, the usage of multimedia timers really solves the problem of setting the exact period of timing.

Multimedia timers permit the small immediate deviation from the chosen period, however, the whole time deviation, error is minimized (in accordance of the timer settings) down to 1ms. The different techniques of programming in MS Windows 9x and MS Windows NT/2000 are their disadvantage.

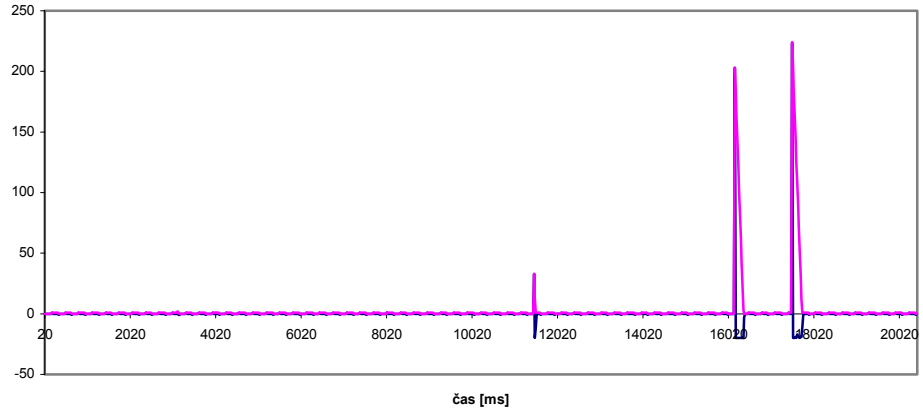

Figure 2. The errors of the multimedia timers in MS Windows 2000, 20 ms

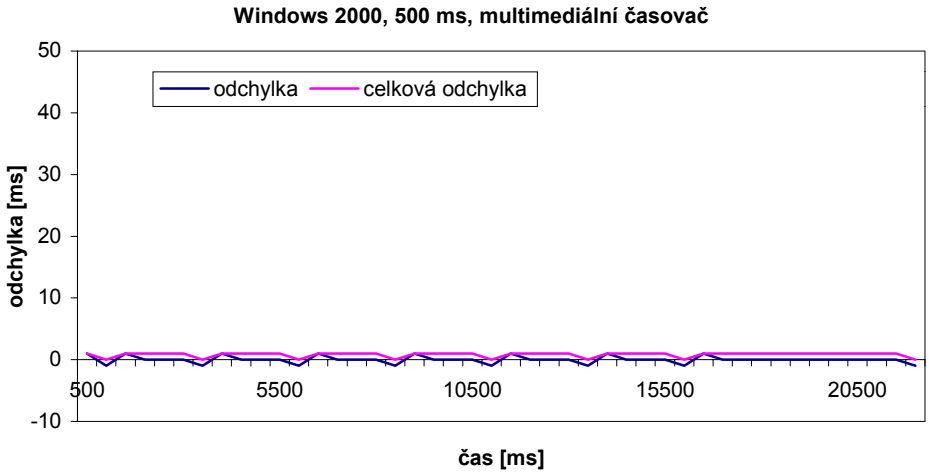

Figure 3. The errors of the multimedia timers in MS Windows 2000

Conclusion

An analysis of the possibilities of the realization of the real-time applications in the operating system MS Windows has been done. It has been found from the problem analysis that it is good to use the system timers as a source of the exact timing for the running of the user programs. For the sufficient accuracy, the multimedia timers were used. They satisfy the time requests of the standard systems for monitoring and control of the technological processes. Based on this solution, the program WCONTROL was created. [Bližňák] It realizes a computer controller. This controller is possible to implement to the real control circuit, moreover, it is possible to choose different types of controllers and easily set their parameters. The choice of the structure Program system is mainly for the educational purposes and laboratory experiments.

References

BLIŽŇÁK, M. 2001. *Real-timové aplikace řídicích a monitorovacích systémů pod OS Windows*. (Diploma work), FT UTB Zlín, 2001.

PROSISE, Jeff 2000. *Programování ve Windows pomocí MFC*., Computer Press 2000, Vydání první .

KRUGLINSKI, D.,J. 1999. *Mistrovství ve Visual C++*. : Computer Press 1999.

ANDREWS, M. 1996. *Programujeme v jazyce C++*, Computer Press 1997.

Microsoft Developer Network Library: Internet, part of Microsoft Developer Studio.

Acknowledgement*:* This work was supported by the Ministry of Education of the Czech Republic in the range of research project No. MSM 281100001 and project No. FRVŠ MŠMT G1/1503/2000.