DOKŁADNY POMIAR CZASU W WINDOWS

Do mierzenia czasu w Windows proponuję używać funkcji

```
BOOL QueryPerformanceCounter(
  __out  LARGE_INTEGER *lpPerformanceCount
);
```

Return Value

BOOL

If the function succeeds, the return value is nonzero.

If the function fails, the return value is zero. To get extended error information, call GetLastError.

Remarks

On a multiprocessor computer, it should not matter which processor is called. However, you can get different results on different processors due to bugs in the basic input/output system (BIOS) or the hardware abstraction layer (HAL). To specify processor affinity for a thread, use the SetThreadAffinityMask function.

```
LARGE_INTEGER startTimer()
{
   LARGE_INTEGER start;
   DWORD_PTR oldmask = SetThreadAffinityMask(GetCurrentThread(), 0);
   QueryPerformanceCounter(&start);
   SetThreadAffinityMask(GetCurrentThread(), oldmask);
   return start;
}
LARGE_INTEGER endTimer()
{
   LARGE_INTEGER stop;
   DWORD_PTR oldmask = SetThreadAffinityMask(GetCurrentThread(), 0);
   QueryPerformanceCounter(&stop);
   SetThreadAffinityMask(GetCurrentThread(), oldmask);
   return stop;
}
```

użycie w programie

```
LARGE_INTEGER performanceCountStart,performanceCountEnd;
performanceCountStart = startTimer(); //zapamiętujemy czas początkowy
display(tab,6);  //tutaj funkcje, których mierzymy wydajność
performanceCountEnd = endTimer(); //zapamiętujemy koniec czasu
double tm = performanceCountEnd.QuadPart - performanceCountStart.QuadPart;
cout << endl << "Time:" <<tm <<endl;
```

BARDZO KRÓTKA INSTRUKCJA DEBUGOWANIA PROGRAMU

W VISUAL STUDIO 2008

Debugowanie

Celem debugowania jest śledzenie wykonywania całego (lub fragmentu) programu linia po linii. Podczas debugowania można podglądać wartości zmiennych i je modyfikować. Aby to było możliwe programu musi być uruchomiony w trybie Debug. W tym trybie program wykonuje się do napotkania pierwszej pułapki i zatrzymuje się na linii w której jest ona ustawiona.

Ustawianie pułapek

Pułapkę można ustawić więcej niż jedną. Pułapkę ustawia się dwoma sposobami: myszą lub z pozycji menu. Aby ustawić pułapkę z pozycji menu należy wykonać: Menu->Debug->Toggle breakpoint. Z pozycji myszy należy najechać kursorem myszy na pionowy margines z lewej strony na wysokości linii na której chcemy założyć pułapkę i naciąć lewy klawisz myszy. W obu przypadkach powinno pojawić się czerwone kółko w tym miejscu. Usuwanie pułapki odbywa się dokładnie w ten sam sposób. Jeśli w tej linii była pułapka, to zostanie usunięta, jeśli nie, to zostanie założona. Pułapki można zakładać i usuwać w trakcie debugowania programu.

Uruchomianie programu w trybie debug

Realizuje się to przez naciśnięcie odpowiedniego klawisza: z menu -> *Debug->Start debug (F5)* lub naciśnięcie odpowiedniej ikony na pasku. Uruchomienie tego trybu powoduje uruchomienie programu i wykonywanie go aż do napotkania pułapki. Po dojściu do pułapki program zatrzymuje się i może być wykonywany linia po linii, mogą być obserwowane i zmieniane zmienne, etc.

Śledzenie programu krok po kroku (linia po linii)

Po dojściu do pułapki można wykonywać program linia po linii zam pomoc dwóch klawiszy: F10 F11. Rożnica między nimi polega na tym, że gdy w podświetlonej linii jest funkcja to naciśnięcie F11 spowoduje wejście w tą funkcję i wykonywanie poszczególnych instrukcji funkcji linia po linii, natomiast F10 potraktuje funkcję jako pojedynczą instrukcję i ją wykona. Gdy nie chcemy dalej śledzić programu linia po linii to naciskamy z menu -> *Debug->Continue (F5)*

<pozycja *Start debugging* zmieniła nazwę na *Continue*> lub naciskamy ikonkę na pasku. Uruchomienie tej opcji spowoduje wykonywanie się programu do następnej pułapki. Jeśli chcemy przerwać działanie programu to menu->*Debug->Stop debugging*

Wniosek! Ponieważ śledzenie jest linia po linii należy zatem przyjąć zasadę pisania programu: jedna instrukcja w jednej linii.

Śledzenie wartości zmiennych

W trybie pracy krokowej można śledzić i zmieniać wartość zmiennych. Najprostszy sposób śledzenia polega na najechaniu myszą na zmienną i chwilę potrzymaniu jej nieruchomo - powinno pojawić się okienko wyświetlające wartość zmiennej. Dodatkowo na dole ekranu jest okno z zakładkami: *Autos, Threads, Modules, Watch1*. W zakładkach *Autos* i *Watch1* można podejrzeć wartość aktualnych zmiennych. W zakładce *Watch1* można dodać zmienne do śledzenia (należy dwukrotnie nacisnąć myszą na ostatnią pustą linię w jej pierwszą kolumnę).

Szybki podgląd zmiennych można uruchomić z menu: *Debug->QuickWatch*

Zmiana wartości zmiennej w trakcie debugowania

Można zmienić wartość zmiennej (także pojedynczy element tablicy). Wystarczy w zakładce *Autos* lub *Watch1* klikną lewym przyciskiem myszy na zmienną. Pojawi się wówczas menu z którego należy wybrać *Edit Value*. Zmianę wartości można również przeprowadzić za pomocą opcji *Debug->QuickWatch*