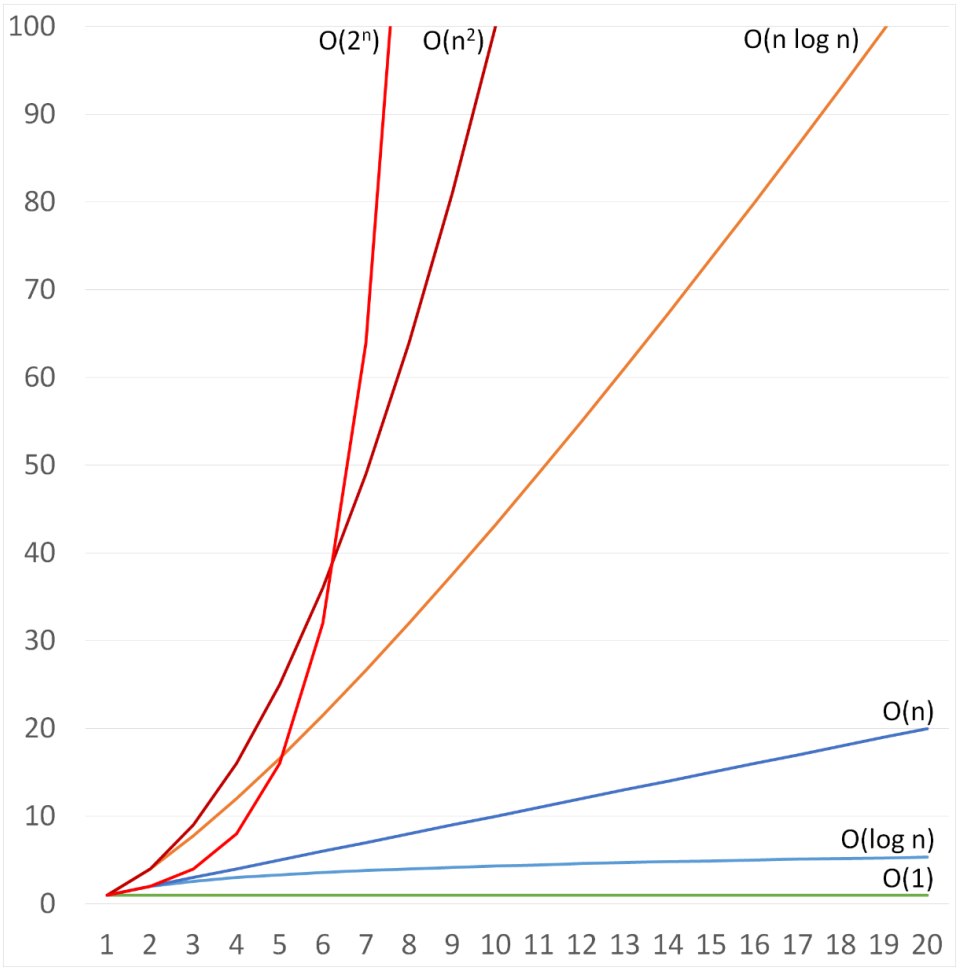
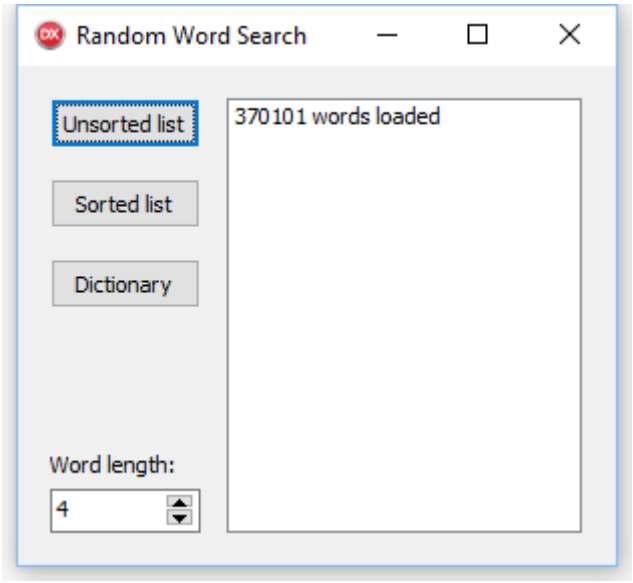
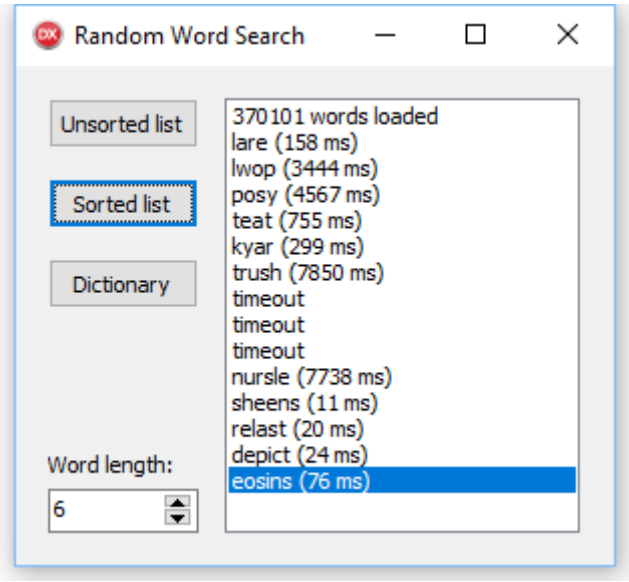
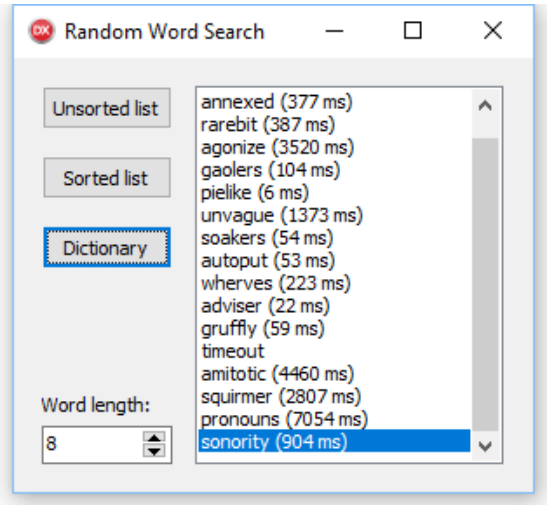
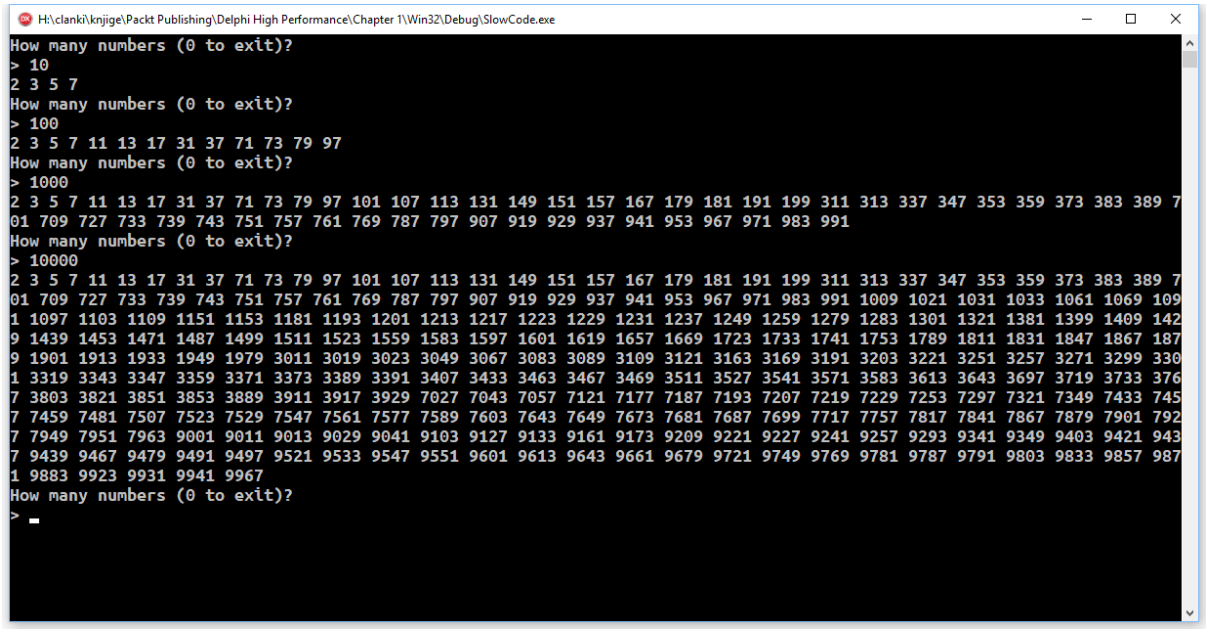
Chapter 1: About Performance



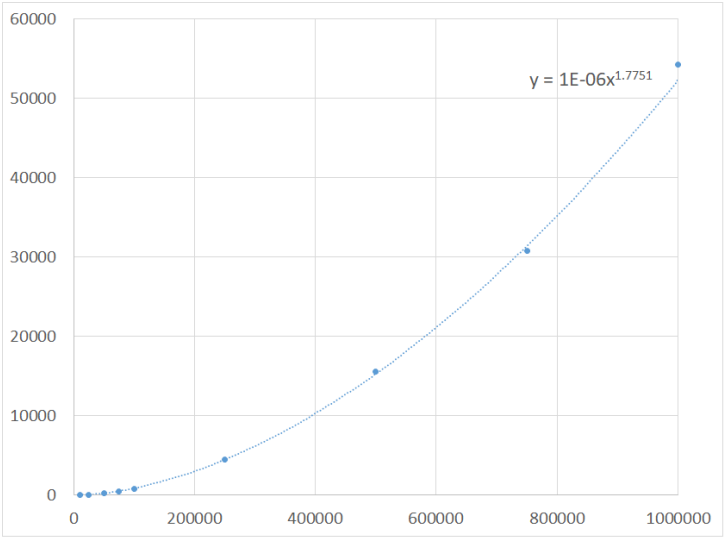


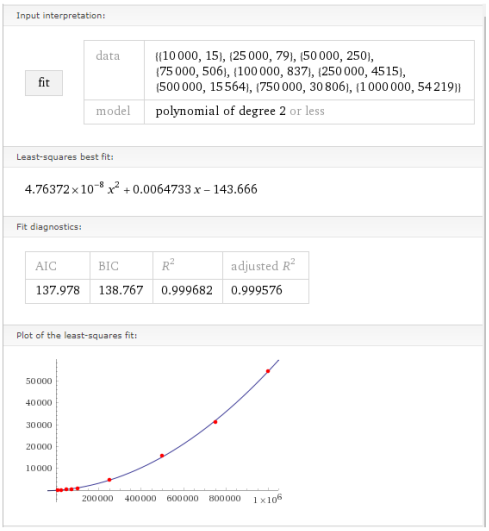

Random Word Search
Unsorted list
Sorted list
Dictionary
Word length:
6
370101 words loaded
lare (158 ms)
lwop (3444 ms)
posy (4567 ms)
teat (755 ms)
kyar (299 ms)
trush (7850 ms)
timeout
timeout
timeout
nursle (7738 ms)
sheens (11 ms)
relast (20 ms)
depict (24 ms)
eosins (76 ms)

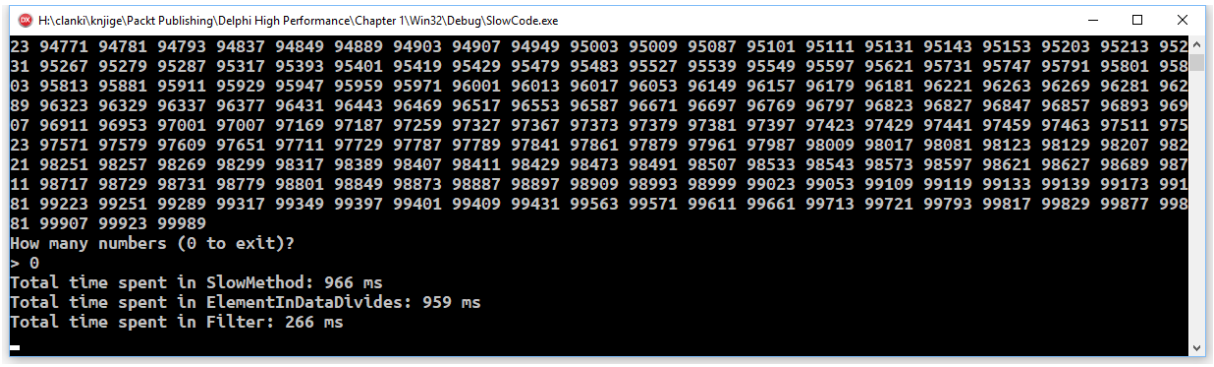

Random Word Search
Unsorted list
Sorted list
Dictionary
Word length:
8
annexed (377 ms)
rarebit (387 ms)
agonize (3520 ms)
gaolers (104 ms)
pielike (6 ms)
unvague (1373 ms)
soakers (54 ms)
autoput (53 ms)
wherves (223 ms)
adviser (22 ms)
gruffly (59 ms)
timeout
amitotic (4460 ms)
squirmer (2807 ms)
pronouns (7054 ms)
sonority (904 ms)

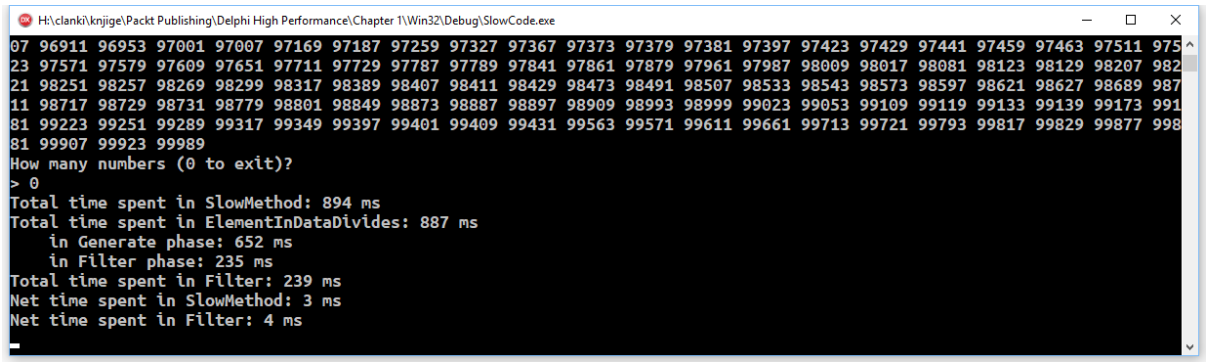

H:\clanki\knjige\Packt Publishing\Delphi High Performance\Chapter 1\Win32\Debug\SlowCode.exe
How many numbers (0 to exit)?
> 10
2 3 5 7
How many numbers (0 to exit)?
> 100
2 3 5 7 11 13 17 31 37 71 73 79 97
How many numbers (0 to exit)?
> 1000
2 3 5 7 11 13 17 31 37 71 73 79 97 101 107 113 131 149 151 157 167 179 181 191 199 311 313 337 347 353 359 373 383 389 7
01 709 727 733 739 743 751 757 761 769 787 797 907 919 929 937 941 953 967 971 983 991
How many numbers (0 to exit)?
> 10000
2 3 5 7 11 13 17 31 37 71 73 79 97 101 107 113 131 149 151 157 167 179 181 191 199 311 313 337 347 353 359 373 383 389 7
01 709 727 733 739 743 751 757 761 769 787 797 907 919 929 937 941 953 967 971 983 991 1009 1021 1031 1033 1061 1069 109
1 1097 1103 1109 1151 1153 1181 1193 1201 1213 1217 1223 1229 1231 1237 1249 1259 1279 1283 1301 1321 1381 1399 1409 142
9 1439 1453 1471 1487 1499 1511 1523 1559 1583 1597 1601 1619 1657 1669 1723 1733 1741 1753 1789 1811 1831 1847 1867 187
9 1901 1913 1933 1949 1979 3011 3019 3023 3049 3067 3083 3089 3109 3121 3163 3169 3191 3203 3221 3251 3257 3271 3299 330
1 3319 3343 3347 3359 3371 3373 3389 3391 3407 3433 3463 3467 3469 3511 3527 3541 3571 3583 3613 3643 3697 3719 3733 376
7 3803 3821 3851 3853 3889 3911 3917 3929 7027 7043 7057 7121 7177 7187 7193 7207 7219 7229 7253 7297 7321 7349 7433 745
7 7459 7481 7507 7523 7529 7547 7561 7577 7589 7603 7643 7649 7673 7681 7687 7699 7717 7757 7817 7841 7867 7879 7901 792
7 7949 7951 7963 9001 9011 9013 9029 9041 9103 9127 9133 9161 9173 9209 9221 9227 9241 9257 9293 9341 9349 9403 9421 943
7 9439 9467 9479 9491 9497 9521 9533 9547 9551 9601 9613 9643 9661 9679 9721 9749 9769 9781 9787 9791 9803 9833 9857 987
1 9883 9923 9931 9941 9967
How many numbers (0 to exit)?
>

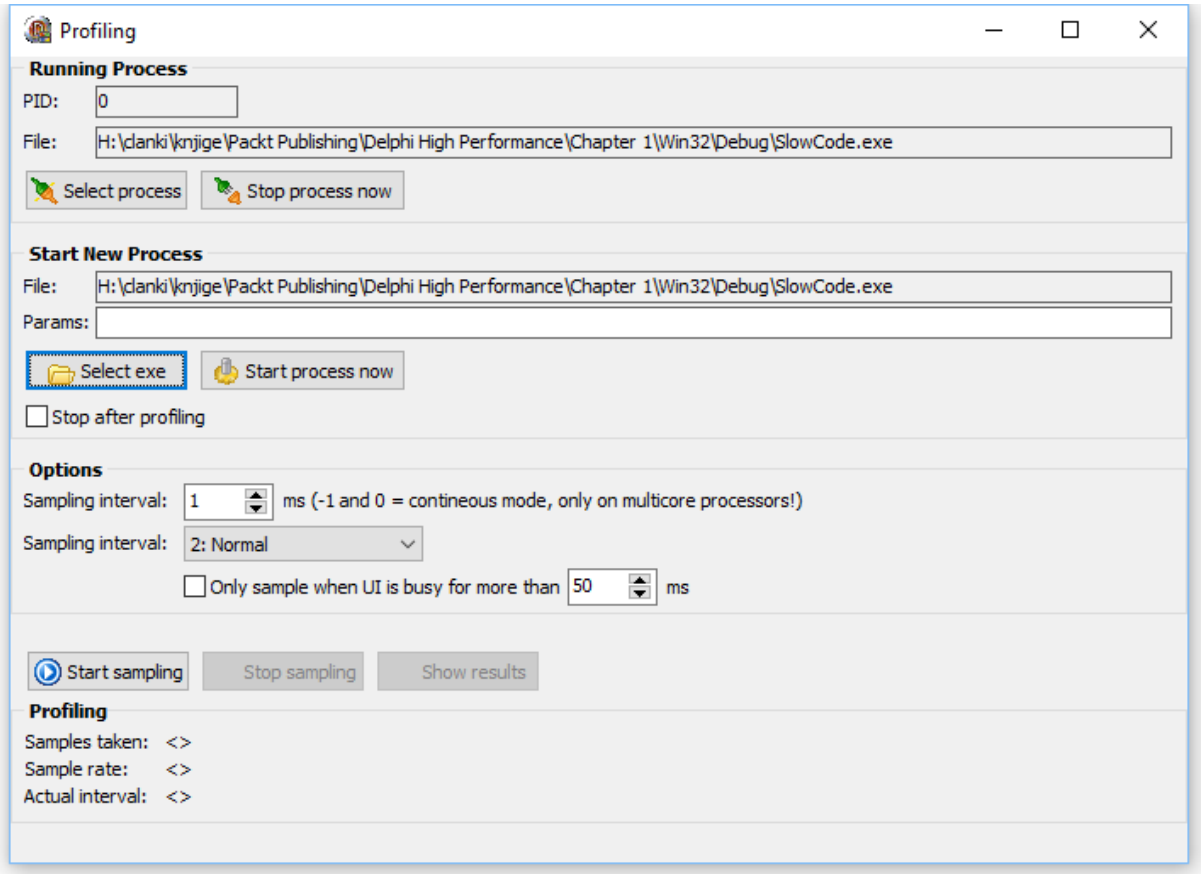
Chapter 2: Profiling the Code

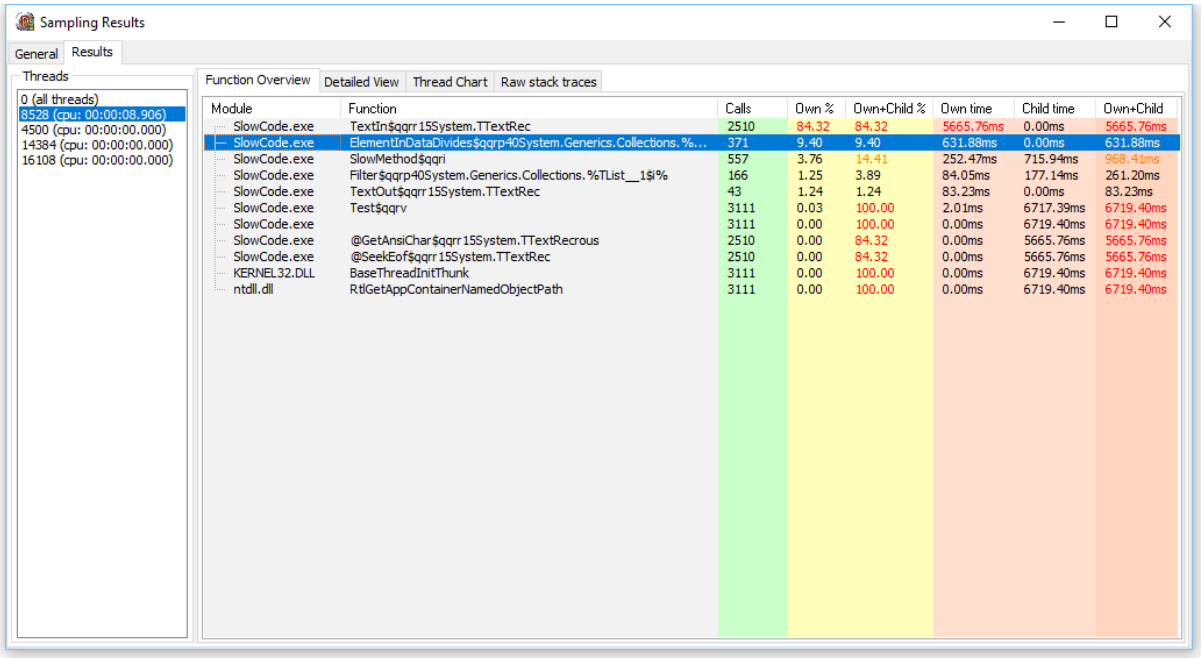


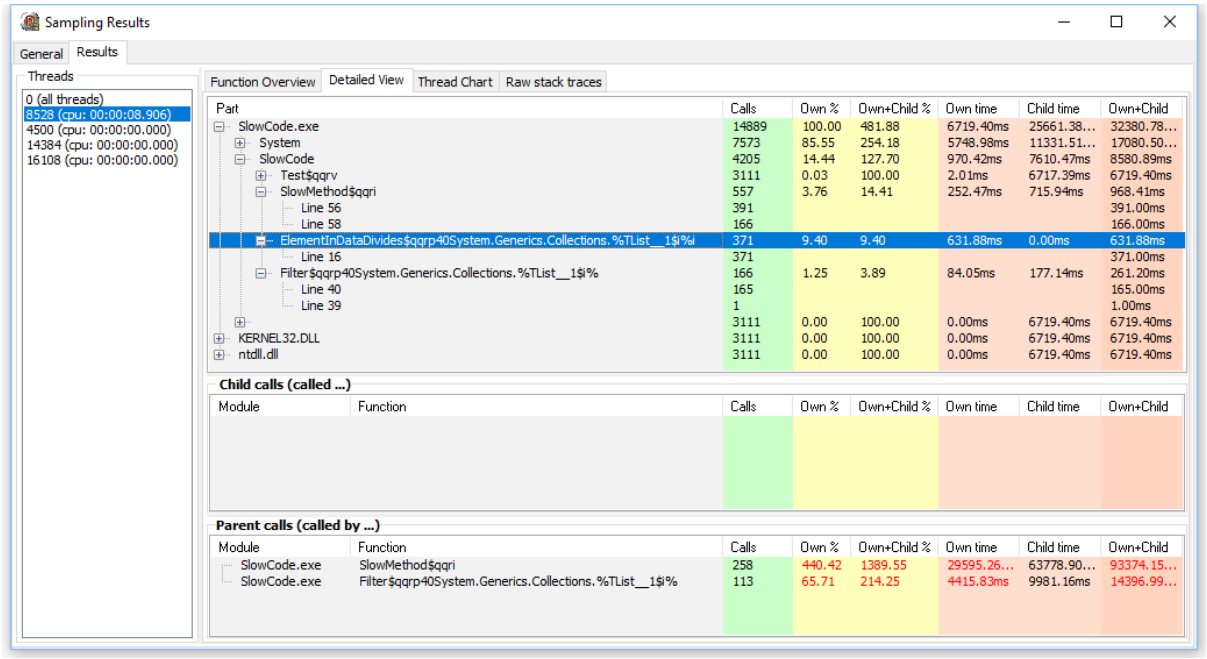


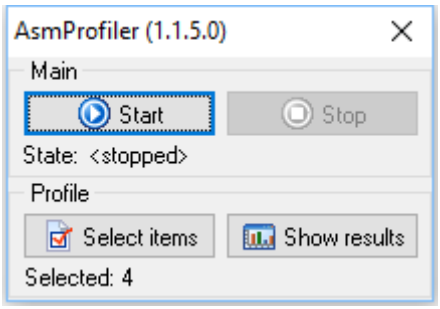


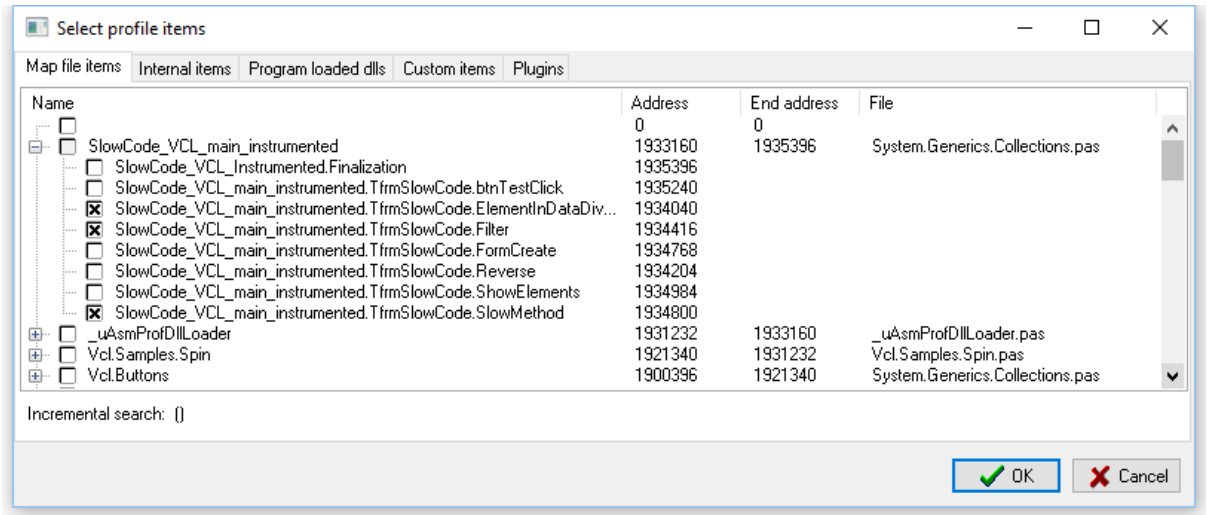


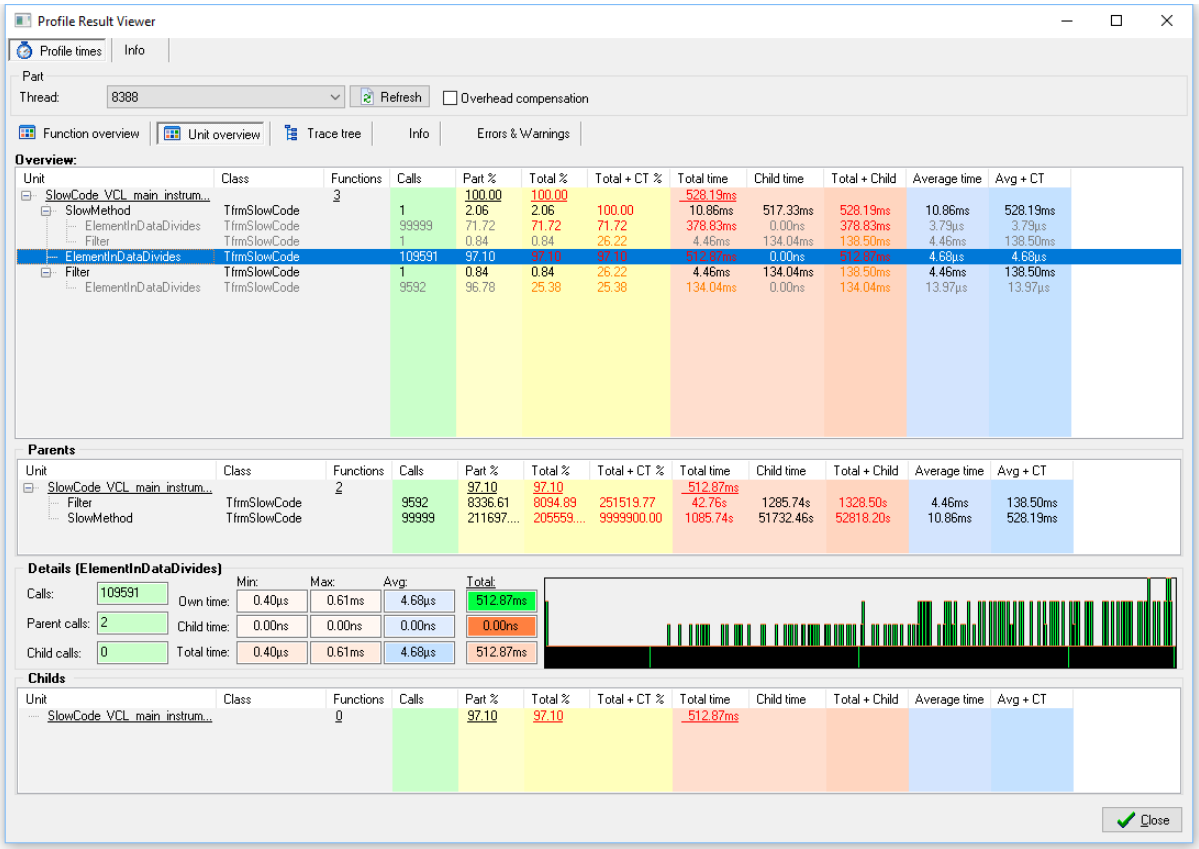

Profiling
Running Process
PID:
0
File:
H:\clanki\knjige\Packt Publishing\Delphi High Performance\Chapter 1\Win32\Debug\SlowCode.exe
Select process
Stop process now
Start New Process
File:
H:\clanki\knjige\Packt Publishing\Delphi High Performance\Chapter 1\Win32\Debug\SlowCode.exe
Params:
Select exe
Start process now
Stop after profiling
Options
Sampling interval:
1
ms (-1 and 0 = contineous mode, only on multicore processors!)
Sampling interval:
2: Normal
Only sample when UI is busy for more than
50
ms
Start sampling
Stop sampling
Show results
Profiling
Samples taken: <>
Sample rate: <>
Actual interval: <>

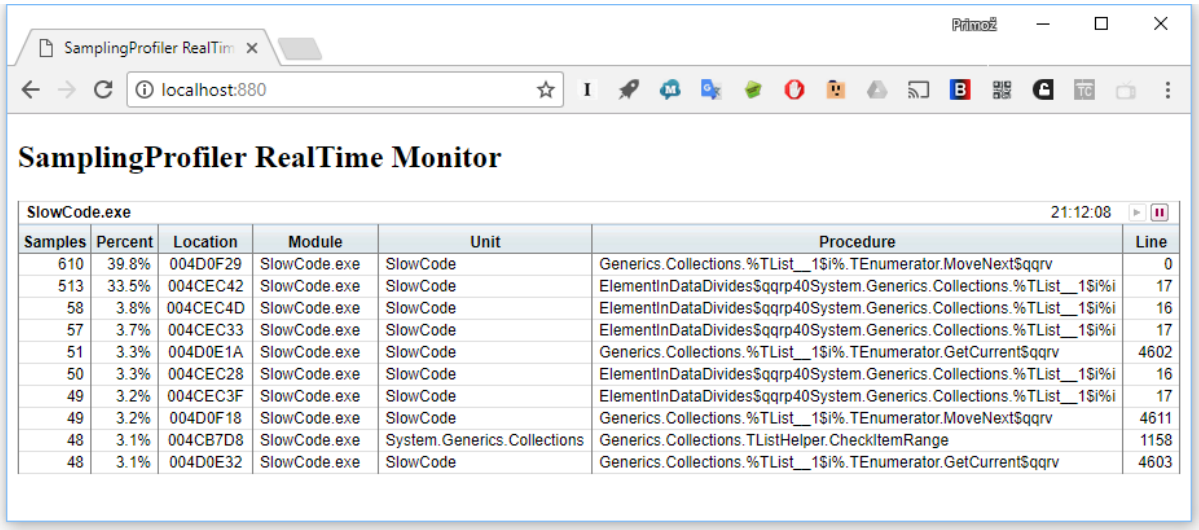

Sampling Results
General
Results
Threads
0 (all threads)
8528 (cpu: 00:00:08.906)
4500 (cpu: 00:00:00.000)
14384 (cpu: 00:00:00.000)
16108 (cpu: 00:00:00.000)
Function Overview
Detailed View
Thread Chart
Raw stack traces
Module
Function
Calls
Own %
Own+Child %
Own time
Child time
Own+Child
SlowCode.exe TextIn$qqrr15System.TTextRec 2510 84.32 84.32 5665.76ms 0.00ms 5665.76ms
SlowCode.exe ElementInDataDivides$qqrp40System.Generics.Collections.%... 371 9.40 9.40 631.88ms 0.00ms 631.88ms
SlowCode.exe SlowMethod$qqri 557 3.76 14.41 252.47ms 715.94ms 968.41ms
SlowCode.exe Filter$qqrp40System.Generics.Collections.%TList__1$i% 166 1.25 3.89 84.05ms 177.14ms 261.20ms
SlowCode.exe TextOut$qqrr15System.TTextRec 43 1.24 1.24 83.23ms 0.00ms 83.23ms
SlowCode.exe Test$qqrv 3111 0.03 100.00 2.01ms 6717.39ms 6719.40ms
SlowCode.exe 3111 0.00 100.00 0.00ms 6719.40ms 6719.40ms
SlowCode.exe @GetAnsiChar$qqrr15System.TTextRecrous 2510 0.00 84.32 0.00ms 5665.76ms 5665.76ms
SlowCode.exe @SeekEof$qqrr15System.TTextRec 2510 0.00 84.32 0.00ms 5665.76ms 5665.76ms
KERNEL32.DLL BaseThreadInitThunk 3111 0.00 100.00 0.00ms 6719.40ms 6719.40ms
ntdll.dll RtlGetAppContainerNamedObjectPath 3111 0.00 100.00 0.00ms 6719.40ms 6719.40ms

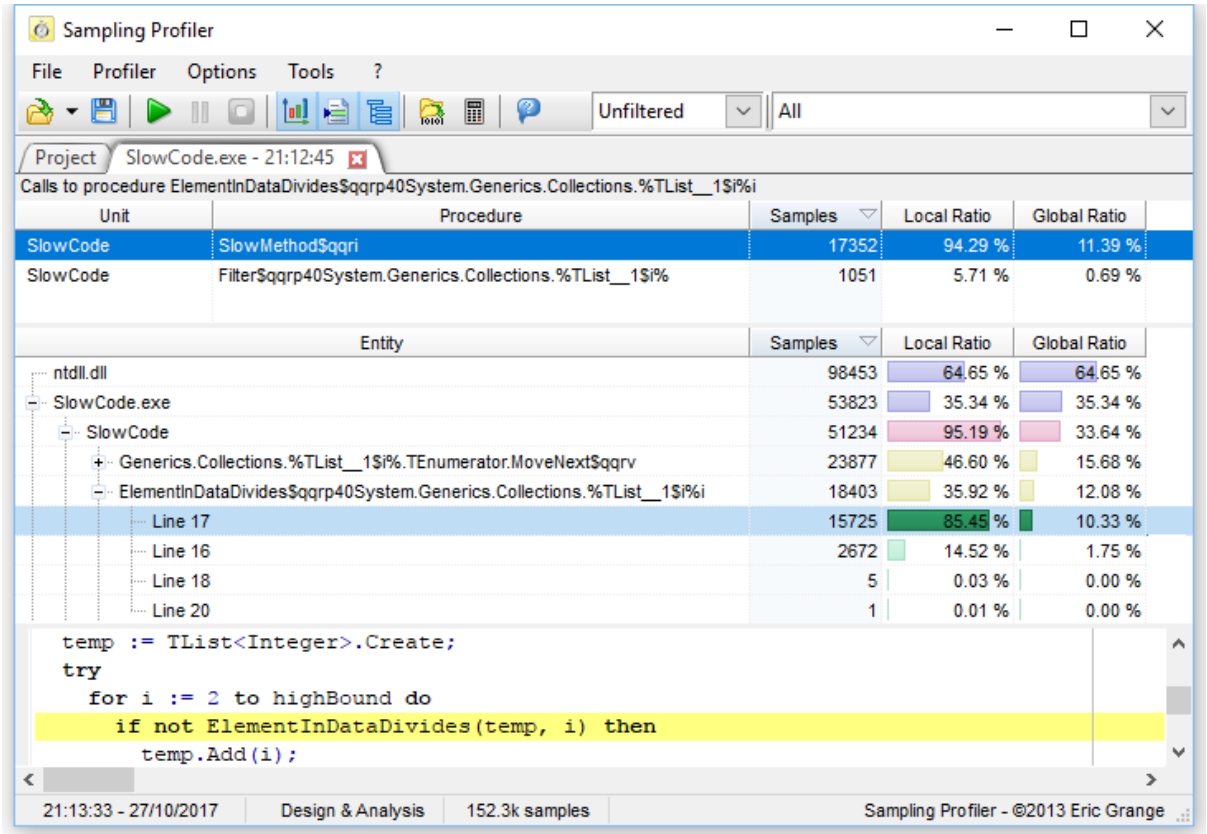

Sampling Results
General
Results
Threads
0 (all threads)
8528 (cpu: 00:00:08.906)
4500 (cpu: 00:00:00.000)
14384 (cpu: 00:00:00.000)
16108 (cpu: 00:00:00.000)
Function Overview
Detailed View
Thread Chart
Raw stack traces
Part
Calls
Own %
Own+Child %
Own time
Child time
Own+Child
SlowCode.exe 14889 100.00 481.88 6719.40ms 25661.38... 32380.78...
System 7573 85.55 254.18 5748.98ms 11331.51... 17080.50...
SlowCode 4205 14.44 127.70 970.42ms 7610.47ms 8580.89ms
Test$qqrv 3111 0.03 100.00 2.01ms 6717.39ms 6719.40ms
SlowMethod$qqri 557 3.76 14.41 252.47ms 715.94ms 968.41ms
Line 56 391 391.00ms
Line 58 166 166.00ms
ElementInDataDivides$qqrp40System.Generics.Collections.%TList__1$i%i 371 9.40 9.40 631.88ms 0.00ms 631.88ms
Line 16 371 371.00ms
Filter$qqrp40System.Generics.Collections.%TList__1$i% 166 1.25 3.89 84.05ms 177.14ms 261.20ms
Line 40 165 165.00ms
Line 39 1 1.00ms
3111 0.00 100.00 0.00ms 6719.40ms 6719.40ms
KERNEL32.DLL 3111 0.00 100.00 0.00ms 6719.40ms 6719.40ms
ntdll.dll 3111 0.00 100.00 0.00ms 6719.40ms 6719.40ms
Child calls (called ...)
Module
Function
Calls
Own %
Own+Child %
Own time
Child time
Own+Child
Parent calls (called by ...)
Module
Function
Calls
Own %
Own+Child %
Own time
Child time
Own+Child
SlowCode.exe SlowMethod$qqri 258 440.42 1389.55 29595.26... 63778.90... 93374.15...
SlowCode.exe Filter$qqrp40System.Generics.Collections.%TList__1$i% 113 65.71 214.25 4415.83ms 9981.16ms 14396.99...

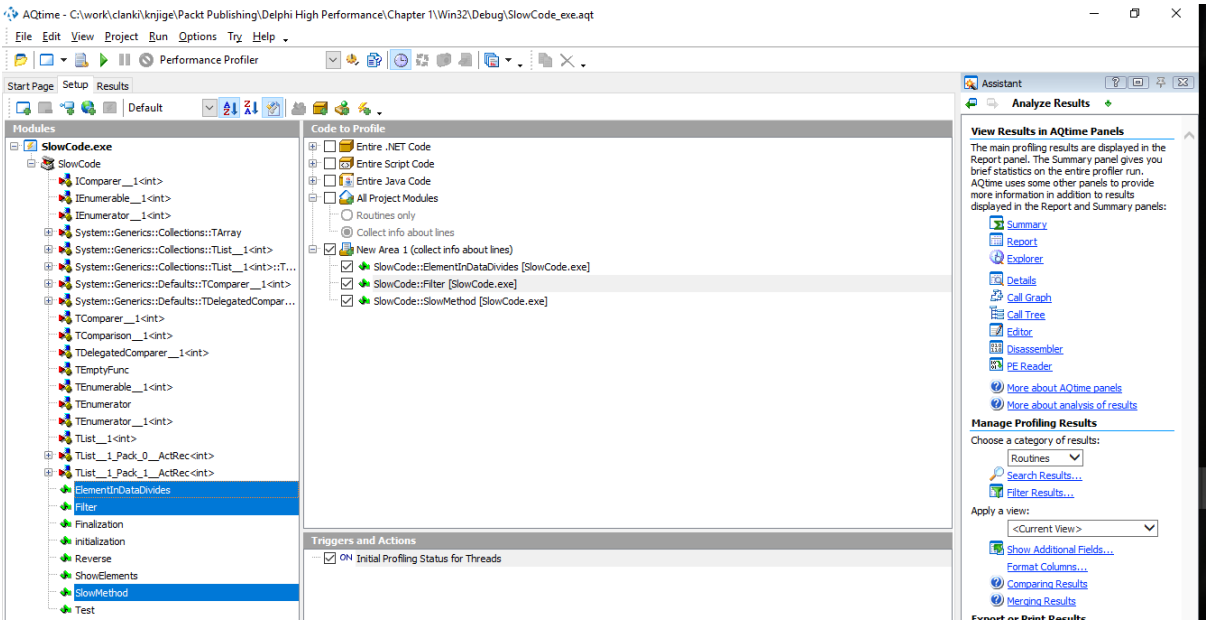

AsmProfiler (1.1.5.0)
Main
Start
Stop
State: <stopped>
Profile
Select items
Show results
Selected: 4

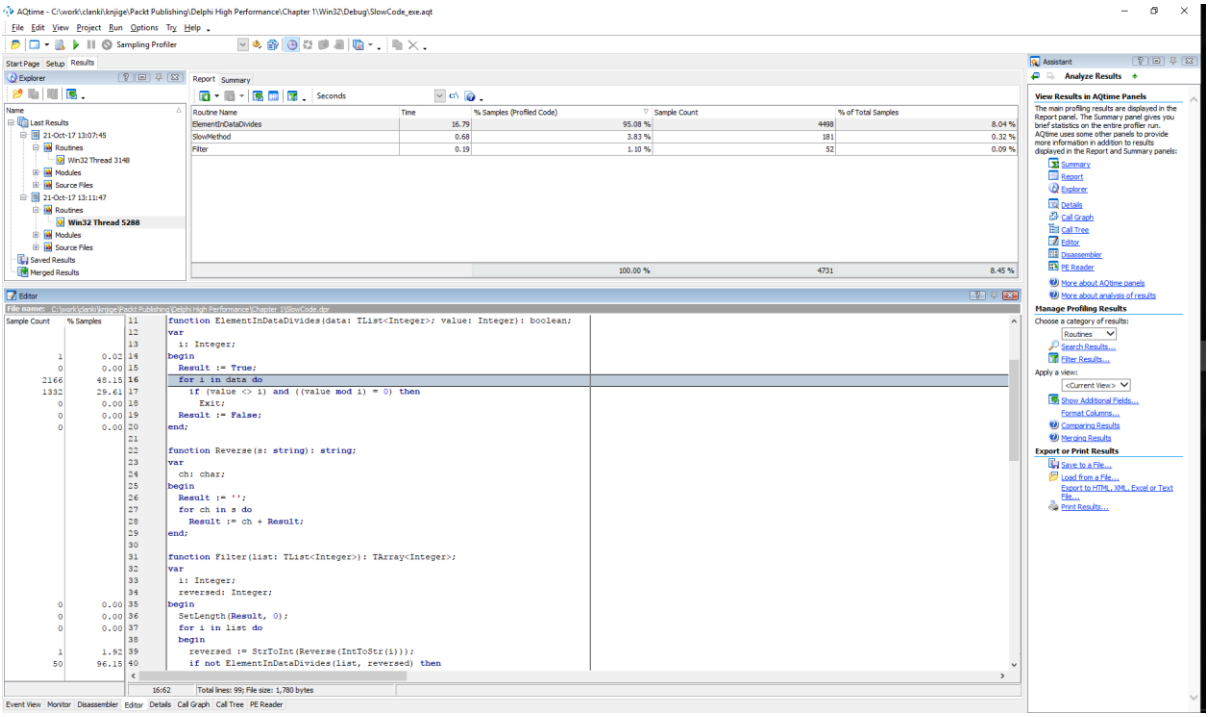

Select profile items
Map file items
Internal items
Program loaded dlls
Custom items
Plugins
Name
Address
End address
File
0 0
SlowCode_VCL_main_instrumented 1933160 1935396 System.Generics.Collections.pas
SlowCode_VCL_Instrumented.Finalization 1935396
SlowCode_VCL_main_instrumented.TfrmSlowCode.btnTestClick 1935240
SlowCode_VCL_main_instrumented.TfrmSlowCode.ElementInDataDiv... 1934040
SlowCode_VCL_main_instrumented.TfrmSlowCode.Filter 1934416
SlowCode_VCL_main_instrumented.TfrmSlowCode.FormCreate 1934768
SlowCode_VCL_main_instrumented.TfrmSlowCode.Reverse 1934204
SlowCode_VCL_main_instrumented.TfrmSlowCode.ShowElements 1934984
SlowCode_VCL_main_instrumented.TfrmSlowCode.SlowMethod 1934800
_uAsmProfDllLoader 1931232 1933160 _uAsmProfDllLoader.pas
Vcl.Samples.Spin 1921340 1931232 Vcl.Samples.Spin.pas
Vcl.Buttons 1900396 1921340 System.Generics.Collections.pas
Incremental search: ()
OK
Cancel

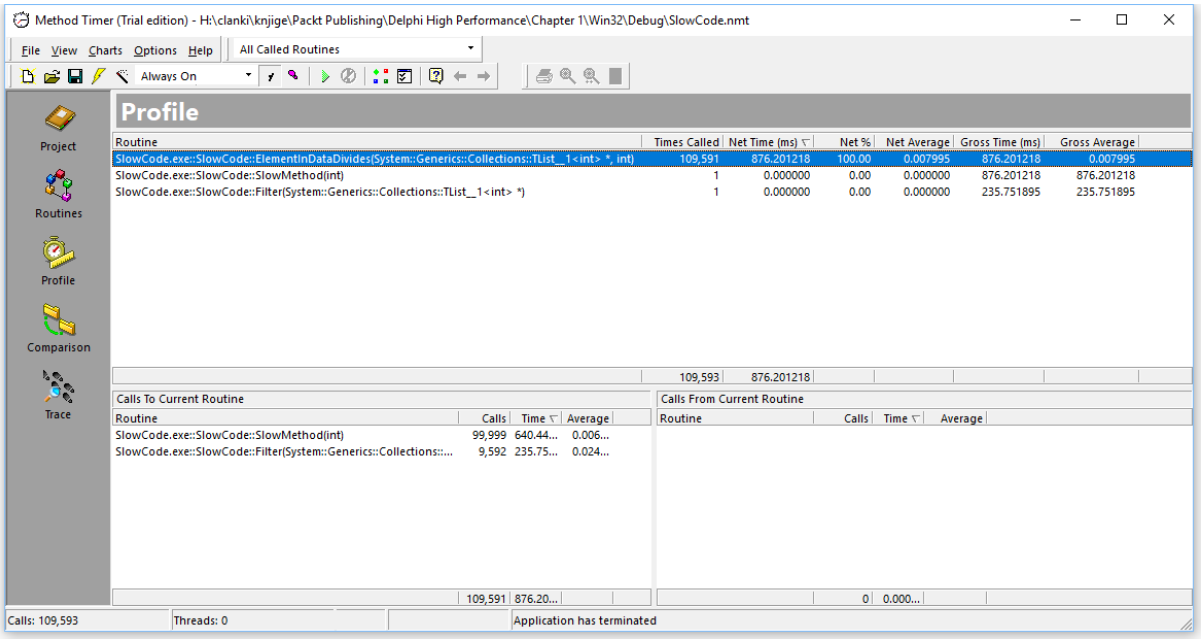

Profile Result Viewer
Profile times
Info
Part
Thread:
8388
Refresh
Overhead compensation
Function overview
Unit overview
Trace tree
Info
Errors & Warnings
Overview:
Unit
Class
Functions
Calls
Part %
Total %
Total + CT %
Total time
Child time
Total + Child
Average time
Avg + CT
SlowCode_VCL_main_instrum...
SlowMethod
ElementInDataDivides
Filter
Parents
Details (ElementInDataDivides)
Calls:
Parent calls:
Child calls:
Own time:
Child time:
Total time:
Min:
Max:
Avg:
Total:
Childs
Close

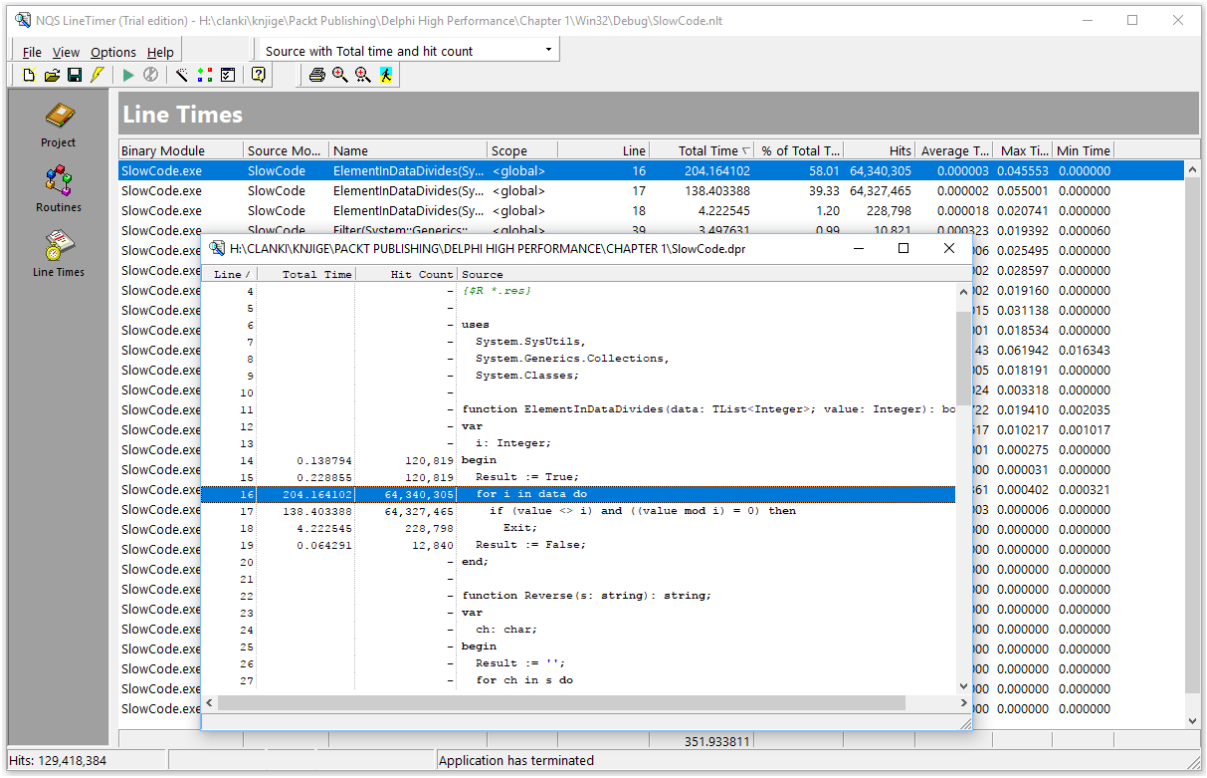

SamplingProfiler RealTim
localhost:880
SamplingProfiler RealTime Monitor
SlowCode.exe
21:12:08
Samples
Percent
Location
Module
Unit
Procedure
Line
610 39.8% 004D0F29 SlowCode.exe SlowCode Generics.Collections.%TList__1$i%.TEnumerator.MoveNext$qqrv 0
513 33.5% 004CEC42 SlowCode.exe SlowCode ElementInDataDivides$qqrp40System.Generics.Collections.%TList__1$i%i 17
58 3.8% 004CEC4D SlowCode.exe SlowCode ElementInDataDivides$qqrp40System.Generics.Collections.%TList__1$i%i 16
57 3.7% 004CEC33 SlowCode.exe SlowCode ElementInDataDivides$qqrp40System.Generics.Collections.%TList__1$i%i 17
51 3.3% 004D0E1A SlowCode.exe SlowCode Generics.Collections.%TList__1$i%.TEnumerator.GetCurrent$qqrv 4602
50 3.3% 004CEC28 SlowCode.exe SlowCode ElementInDataDivides$qqrp40System.Generics.Collections.%TList__1$i%i 16
49 3.2% 004CEC3F SlowCode.exe SlowCode ElementInDataDivides$qqrp40System.Generics.Collections.%TList__1$i%i 17
49 3.2% 004D0F18 SlowCode.exe SlowCode Generics.Collections.%TList__1$i%.TEnumerator.MoveNext$qqrv 4611
48 3.1% 004CB7D8 SlowCode.exe System.Generics.Collections Generics.Collections.TListHelper.CheckItemRange 1158
48 3.1% 004D0E32 SlowCode.exe SlowCode Generics.Collections.%TList__1$i%.TEnumerator.GetCurrent$qqrv 4603


Sampling Profiler
File Profiler Options Tools ?
Unfiltered
All
Project SlowCode.exe - 21:12:45
Calls to procedure ElementInDataDivides$qqrp40System.Generics.Collections.%TList__1$i%i
Unit Procedure Samples Local Ratio Global Ratio
SlowCode SlowMethod$qqri 17352 94.29 % 11.39 %
SlowCode Filter$qqrp40System.Generics.Collections.%TList__1$i% 1051 5.71 % 0.69 %
Entity Samples Local Ratio Global Ratio
ntdll.dll 98453 64.65 % 64.65 %
SlowCode.exe 53823 35.34 % 35.34 %
SlowCode 51234 95.19 % 33.64 %
Generics.Collections.%TList__1$i%.TEnumerator.MoveNext$qqrv 23877 46.60 % 15.68 %
ElementInDataDivides$qqrp40System.Generics.Collections.%TList__1$i%i 18403 35.92 % 12.08 %
Line 17 15725 85.45 % 10.33 %
Line 16 2672 14.52 % 1.75 %
Line 18 5 0.03 % 0.00 %
Line 20 1 0.01 % 0.00 %
temp := TList<Integer>.Create;
try
for i := 2 to highBound do
if not ElementInDataDivides(temp, i) then
temp.Add(i);
21:13:33 - 27/10/2017
Design & Analysis
152.3k samples
Sampling Profiler - ©2013 Eric Grange

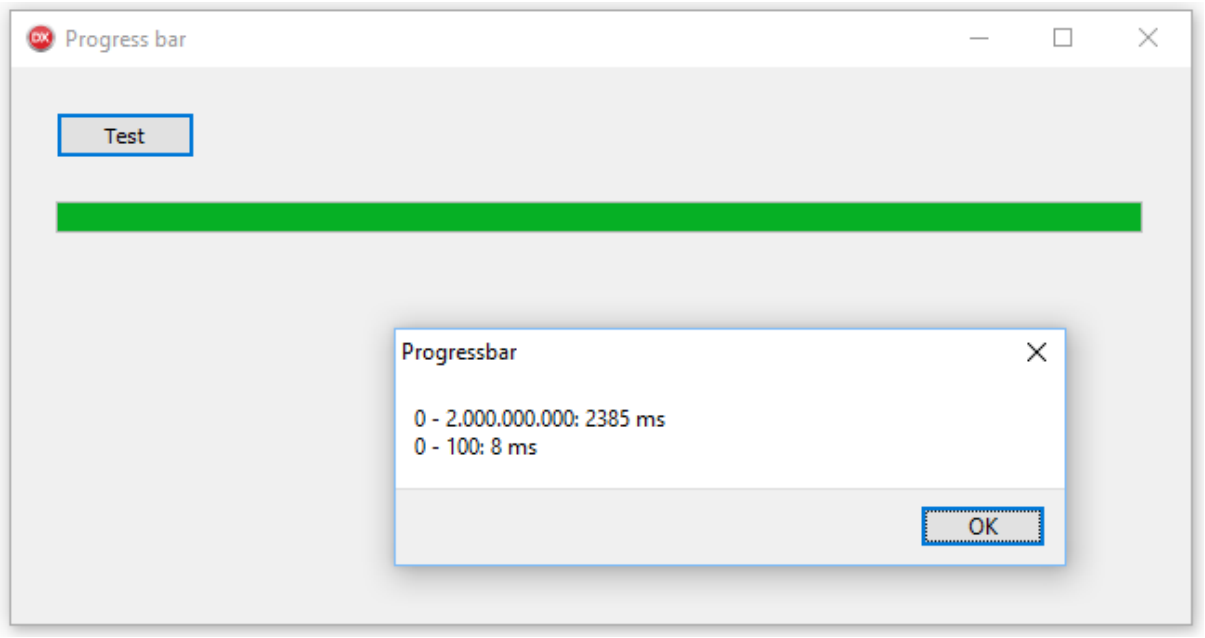

AQtime - C:\work\clanki\knjige\Packt Publishing\Delphi High Performance\Chapter 1\Win32\Debug\SlowCode_exe.aqt
File Edit View Project Run Options Try Help
Performance Profiler
Start Page Setup Results
Modules
SlowCode.exe
SlowCode
Code to Profile
Entire .NET Code
Entire Script Code
Entire Java Code
All Project Modules
Routines only
Collect info about lines
New Area 1 (collect info about lines)
SlowCode::ElementInDataDivides [SlowCode.exe]
SlowCode::Filter [SlowCode.exe]
SlowCode::SlowMethod [SlowCode.exe]
Triggers and Actions
Initial Profiling Status for Threads
Assistant
Analyze Results
View Results in AQtime Panels
The main profiling results are displayed in the Report panel. The Summary panel gives you brief statistics on the entire profiler run. AQtime uses some other panels to provide more information in addition to results displayed in the Report and Summary panels:
Summary
Report
Explorer
Details
Call Graph
Call Tree
Editor
Disassembler
PE Reader
More about AQtime panels
More about analysis of results
Manage Profiling Results
Choose a category of results:
Routines
Search Results...
Filter Results...
Apply a view:
<Current View>
Show Additional Fields...
Format Columns...
Comparing Results
Merging Results

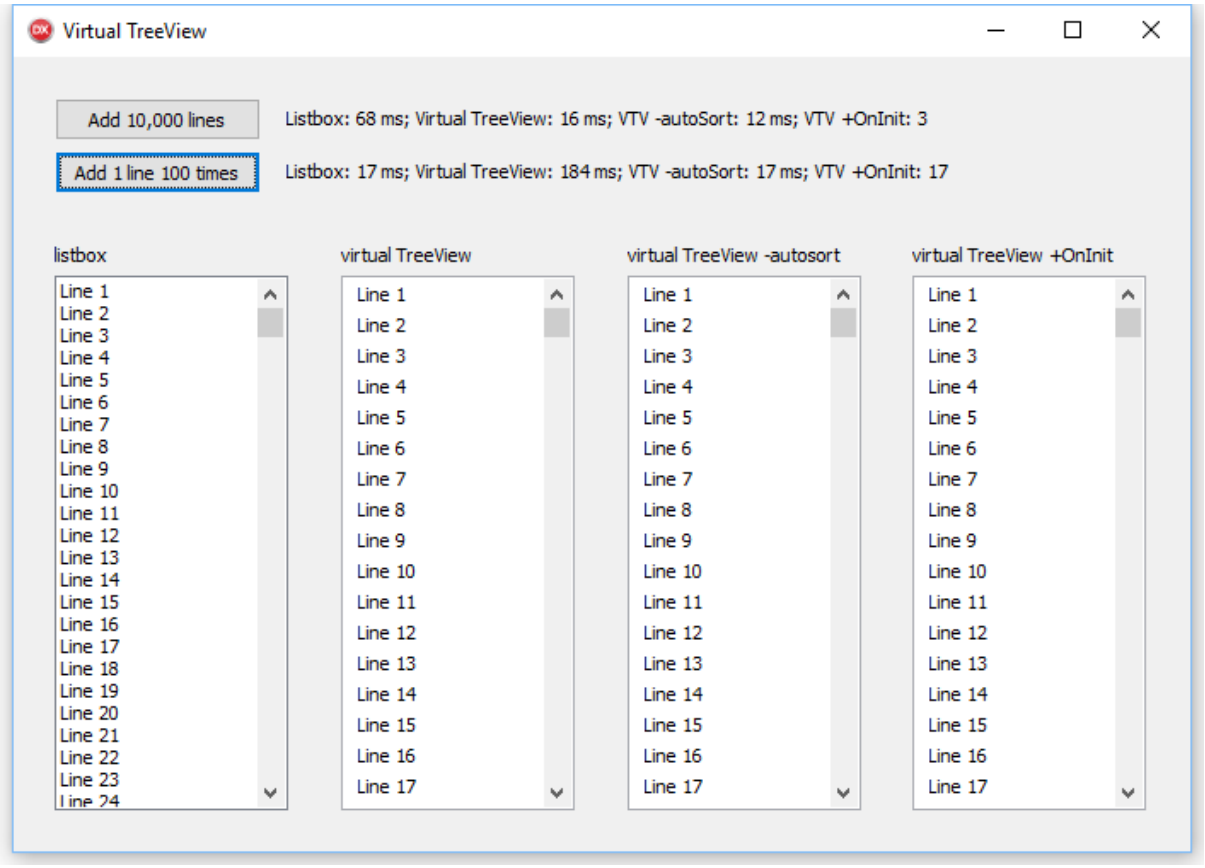

AQtime - C:\work\clanki\knjige\Packt Publishing\Delphi High Performance\Chapter 1\Win32\Debug\SlowCode_exe.aqt
Sampling Profiler
Routine Name
Time
% Samples (Profiled Code)
Sample Count
% of Total Samples
ElementInDataDivides
SlowMethod
Filter
Editor
Assistant
Analyze Results

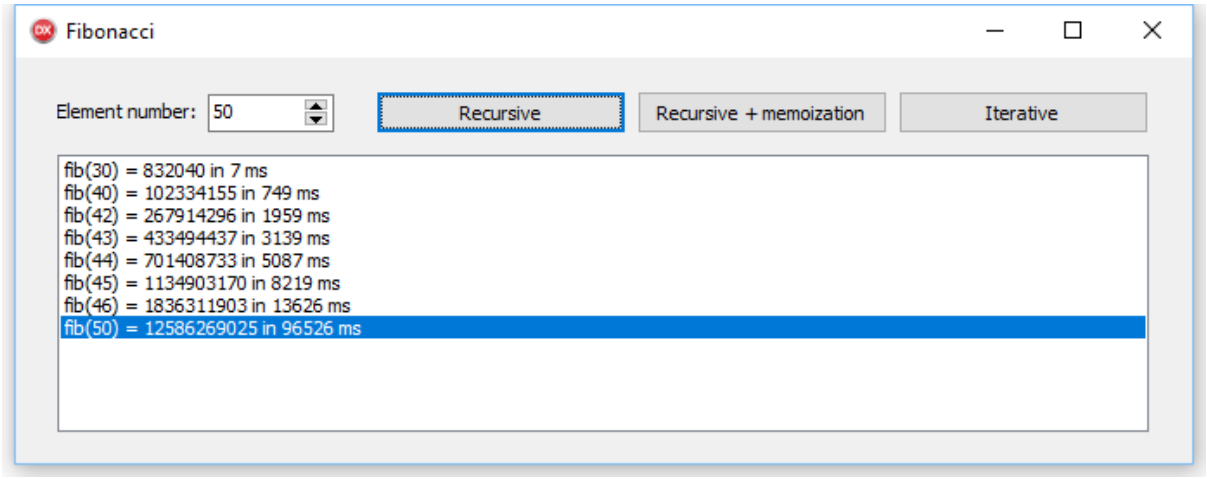

Method Timer (Trial edition) - H:\clanki\knjige\Packt Publishing\Delphi High Performance\Chapter 1\Win32\Debug\SlowCode.nmt
File View Charts Options Help
All Called Routines
Always On
Profile
Project
Routines
Profile
Comparison
Trace
Routine
Times Called
Net Time (ms)
Net %
Net Average
Gross Time (ms)
Gross Average
SlowCode.exe::SlowCode::ElementInDataDivides(System::Generics::Collections::TList__1<int> *, int)
SlowCode.exe::SlowCode::SlowMethod(int)
SlowCode.exe::SlowCode::Filter(System::Generics::Collections::TList__1<int> *)
Calls To Current Routine
Calls From Current Routine
Calls
Time
Average
Calls: 109,593
Threads: 0
Application has terminated

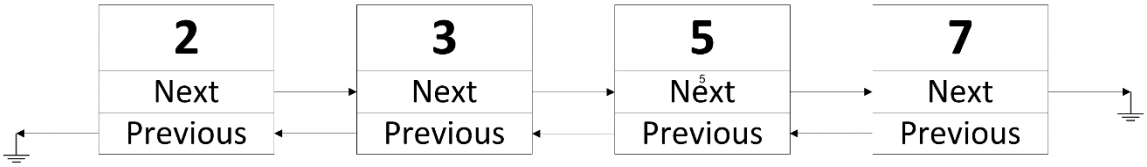

NQS LineTimer (Trial edition) - H:\clanki\knjige\Packt Publishing\Delphi High Performance\Chapter 1\Win32\Debug\SlowCode.nlt
File View Options Help
Source with Total time and hit count
Project
Routines
Line Times
Line Times
Binary Module Source Mo... Name Scope Line Total Time % of Total T... Hits Average T... Max Ti... Min Time
SlowCode.exe SlowCode ElementInDataDivides(Sy... <global> 16 204.164102 58.01 64,340,305 0.000003 0.045553 0.000000
SlowCode.exe SlowCode ElementInDataDivides(Sy... <global> 17 138.403388 39.33 64,327,465 0.000002 0.055001 0.000000
SlowCode.exe SlowCode ElementInDataDivides(Sy... <global> 18 4.222545 1.20 228,798 0.000018 0.020741 0.000000
SlowCode.exe SlowCode Filter(System::Generics... <global> 39 3.497631 0.99 10,821 0.000323 0.019392 0.000060
H:\CLANKI\KNJIGE\PACKT PUBLISHING\DELPHI HIGH PERFORMANCE\CHAPTER 1\SlowCode.dpr
Line / Total Time Hit Count Source
4 - {$R *.res}
5 -
6 - uses
7 - System.SysUtils,
8 - System.Generics.Collections,
9 - System.Classes;
10 -
11 - function ElementInDataDivides(data: TList<Integer>; value: Integer): bo
12 - var
13 - i: Integer;
14 0.138794 120,819 begin
15 0.228855 120,819 Result := True;
16 204.164102 64,340,305 for i in data do
17 138.403388 64,327,465 if (value <> i) and ((value mod i) = 0) then
18 4.222545 228,798 Exit;
19 0.064291 12,840 Result := False;
20 - end;
21 -
22 - function Reverse(s: string): string;
23 - var
24 - ch: char;
25 - begin
26 - Result := '';
27 - for ch in s do
351.933811
Hits: 129,418,384
Application has terminated

Chapter 3: Fixing the Algorithm

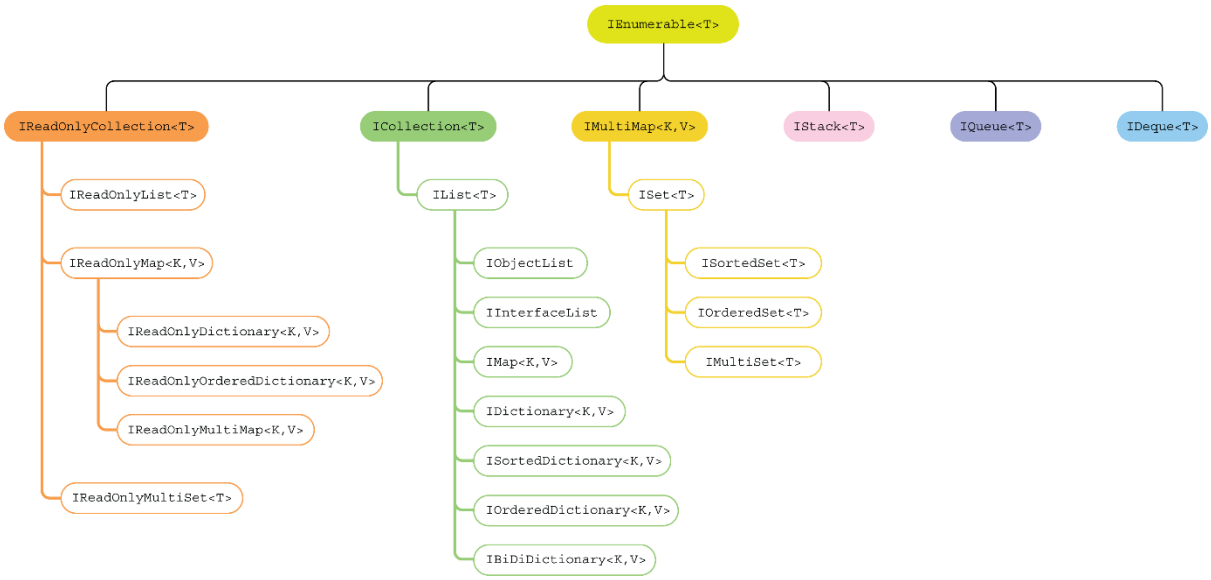


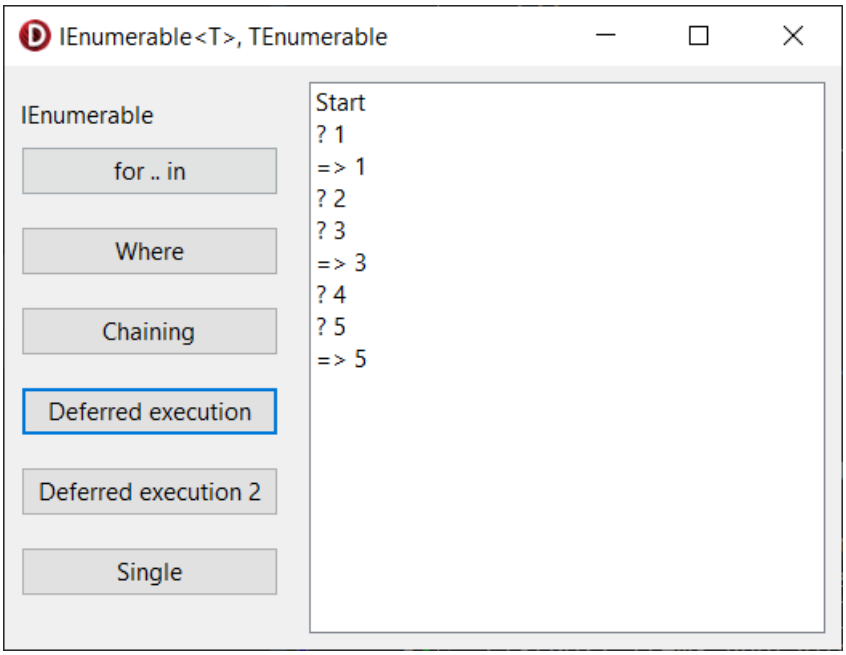


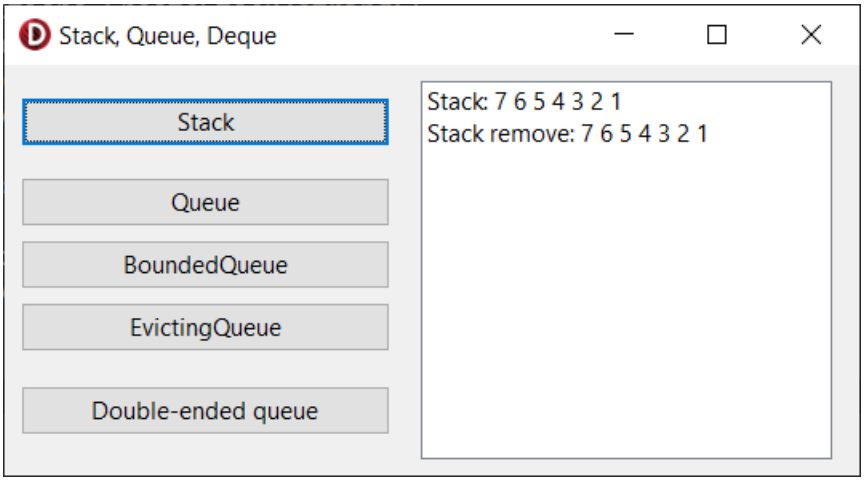

Fibonacci
Element number:
50
Recursive
Recursive + memoization
Iterative
fib(30) = 832040 in 7 ms
fib(40) = 102334155 in 749 ms
fib(42) = 267914296 in 1959 ms
fib(43) = 433494437 in 3139 ms
fib(44) = 701408733 in 5087 ms
fib(45) = 1134903170 in 8219 ms
fib(46) = 1836311903 in 13626 ms
fib(50) = 12586269025 in 96526 ms

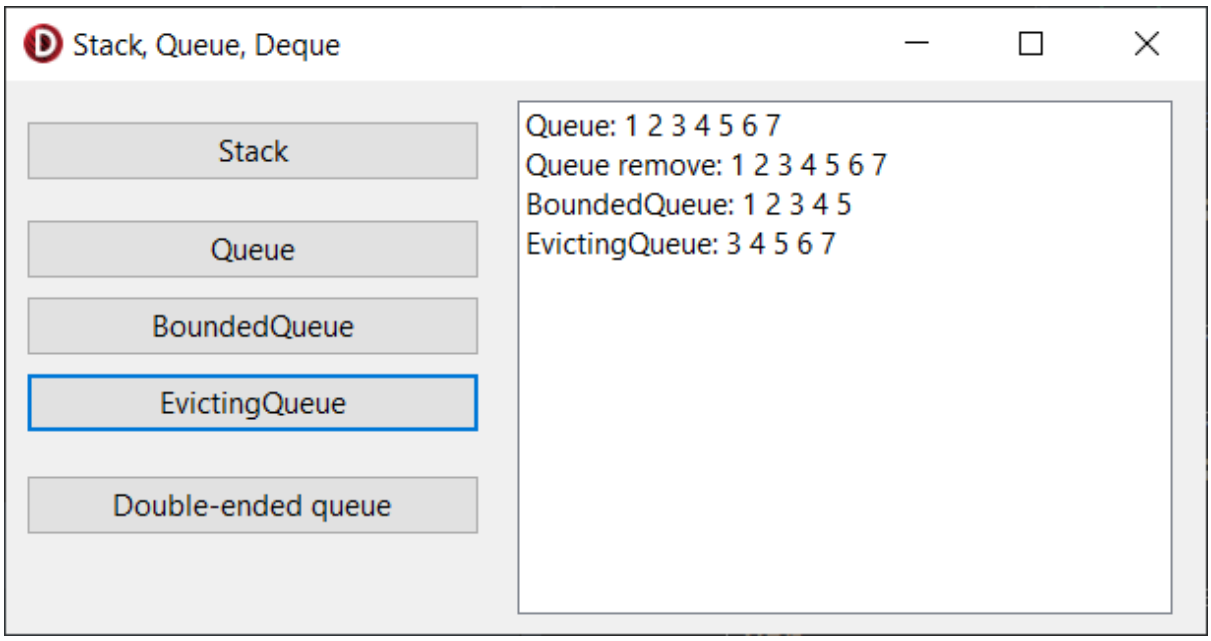

2
Next
Previous
3
Next
Previous
5
Next
Previous
7
Next
Previous

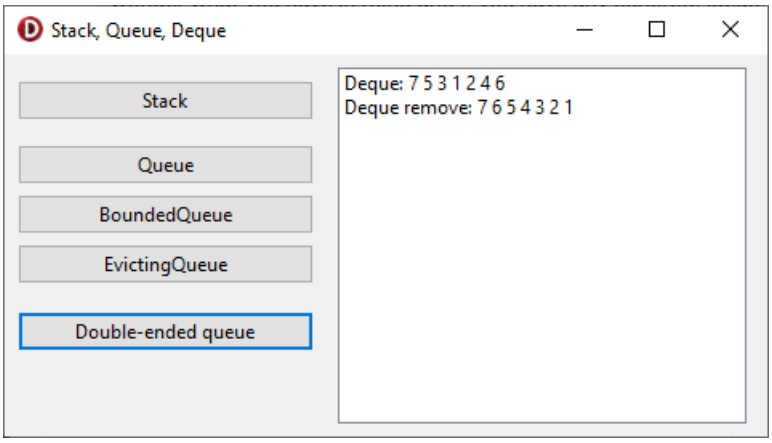
Chapter 4: Don’t Reinvent, Reuse

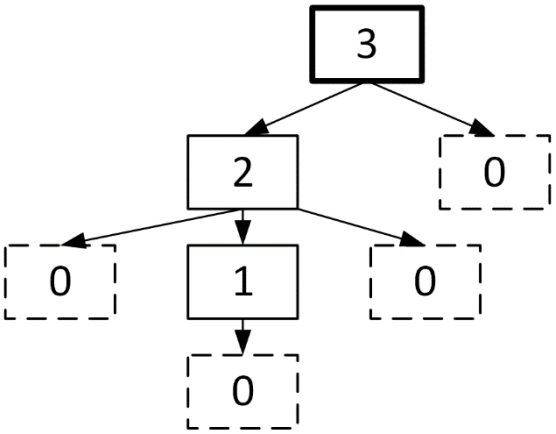


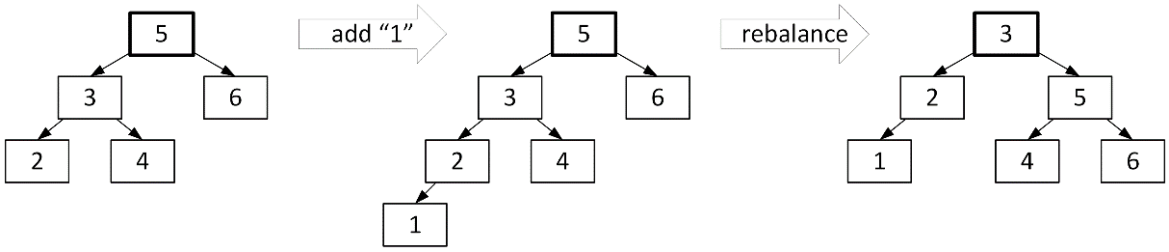


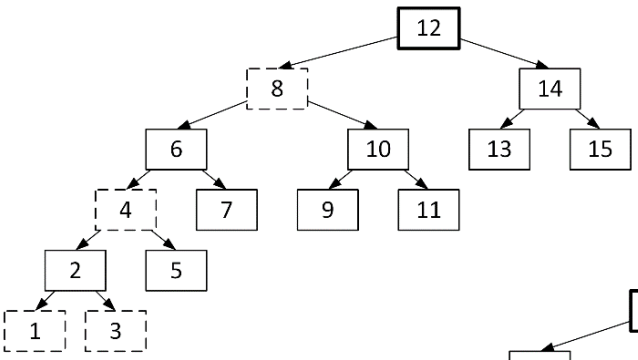


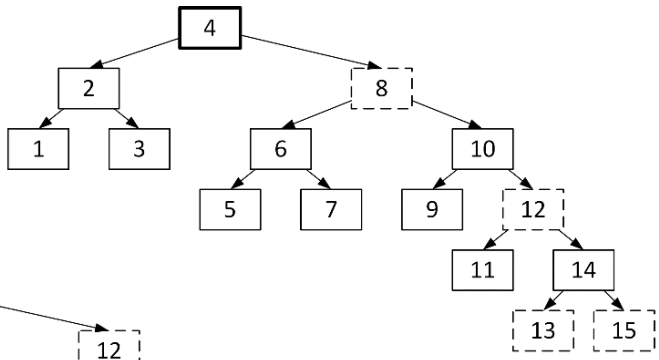

Stack, Queue, Deque
Stack
Queue
BoundedQueue
EvictingQueue
Double-ended queue
Queue: 1 2 3 4 5 6 7
Queue remove: 1 2 3 4 5 6 7
BoundedQueue: 1 2 3 4 5
EvictingQueue: 3 4 5 6 7

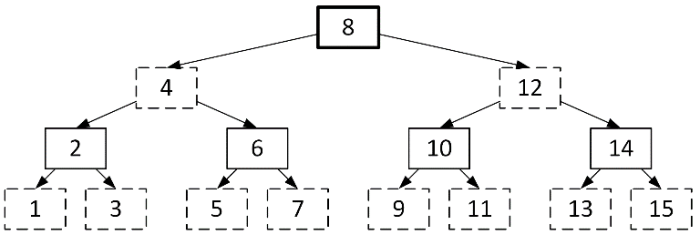

Stack, Queue, Deque
Stack
Queue
BoundedQueue
EvictingQueue
Double-ended queue
Deque: 7 5 3 1 2 4 6
Deque remove: 7 6 5 4 3 2 1

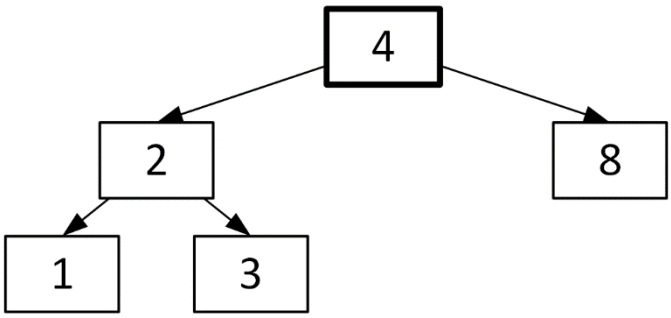

3
2
0
0
1
0
0

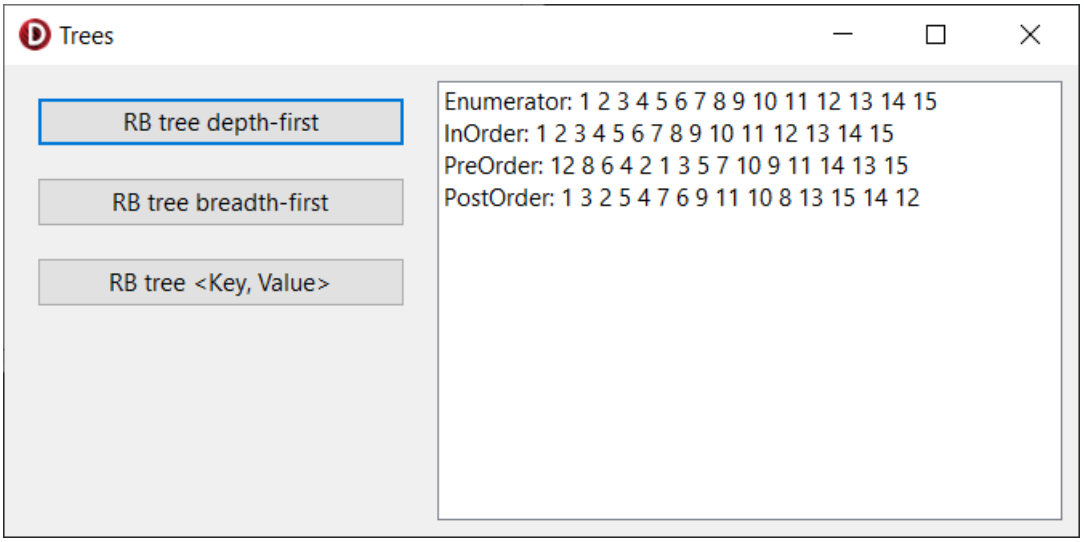

5
3
6
2
4
add “1”
5
3
6
2
4
1
rebalance
3
2
5
1
4
6

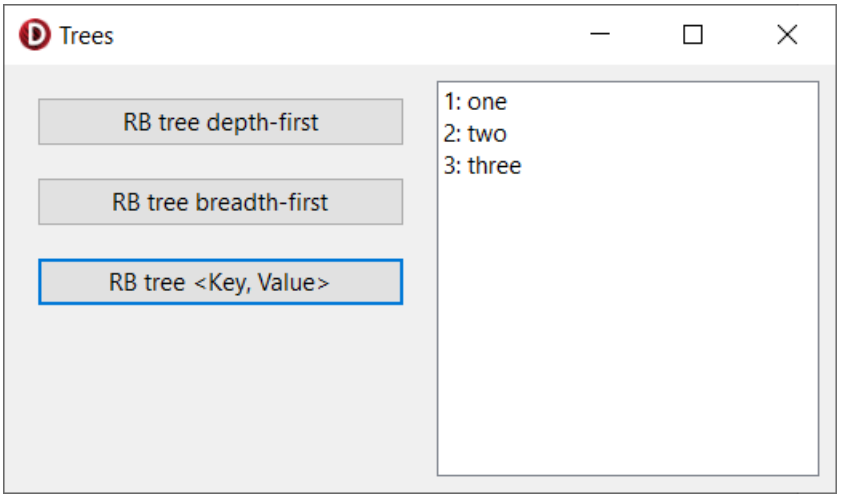

12
8
14
6
10
13
15
4
7
9
11
2
5
1
3

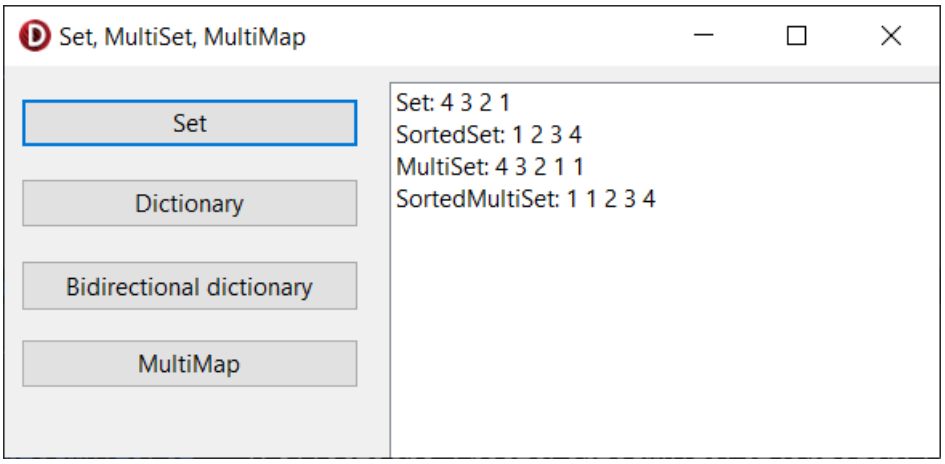

4
2
8
1
3
6
10
5
7
9
12
11
14
13
15

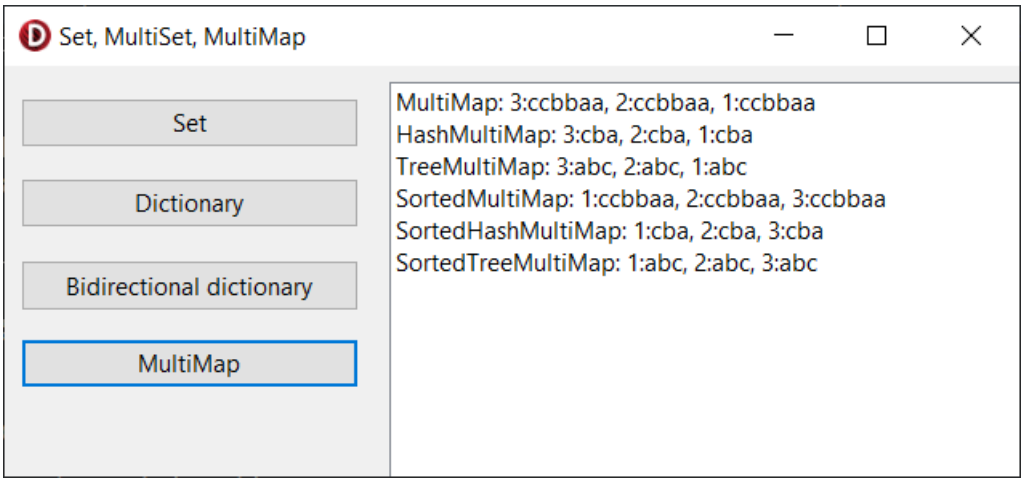

8
4
12
2
6
10
14
1
3
5
7
9
11
13
15


4
2
8
1
3

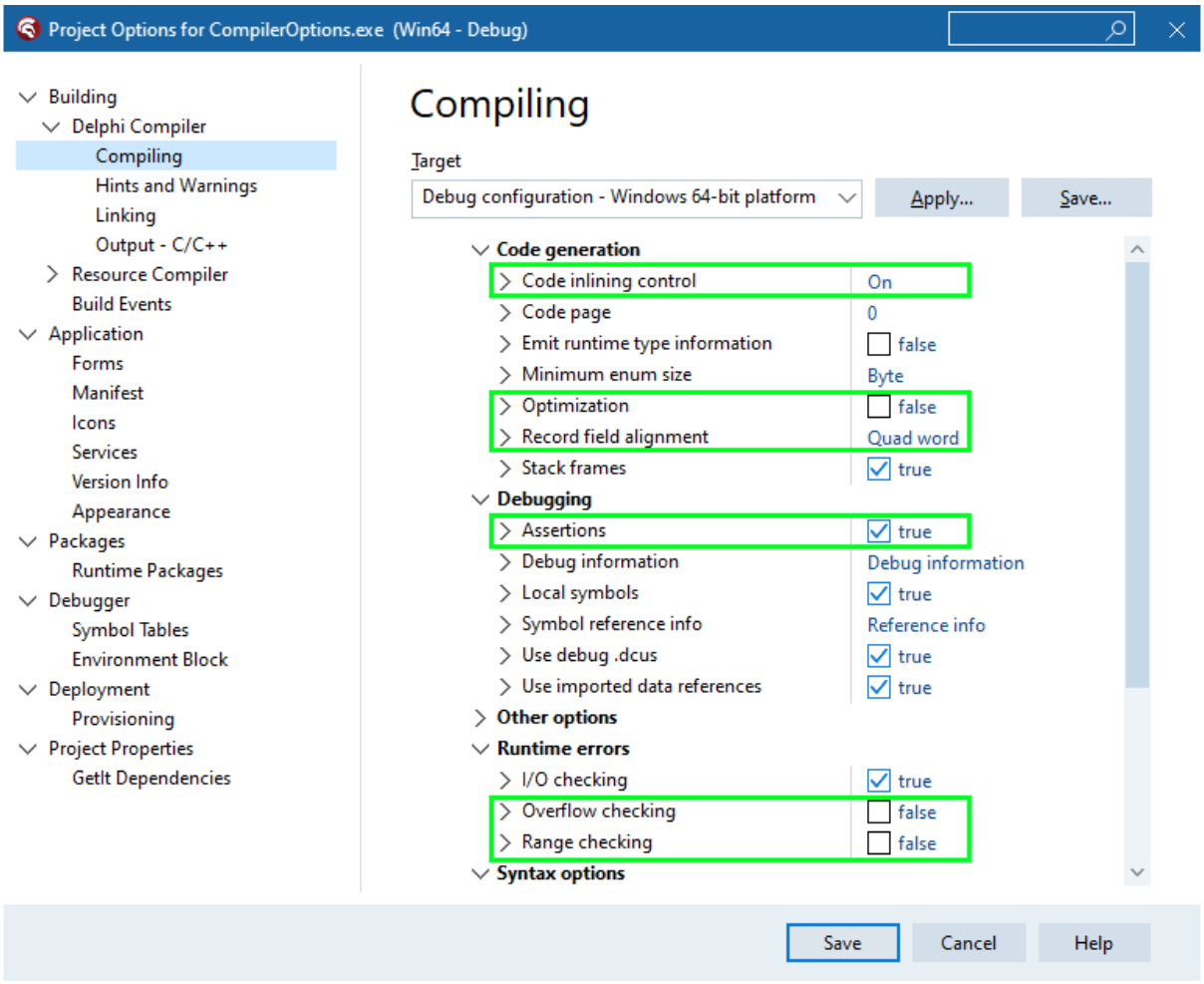

Trees
RB tree depth-first
RB tree breadth-first
RB tree <Key, Value>
Enumerator: 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
InOrder: 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
PreOrder: 12 8 6 4 2 1 3 5 7 10 9 11 14 13 15
PostOrder: 1 3 2 5 4 7 6 9 11 10 8 13 15 14 12

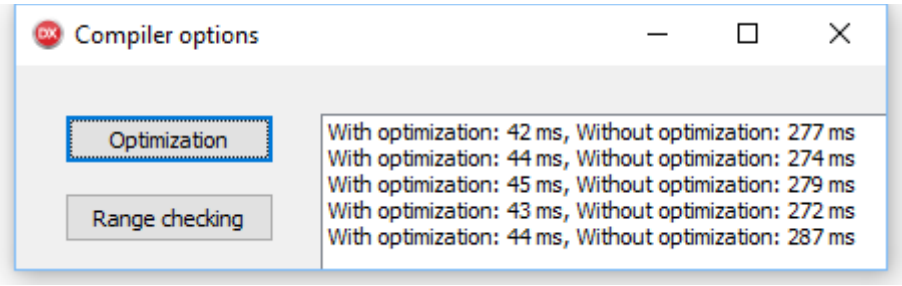

Trees
RB tree depth-first
RB tree breadth-first
RB tree <Key, Value>
1: one
2: two
3: three


Set, MultiSet, MultiMap
Set
Dictionary
Bidirectional dictionary
MultiMap
Set: 4 3 2 1
SortedSet: 1 2 3 4
MultiSet: 4 3 2 1 1
SortedMultiSet: 1 1 2 3 4

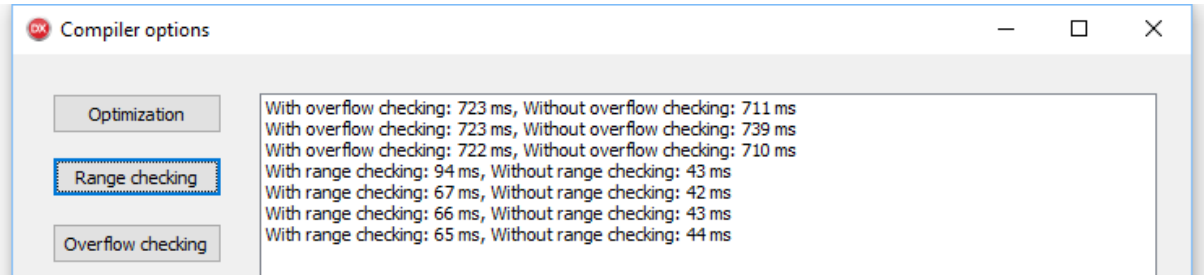

Set, MultiSet, MultiMap
Set
Dictionary
Bidirectional dictionary
MultiMap
MultiMap: 3:ccbbaa, 2:ccbbaa, 1:ccbbaa
HashMultiMap: 3:cba, 2:cba, 1:cba
TreeMultiMap: 3:abc, 2:abc, 1:abc
SortedMultiMap: 1:ccbbaa, 2:ccbbaa, 3:ccbbaa
SortedHashMultiMap: 1:cba, 2:cba, 3:cba
SortedTreeMultiMap: 1:abc, 2:abc, 3:abc

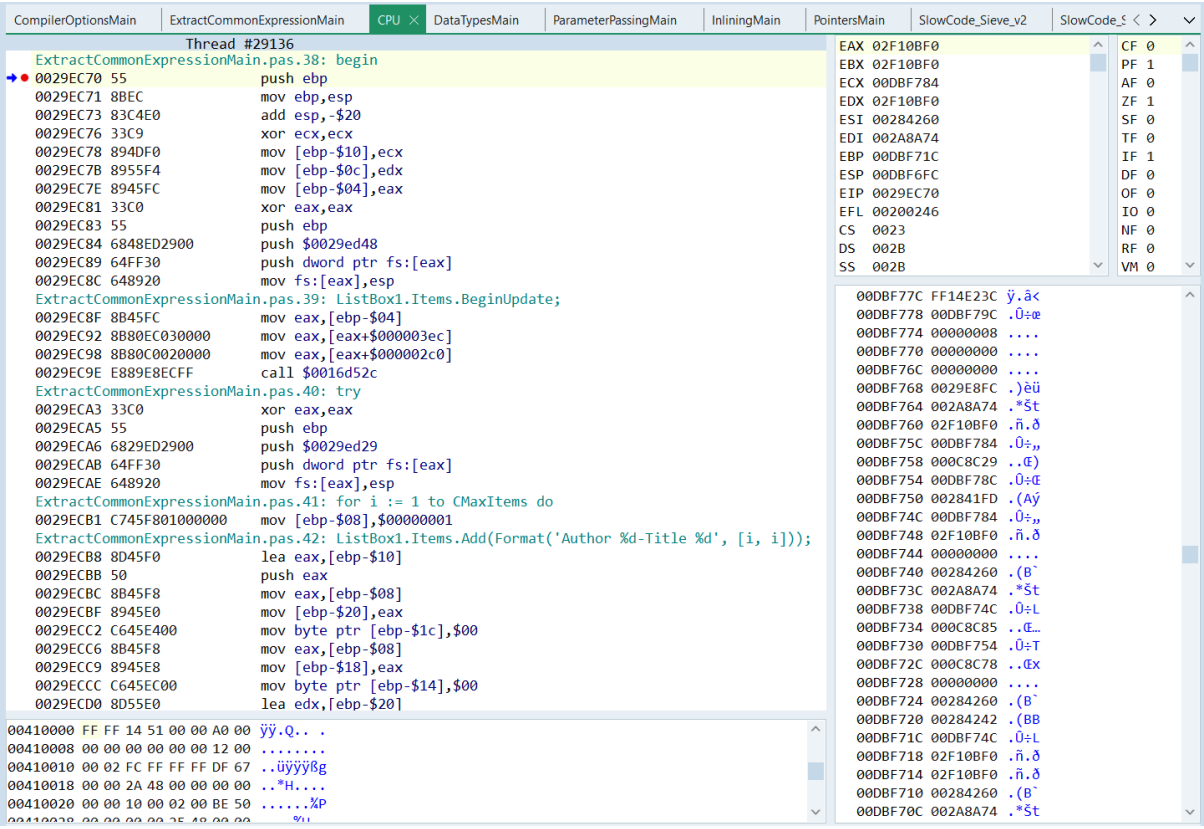
Chapter 5: Fine-Tuning the Code





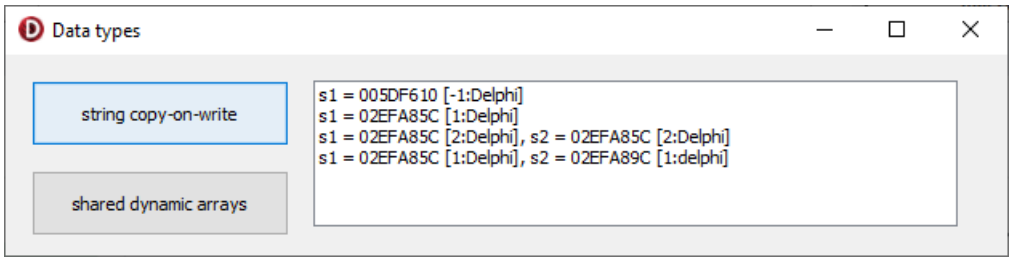
address	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
alignment																																
$A1	F1	F2								F3		F4																				
$A2	F1	×	F2								F3		F4																			
$A4	F1	×	×	×	F2								F3		×	×	F4															
$A8	F1	×	×	×	×	×	×	×	F2								F3		×	×	×	×	×	×	F4							

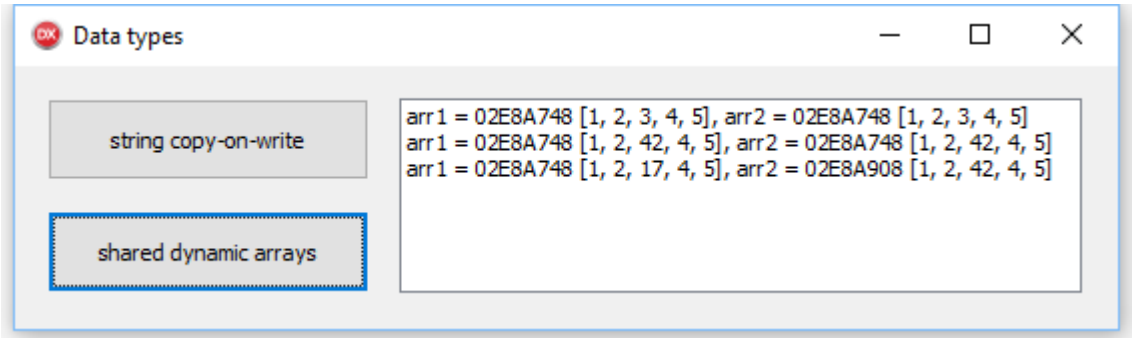

Compiler options
Optimization
Range checking
Overflow checking
With overflow checking: 723 ms, Without overflow checking: 711 ms
With overflow checking: 723 ms, Without overflow checking: 739 ms
With overflow checking: 722 ms, Without overflow checking: 710 ms
With range checking: 94 ms, Without range checking: 43 ms
With range checking: 67 ms, Without range checking: 42 ms
With range checking: 66 ms, Without range checking: 43 ms
With range checking: 65 ms, Without range checking: 44 ms

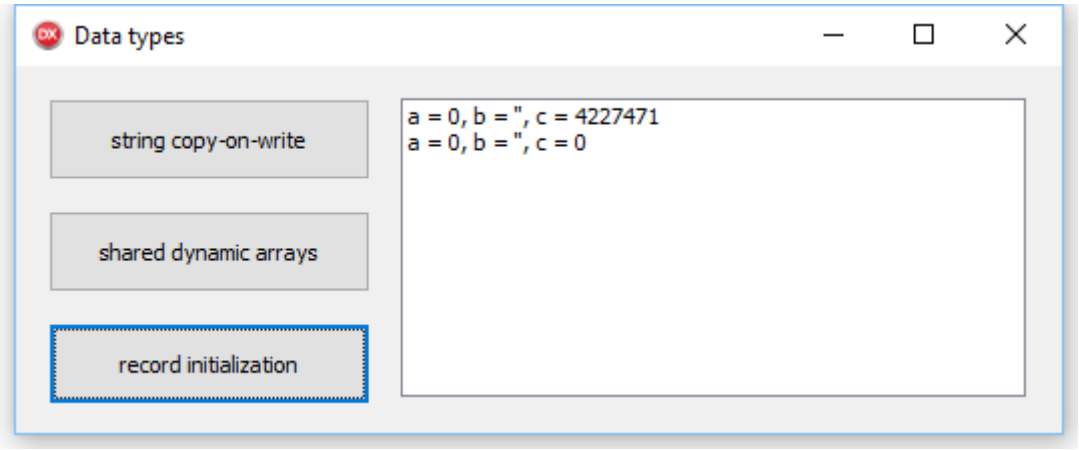

CompilerOptionsMain
ExtractCommonExpressionMain
CPU
DataTypesMain
ParameterPassingMain
InliningMain
PointersMain
SlowCode_Sieve_v2
Thread #29136
ExtractCommonExpressionMain.pas.38: begin
0029EC70 55 push ebp
0029EC71 8BEC mov ebp,esp
0029EC73 83C4E0 add esp,-$20
0029EC76 33C9 xor ecx,ecx
0029EC78 894DF0 mov [ebp-$10],ecx
0029EC7B 8955F4 mov [ebp-$0c],edx
0029EC7E 8945FC mov [ebp-$04],eax
0029EC81 33C0 xor eax,eax
0029EC83 55 push ebp
0029EC84 6848ED2900 push $0029ed48
0029EC89 64FF30 push dword ptr fs:[eax]
0029EC8C 648920 mov fs:[eax],esp
ExtractCommonExpressionMain.pas.39: ListBox1.Items.BeginUpdate;
0029EC8F 8B45FC mov eax,[ebp-$04]
0029EC92 8B80EC030000 mov eax,[eax+$000003ec]
0029EC98 8B80C0020000 mov eax,[eax+$000002c0]
0029EC9E E889E8ECFF call $0016d52c
ExtractCommonExpressionMain.pas.40: try
0029ECA3 33C0 xor eax,eax
0029ECA5 55 push ebp
0029ECA6 6829ED2900 push $0029ed29
0029ECAB 64FF30 push dword ptr fs:[eax]
0029ECAE 648920 mov fs:[eax],esp
ExtractCommonExpressionMain.pas.41: for i := 1 to CMaxItems do
0029ECB1 C745F801000000 mov [ebp-$08],$00000001
ExtractCommonExpressionMain.pas.42: ListBox1.Items.Add(Format('Author %d-Title %d', [i, i]));
0029ECB8 8D45F0 lea eax,[ebp-$10]
0029ECBB 50 push eax
0029ECBC 8B45F8 mov eax,[ebp-$08]
0029ECBF 8945E0 mov [ebp-$20],eax
0029ECC2 C645E400 mov byte ptr [ebp-$1c],$00
0029ECC6 8B45F8 mov eax,[ebp-$08]
0029ECC9 8945E8 mov [ebp-$18],eax
0029ECCC C645EC00 mov byte ptr [ebp-$14],$00
0029ECD0 8D55E0 lea edx,[ebp-$20]
00410000 FF FF 14 51 00 00 A0 00 ÿÿ.Q.. .
00410008 00 00 00 00 00 00 12 00 ....
00410010 00 02 FC FF FF FF DF 67 ..üÿÿÿßg
00410018 00 00 2A 48 00 00 00 00 ..*H....
00410020 00 00 10 00 02 00 BE 50 ......¾P
EAX 02F10BF0
EBX 02F10BF0
ECX 00DBF784
EDX 02F10BF0
ESI 00284260
EDI 002A8A74
EBP 00DBF71C
ESP 00DBF6FC
EIP 0029EC70
EFL 00200246
CS 0023
DS 002B
SS 002B
CF 0
PF 1
AF 0
ZF 1
SF 0
TF 0
IF 1
DF 0
OF 0
IO 0
NF 0
RF 0
VM 0
00DBF77C FF14E23C ÿ.â<
00DBF778 00DBF79C .Û÷œ
00DBF774 00000008 ....
00DBF770 00000000 ....
00DBF76C 00000000 ....
00DBF768 0029E8FC .)èü
00DBF764 002A8A74 .*Št
00DBF760 02F10BF0 .ñ.ð
00DBF75C 00DBF784 .Û÷„
00DBF758 000C8C29 ..Œ)
00DBF754 00DBF78C .Û÷Œ
00DBF750 002841FD .(Aý
00DBF74C 00DBF784 .Û÷„
00DBF748 02F10BF0 .ñ.ð
00DBF744 00000000 ....
00DBF740 00284260 .(B`
00DBF73C 002A8A74 .*Št
00DBF738 00DBF74C .Û÷L
00DBF734 000C8C85 ..Œ…
00DBF730 00DBF754 .Û÷T
00DBF72C 000C8C78 ..Œx
00DBF728 00000000 ....
00DBF724 00284260 .(B`
00DBF720 00284242 .(BB
00DBF71C 00DBF74C .Û÷L
00DBF718 02F10BF0 .ñ.ð
00DBF714 02F10BF0 .ñ.ð
00DBF710 00284260 .(B`
00DBF70C 002A8A74 .*Št

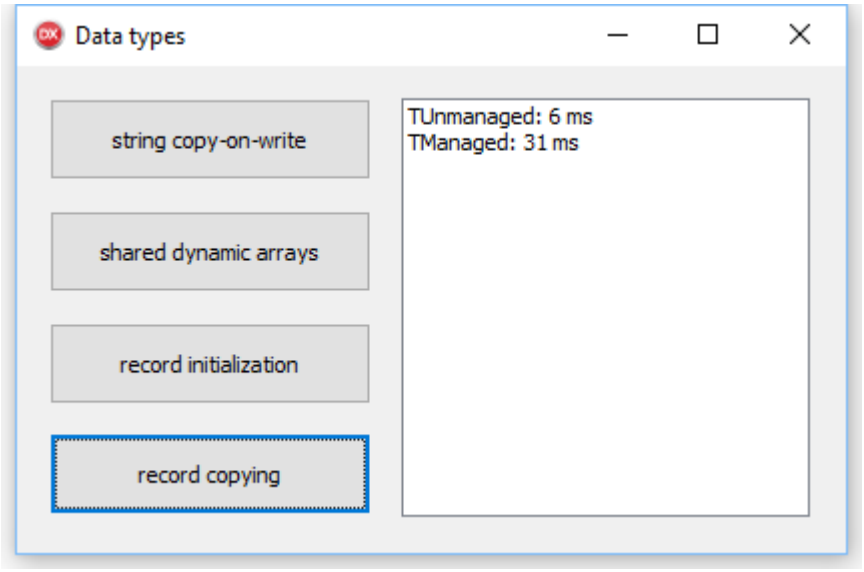
```
ExtractCommonExpressionMain.pas.55: ListBox1.Items[i] :=
005DEE0F 8D45D0            lea eax,[ebp-$30]
005DEE12 50                push eax
005DEE13 8D4DCC            lea ecx,[ebp-$34]
005DEE16 8B45FC            mov eax,[ebp-$04]
005DEE19 8B80EC030000      mov eax,[eax+$000003ec]
005DEE1F 8B80C0020000      mov eax,[eax+$000002c0]
005DEE25 8B55F8            mov edx,[ebp-$08]
005DEE28 8B18              mov ebx,[eax]
005DEE2A FF530C            call dword ptr [ebx+$0c]
005DEE2D 8B45CC            mov eax,[ebp-$34]
005DEE30 E85FBAE2FF        call @UStrLen
005DEE35 50                push eax
005DEE36 8D4DC8            lea ecx,[ebp-$38]
005DEE39 8B45FC            mov eax,[ebp-$04]
005DEE3C 8B80EC030000      mov eax,[eax+$000003ec]
005DEE42 8B80C0020000      mov eax,[eax+$000002c0]
005DEE48 8B55F8            mov edx,[ebp-$08]
005DEE4B 8B18              mov ebx,[eax]
005DEE4D FF530C            call dword ptr [ebx+$0c]
005DEE50 8B55C8            mov edx,[ebp-$38]
005DEE53 B901000000        mov ecx,$00000001
005DEE58 B8ACEF5D00        mov eax,$005defac
005DEE5D E8B665E2FF        call Pos
005DEE62 40                inc eax
005DEE63 50                push eax
005DEE64 8D4DC4            lea ecx,[ebp-$3c]
005DEE67 8B45FC            mov eax,[ebp-$04]
005DEE6A 8B80EC030000      mov eax,[eax+$000003ec]
005DEE70 8B80C0020000      mov eax,[eax+$000002c0]
005DEE76 8B55F8            mov edx,[ebp-$08]
005DEE79 8B18              mov ebx,[eax]
005DEE7B FF530C            call dword ptr [ebx+$0c]
005DEE7E 8B45C4            mov eax,[ebp-$3c]
005DEE81 5A                pop edx
005DEE82 59                pop ecx
005DEE83 E854C3E2FF        call @UStrCopy
005DEE88 FF75D0            push dword ptr [ebp-$30]
005DEE8B 68ACEF5D00        push $005defac
005DEE90 8D45C0            lea eax,[ebp-$40]
005DEE93 50                push eax
005DEE94 8D4DBC            lea ecx,[ebp-$44]
005DEE97 8B45FC            mov eax,[ebp-$04]
005DEE9A 8B80EC030000      mov eax,[eax+$000003ec]
```

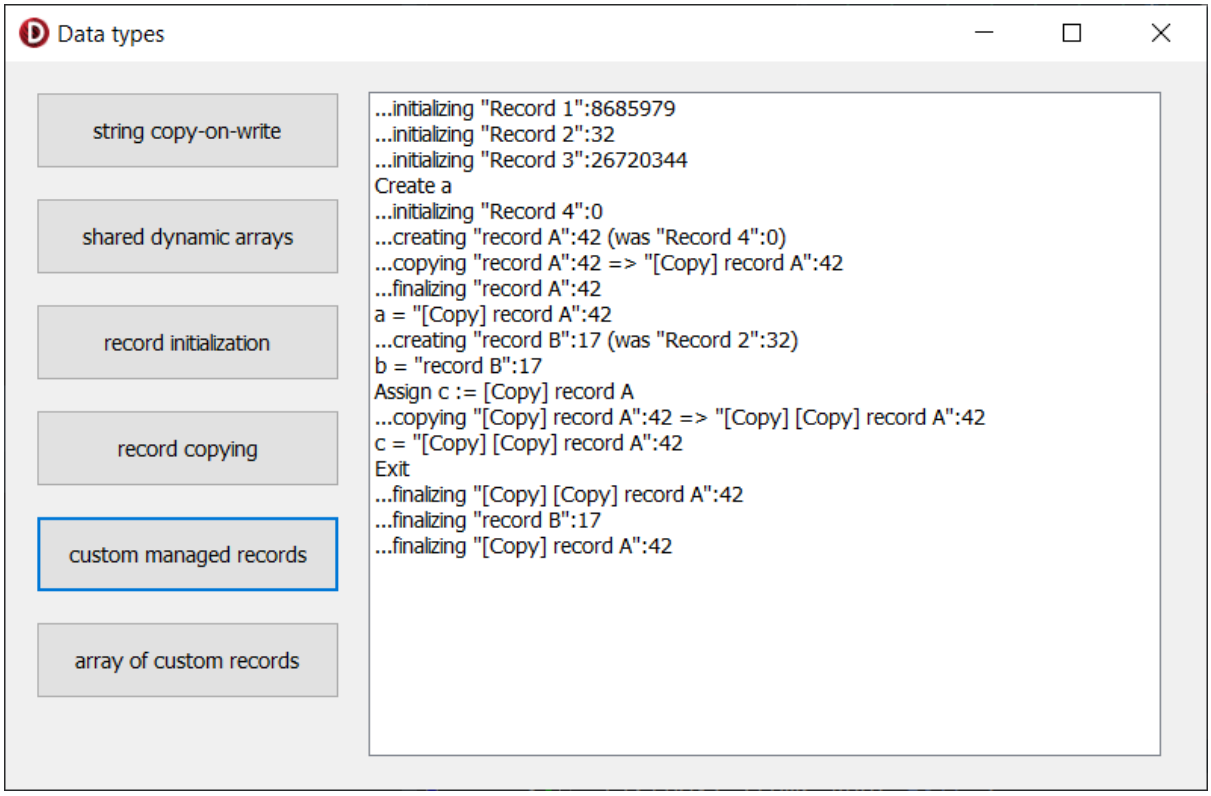
```
ExtractCommonExpressionMain.pas.75: s := ListBox1.Items[i];
005DF02B 8D4DF4            lea ecx,[ebp-$0c]
005DF02E 8B45FC            mov eax,[ebp-$04]
005DF031 8B80EC030000      mov eax,[eax+$000003ec]
005DF037 8B80C0020000      mov eax,[eax+$000002c0]
005DF03D 8B55F8            mov edx,[ebp-$08]
005DF040 8B18              mov ebx,[eax]
005DF042 FF530C            call dword ptr [ebx+$0c]
ExtractCommonExpressionMain.pas.76: p := Pos('-', s);
005DF045 B901000000        mov ecx,$00000001
005DF04A 8B55F4            mov edx,[ebp-$0c]
005DF04D B85CF15D00        mov eax,$005df15c
005DF052 E8C163E2FF        call Pos
005DF057 8945F0            mov [ebp-$10],eax
ExtractCommonExpressionMain.pas.77: ListBox1.Items[i] := Copy(s, p + 1,
005DF05A 8D45C8            lea eax,[ebp-$38]
005DF05D 50                push eax
005DF05E 8B45F4            mov eax,[ebp-$0c]
005DF061 E82EB8E2FF        call @UStrLen
005DF066 8BC8              mov ecx,eax
005DF068 8B55F0            mov edx,[ebp-$10]
005DF06B 42                inc edx
005DF06C 8B45F4            mov eax,[ebp-$0c]
005DF06F E868C1E2FF        call @UStrCopy
005DF074 FF75C8            push dword ptr [ebp-$38]
005DF077 685CF15D00        push $005df15c
005DF07C 8D45C4            lea eax,[ebp-$3c]
005DF07F 50                push eax
005DF080 8B4DF0            mov ecx,[ebp-$10]
005DF083 49                dec ecx
005DF084 BA01000000        mov edx,$00000001
005DF089 8B45F4            mov eax,[ebp-$0c]
005DF08C E84BC1E2FF        call @UStrCopy
005DF091 FF75C4            push dword ptr [ebp-$3c]
005DF094 8D45CC            lea eax,[ebp-$34]
005DF097 BA03000000        mov edx,$00000003
005DF09C E853C0E2FF        call @UStrCatN
005DF0A1 8B4DCC            mov ecx,[ebp-$34]
005DF0A4 8B45FC            mov eax,[ebp-$04]
005DF0A7 8B80EC030000      mov eax,[eax+$000003ec]
005DF0AD 8B80C0020000      mov eax,[eax+$000002c0]
005DF0B3 8B55F8            mov edx,[ebp-$08]
005DF0B6 8B18              mov ebx,[eax]
005DF0B8 FF5320            call dword ptr [ebx+$20]
```

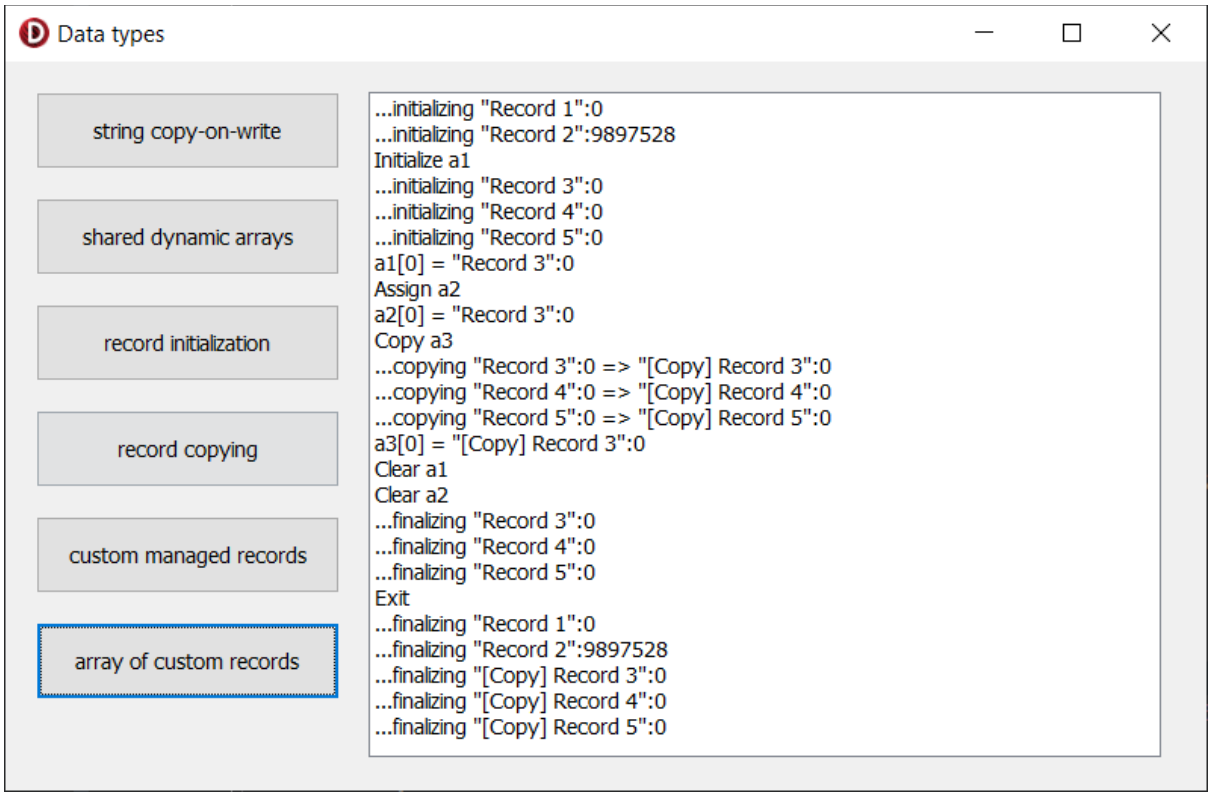




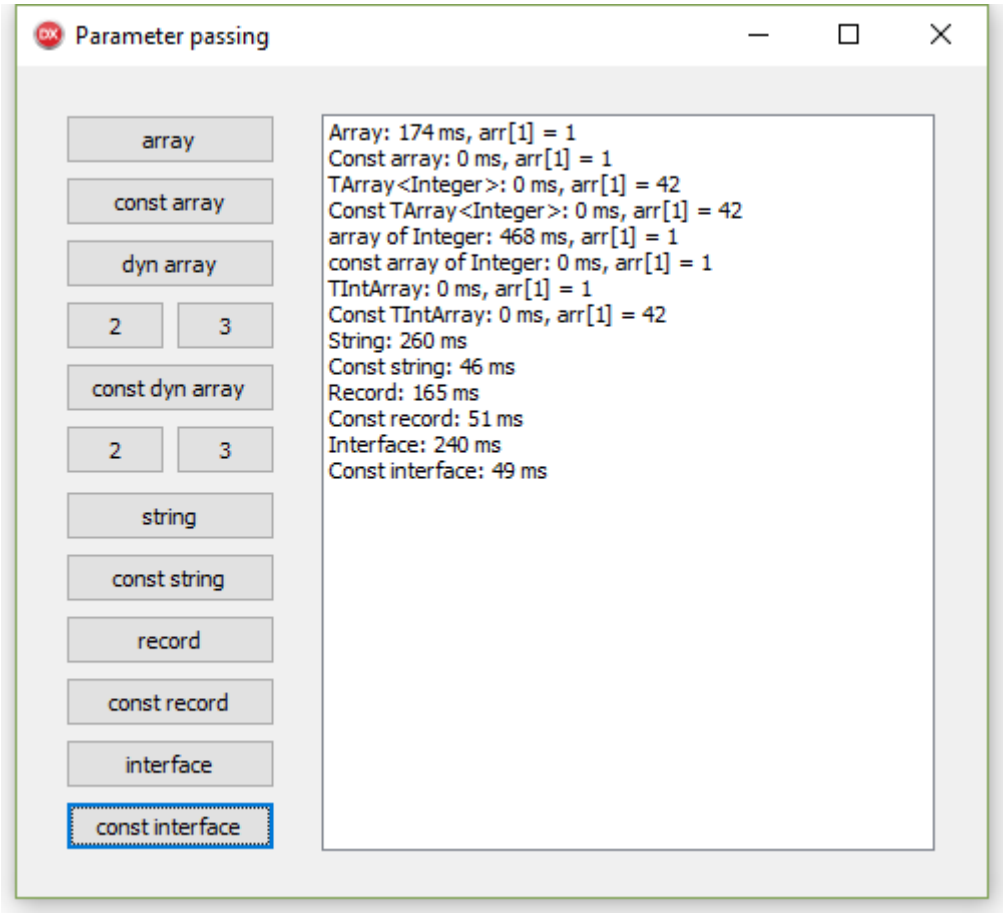

Data types
string copy-on-write
shared dynamic arrays
record initialization
a = 0, b = '', c = 4227471
a = 0, b = '', c = 0

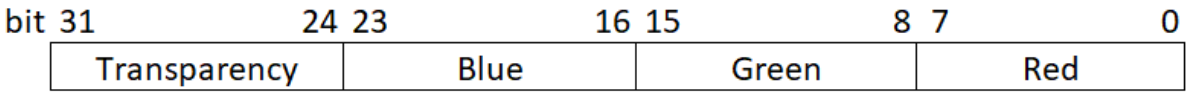

Data types
string copy-on-write
shared dynamic arrays
record initialization
record copying
TUnmanaged: 6 ms
TManaged: 31 ms

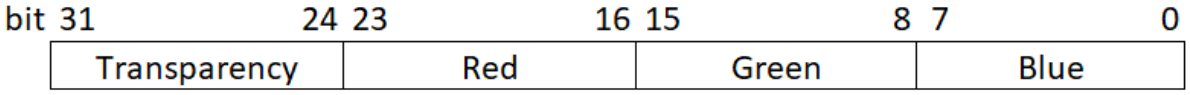

Data types
string copy-on-write
shared dynamic arrays
record initialization
record copying
custom managed records
array of custom records
...initializing "Record 1":8685979
...initializing "Record 2":32
...initializing "Record 3":26720344
Create a
...initializing "Record 4":0
...creating "record A":42 (was "Record 4":0)
...copying "record A":42 => "[Copy] record A":42
...finalizing "record A":42
a = "[Copy] record A":42
...creating "record B":17 (was "Record 2":32)
b = "record B":17
Assign c := [Copy] record A
...copying "[Copy] record A":42 => "[Copy] [Copy] record A":42
c = "[Copy] [Copy] record A":42
Exit
...finalizing "[Copy] [Copy] record A":42
...finalizing "record B":17
...finalizing "[Copy] record A":42

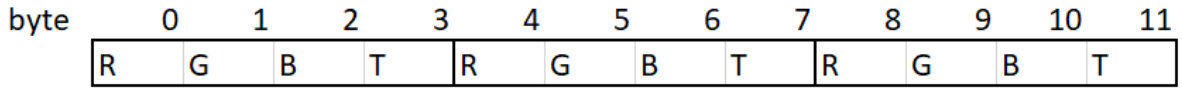

Data types
string copy-on-write
shared dynamic arrays
record initialization
record copying
custom managed records
array of custom records
...initializing "Record 1":0
...initializing "Record 2":9897528
Initialize a1
...initializing "Record 3":0
...initializing "Record 4":0
...initializing "Record 5":0
a1[0] = "Record 3":0
Assign a2
a2[0] = "Record 3":0
Copy a3
...copying "Record 3":0 => "[Copy] Record 3":0
...copying "Record 4":0 => "[Copy] Record 4":0
...copying "Record 5":0 => "[Copy] Record 5":0
a3[0] = "[Copy] Record 3":0
Clear a1
Clear a2
...finalizing "Record 3":0
...finalizing "Record 4":0
...finalizing "Record 5":0
Exit
...finalizing "Record 1":0
...finalizing "Record 2":9897528
...finalizing "[Copy] Record 3":0
...finalizing "[Copy] Record 4":0
...finalizing "[Copy] Record 5":0

```
ParameterPassingMain.pas.271: sum := SumArr(arr);
0109012C 8D55C8            lea edx,[ebp-$38]
0109012F B909000000        mov ecx,$00000009
01090134 8B45FC            mov eax,[ebp-$04]
01090137 E870FFFFFF        call $010900ac
0109013C 8945F4            mov [ebp-$0c],eax
ParameterPassingMain.pas.272: ListBox1.Items.Add(IntToStr(sum));
0109013F 8D55C4            lea edx,[ebp-$3c]
01090142 8B45F4            mov eax,[ebp-$0c]
01090145 E8AE4FE4FF        call $00ed50f8
0109014A 8B55C4            mov edx,[ebp-$3c]
0109014D 8B45FC            mov eax,[ebp-$04]
01090150 8B80F0030000      mov eax,[eax+$000003f0]
01090156 8B80C0020000      mov eax,[eax+$000002c0]
0109015C 8B08              mov ecx,[eax]
0109015E FF513C            call dword ptr [ecx+$3c]
ParameterPassingMain.pas.273: sum := SumArr(Slice(arr, 3));
01090161 8D55C8            lea edx,[ebp-$38]
01090164 B902000000        mov ecx,$00000002
01090169 8B45FC            mov eax,[ebp-$04]
0109016C E83BFFFFFF        call $010900ac
01090171 8945F4            mov [ebp-$0c],eax
```

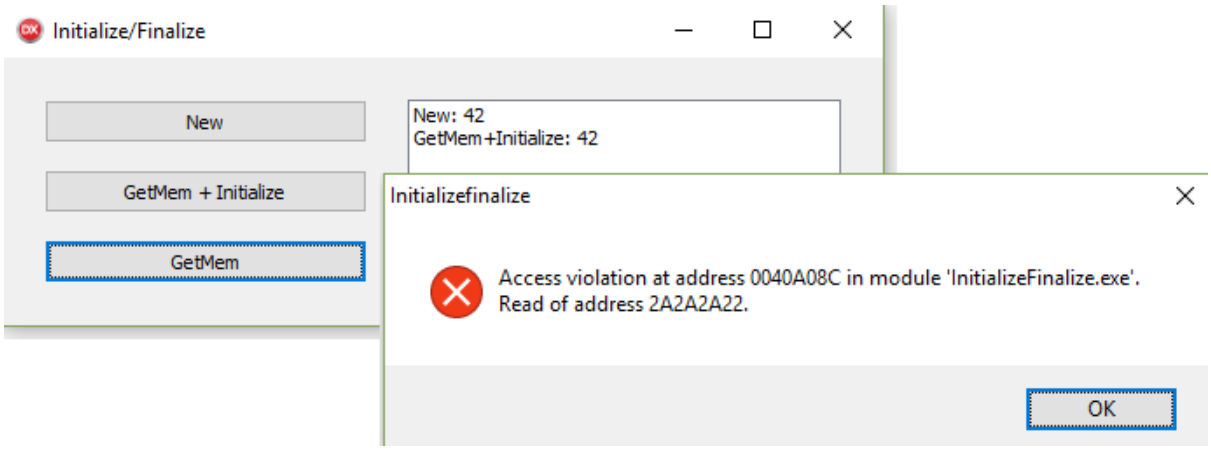


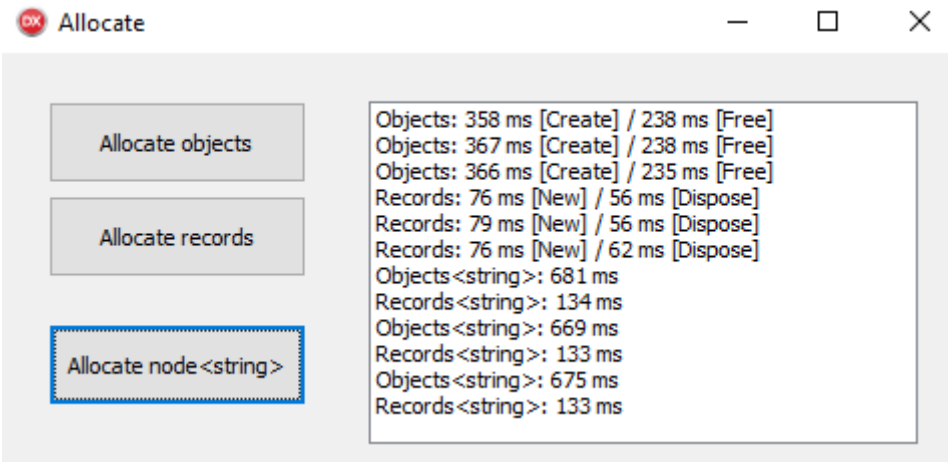

bit 31
24 23
16 15
8 7
0
Transparency
Blue
Green
Red


bit 31
24 23
16 15
8 7
0
Transparency
Red
Green
Blue


byte
0
1
2
3
4
5
6
7
8
9
10
11
R
G
B
T
R
G
B
T
R
G
B
T

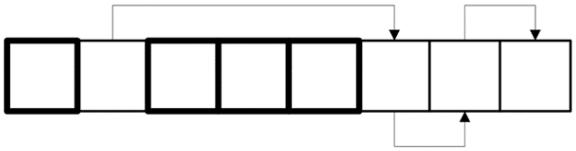
Chapter 6: Memory Management



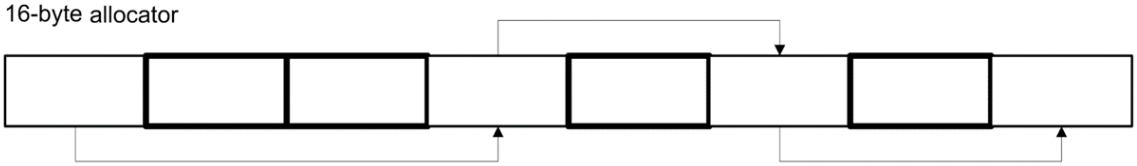


Small block allocators

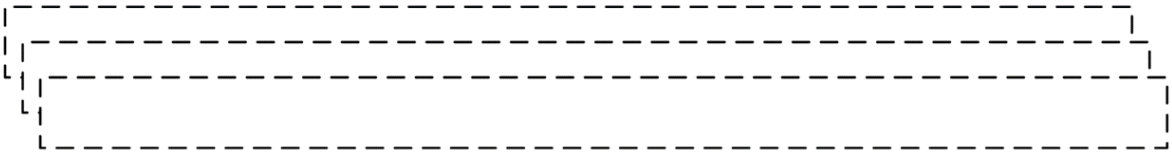
8-byte allocator



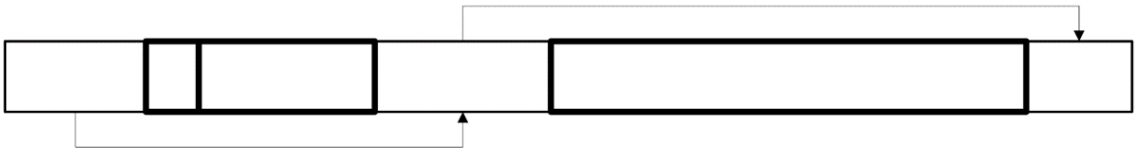
16-byte allocator



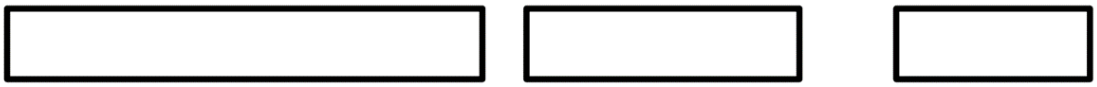
other allocators

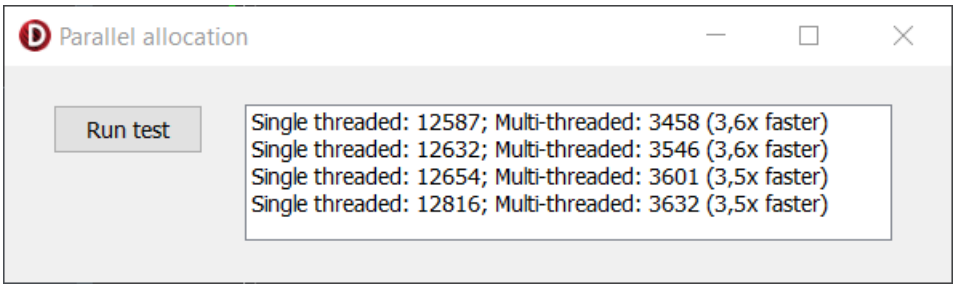


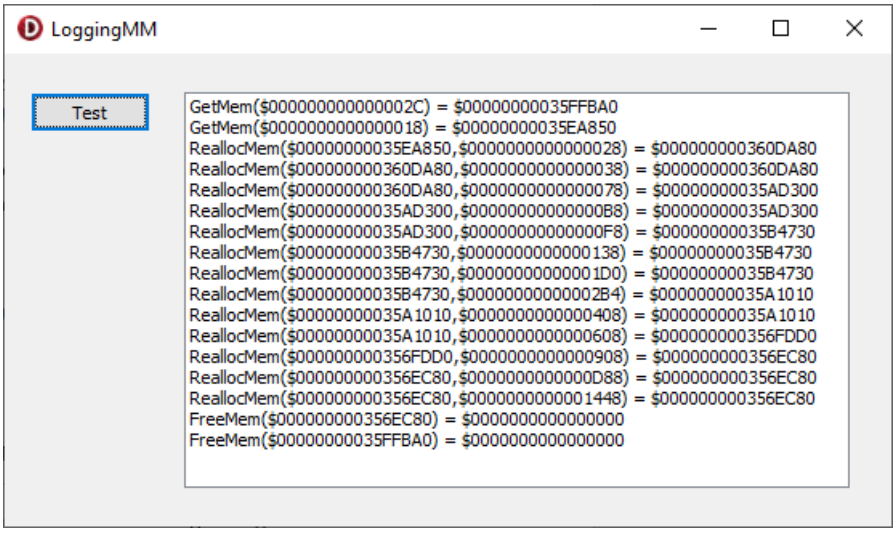
Medium block allocator



Large block allocator





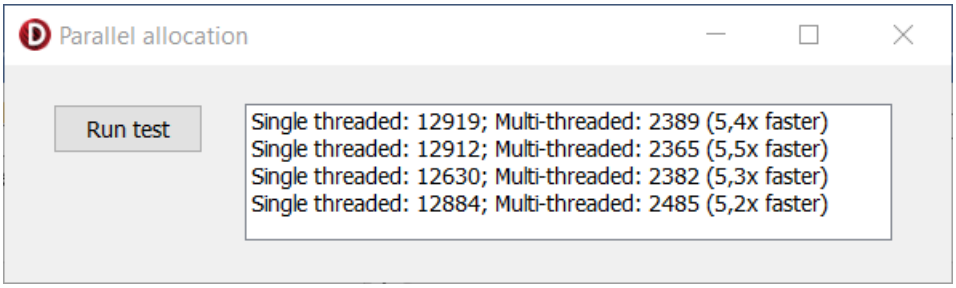


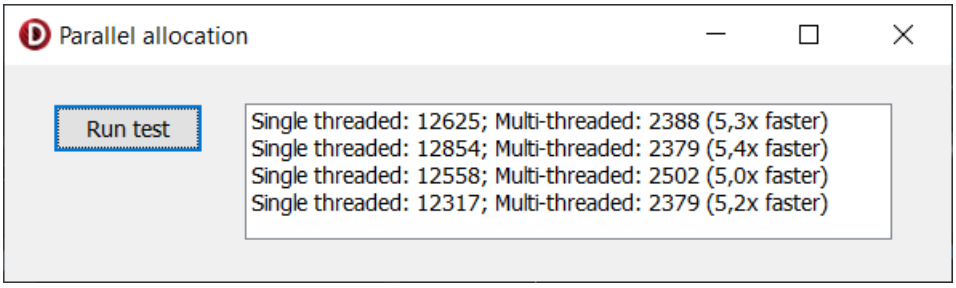
ParallelAllocation.exe: Locking Report

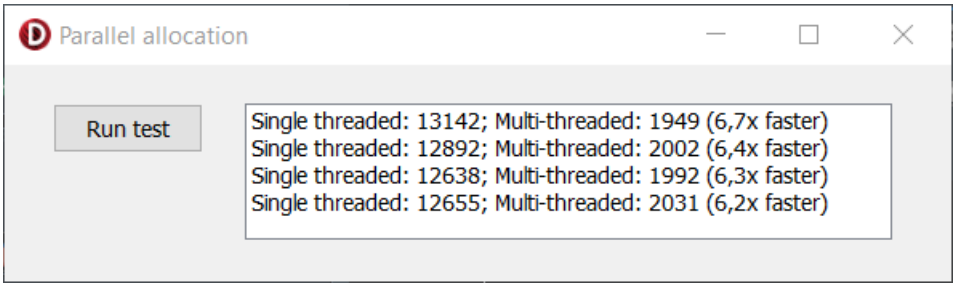
Top locking contention sites

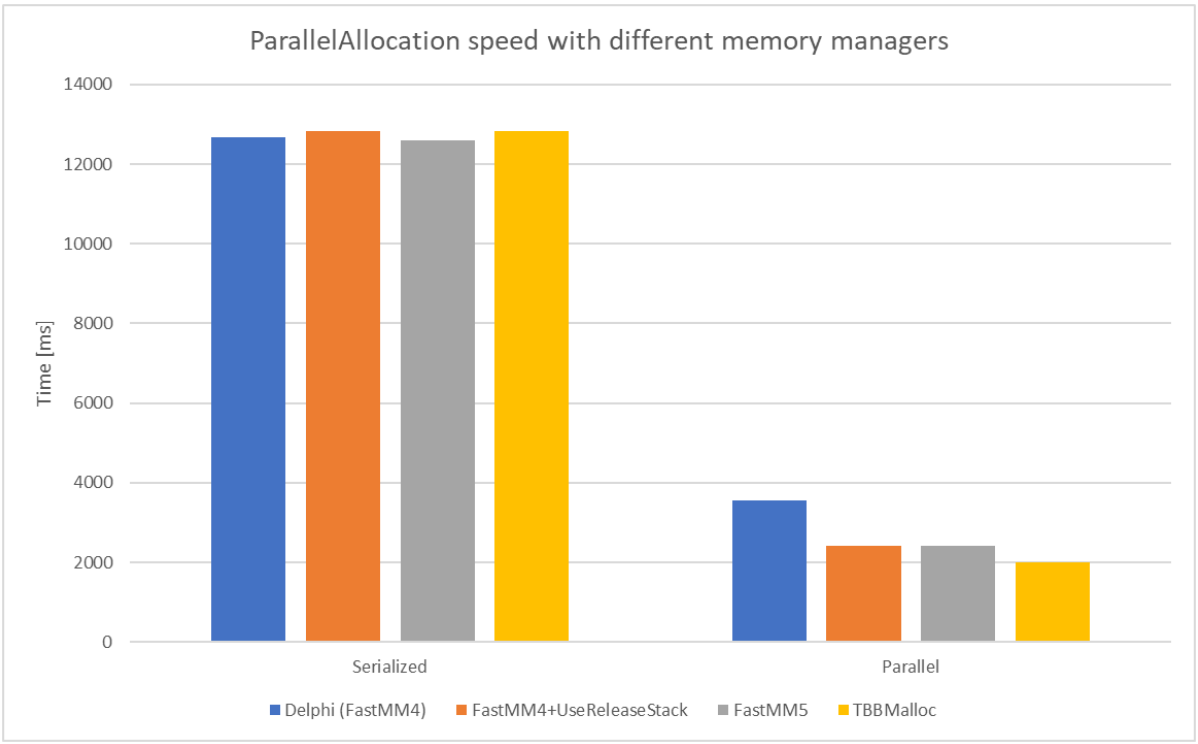
19020 x

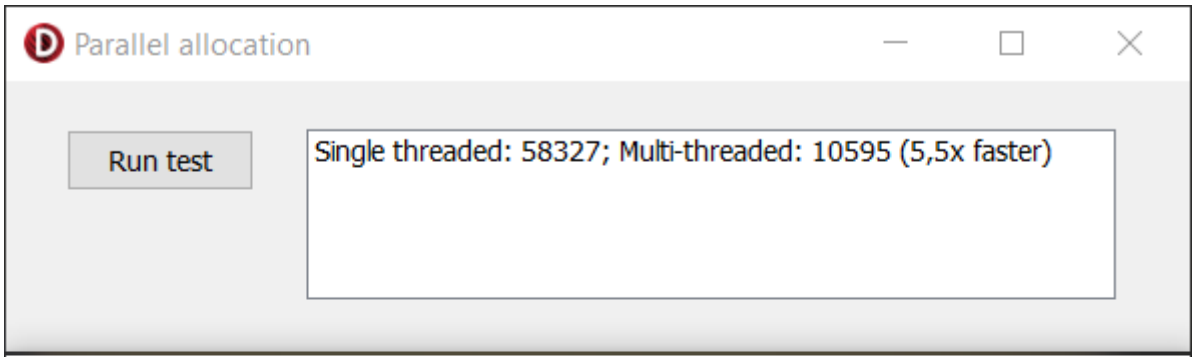
407092 [System.pas][System][@FreeMem$qqrpv][4662]
5FB1A0
[System.Generics.Collections.pas][ParallelAllocationMain][Generics.Collections.%TList__1$p27Parallelallocationmain.TFoo%.InternalNotify$qqrpxv51System.Generics.Collections.TCollectionNotification][4290]
43FE36 [System.Generics.Collections.pas][System.Generics.Collections][Generics.Collections.TListHelper.InternalDoDelete4$qqri51System.Generics.Collections.TCollectionNotification][2963]
5FABC3 [ParallelAllocationMain.pas][ParallelAllocationMain] TSpeedTest.Execute_3_ActRec._0_Body$qqri[130]
5E2B44 [System.Threading.pas][System.Threading][Threading.TParallel.ForWorker_45_ActRec._0_Body$qqrv][1339]
5E436C [System.Threading.pas][System.Threading][Threading.TTask.ExecuteReplicates_116_ActRec._0_Body$qqrv][2046]
5E40B0 [System.Threading.pas][System.Threading][Threading.TTask.Execute$qqrv][2021]
5E4D6E [System.Threading.pas][System.Threading][Threading.TTask.InternalExecute$qqrrp22System.Threading.TTask][2356]
5E4E73 [System.Threading.pas][System.Threading][Threading.TTask.InternalWork$qqro][2383]
5E6DB8
[System.Threading.pas][System.Threading][Threading.TThreadPool.TQueueWorkerThread.ExecuteWorkItem$qqrr75System.%DelphiInterface$48System.Threading.TThreadPool.IThreadPoolWorkItem%][3236]
5E6A10 [System.Threading.pas][System.Threading][Threading.TThreadPool.TQueueWorkerThread.Execute$qqrv][3165]

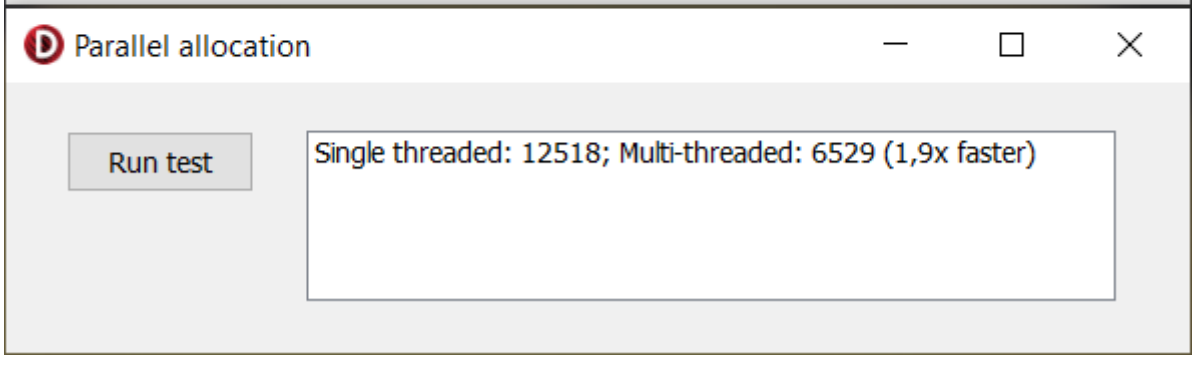




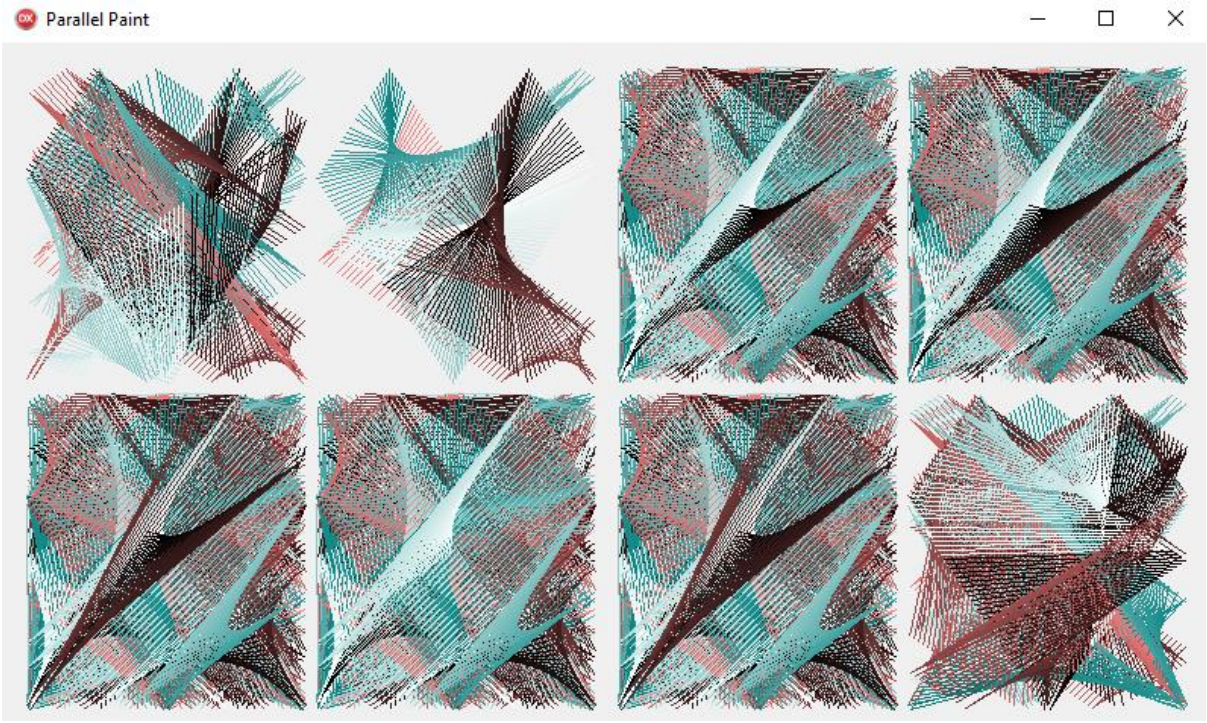

Parallel allocation
Run test
Single threaded: 13142; Multi-threaded: 1949 (6,7x faster)
Single threaded: 12892; Multi-threaded: 2002 (6,4x faster)
Single threaded: 12638; Multi-threaded: 1992 (6,3x faster)
Single threaded: 12655; Multi-threaded: 2031 (6,2x faster)

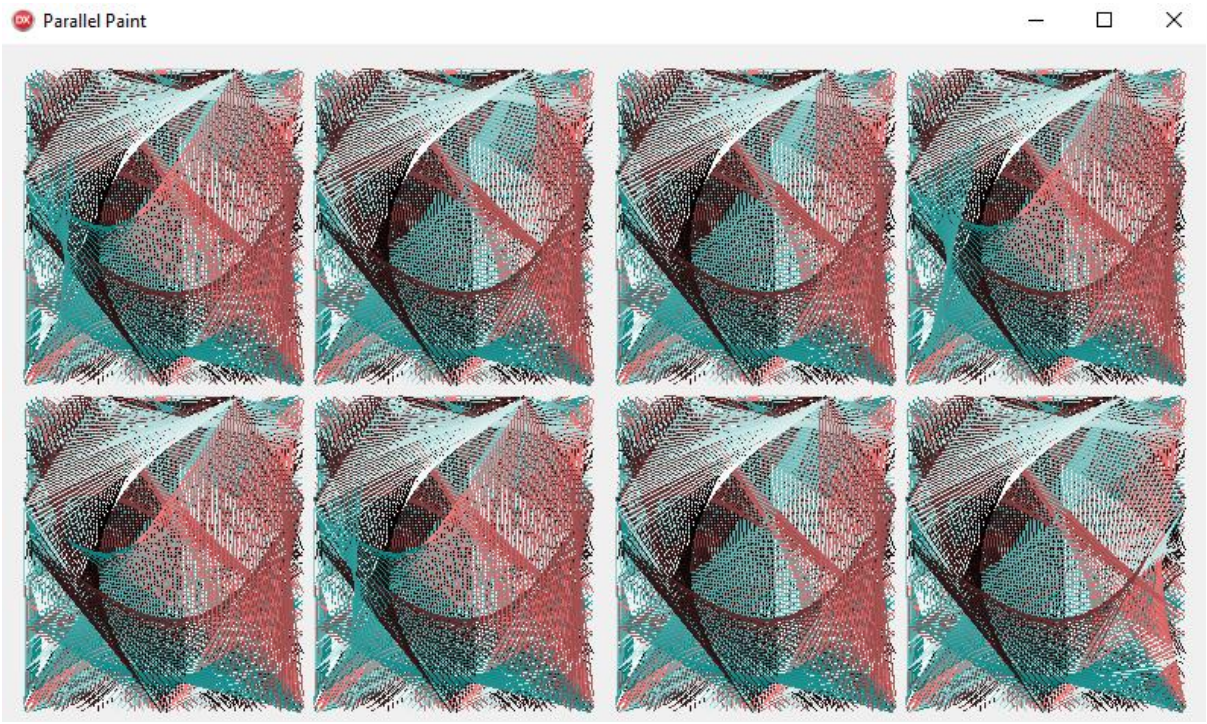

ParallelAllocation speed with different memory managers
14000
12000
10000
8000
6000
4000
2000
0
Time [ms]
Serialized
Parallel
Delphi (FastMM4)
FastMM4+UseReleaseStack
FastMM5
TBBMalloc

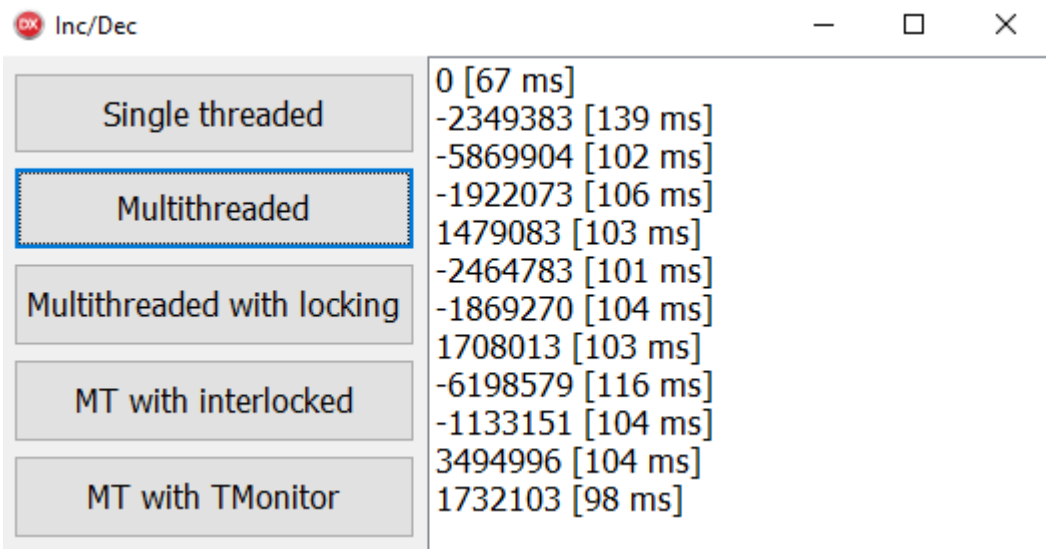

Parallel allocation
Run test
Single threaded: 58327; Multi-threaded: 10595 (5,5x faster)

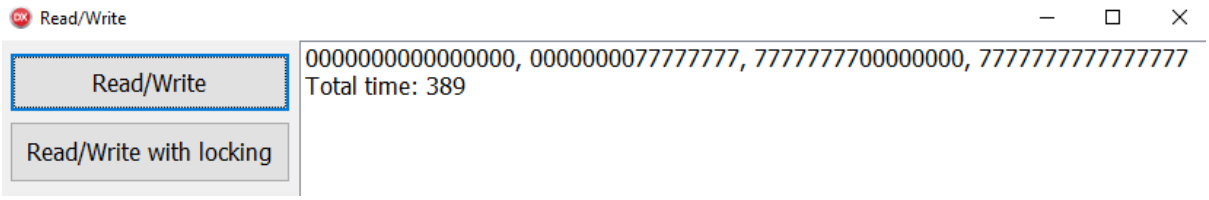

Parallel allocation
Run test
Single threaded: 12518; Multi-threaded: 6529 (1,9x faster)

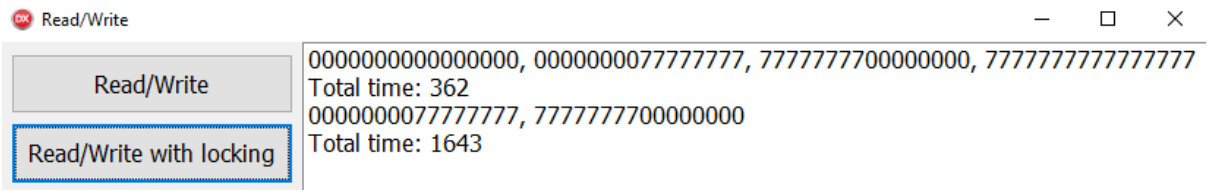
Chapter 7: Getting Started with the Parallel World

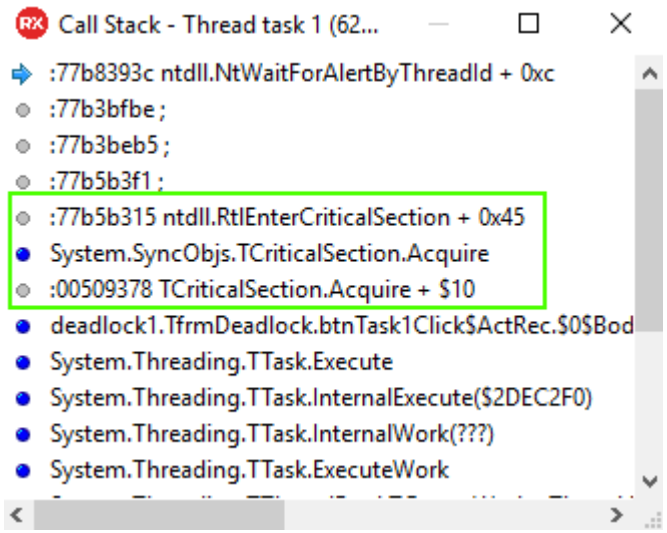


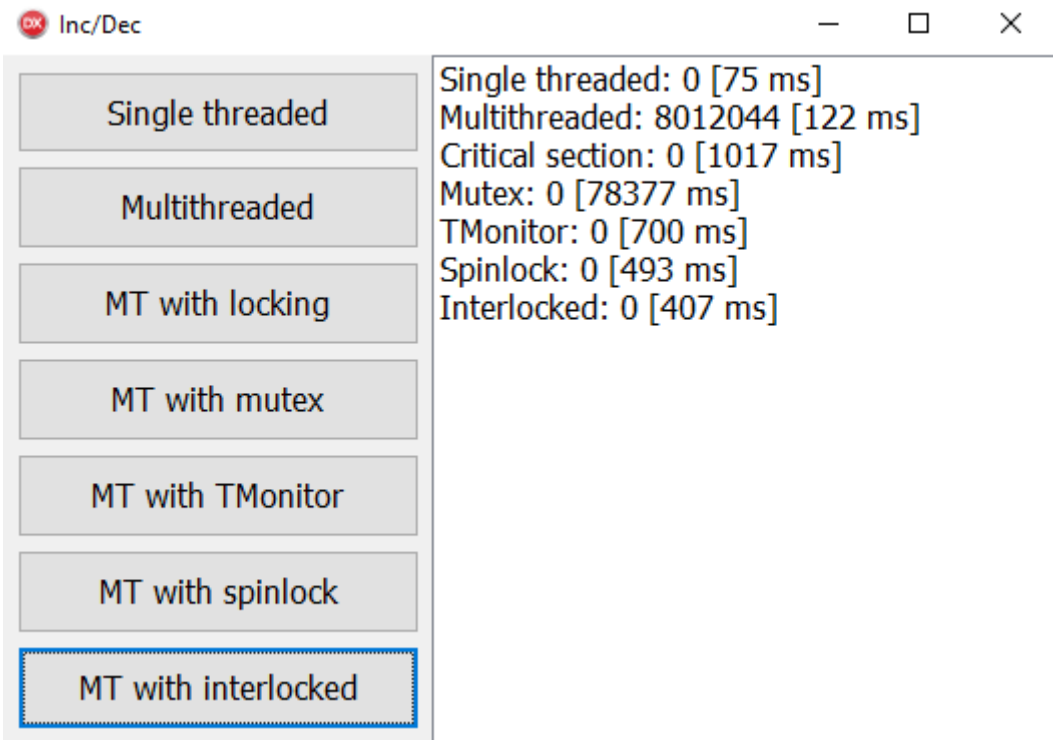


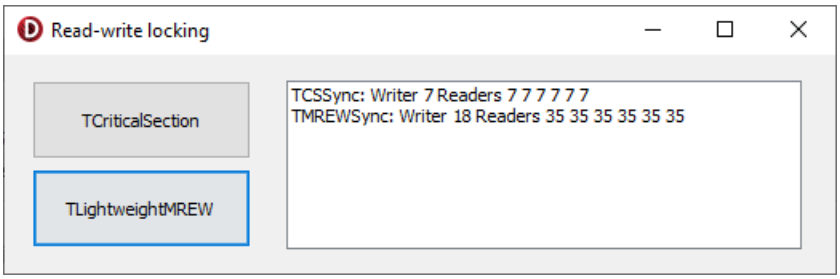

Inc/Dec
Single threaded
Multithreaded
Multithreaded with locking
MT with interlocked
MT with TMonitor
0 [67 ms]
-2349383 [139 ms]
-5869904 [102 ms]
-1922073 [106 ms]
1479083 [103 ms]
-2464783 [101 ms]
-1869270 [104 ms]
1708013 [103 ms]
-6198579 [116 ms]
-1133151 [104 ms]
3494996 [104 ms]
1732103 [98 ms]

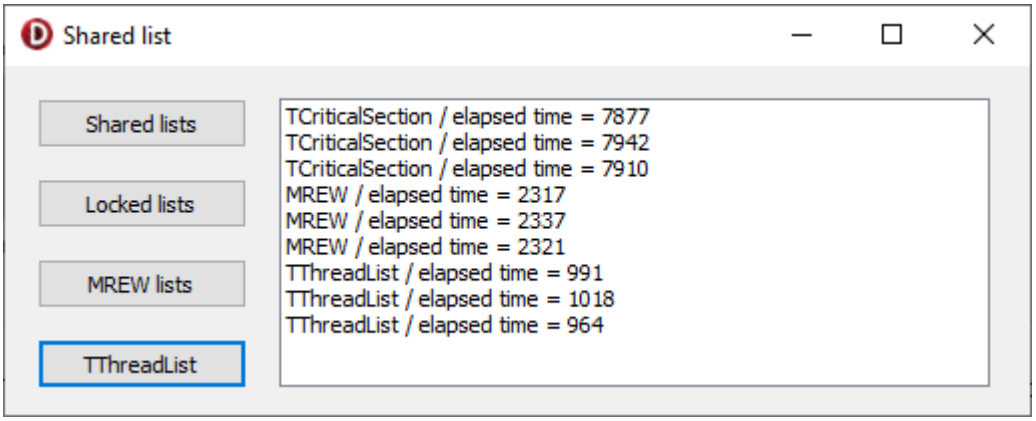

Read/Write
Read/Write
Read/Write with locking
0000000000000000, 0000000077777777, 7777777700000000, 7777777777777777
Total time: 389

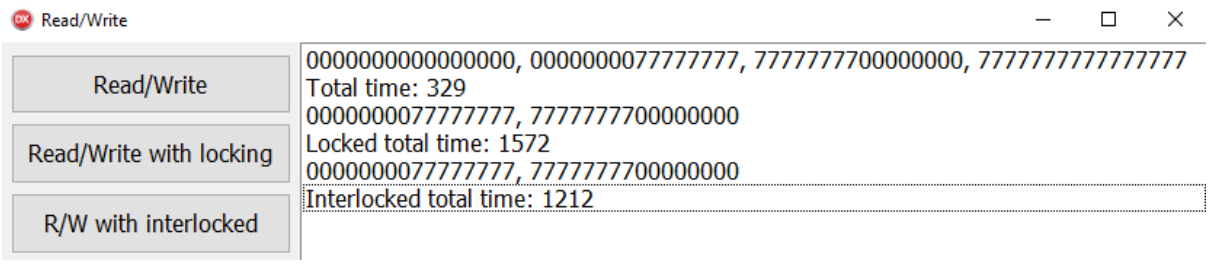

Read/Write
Read/Write
Read/Write with locking
0000000000000000, 0000000077777777, 7777777700000000, 7777777777777777
Total time: 362
0000000077777777, 7777777700000000
Total time: 1643

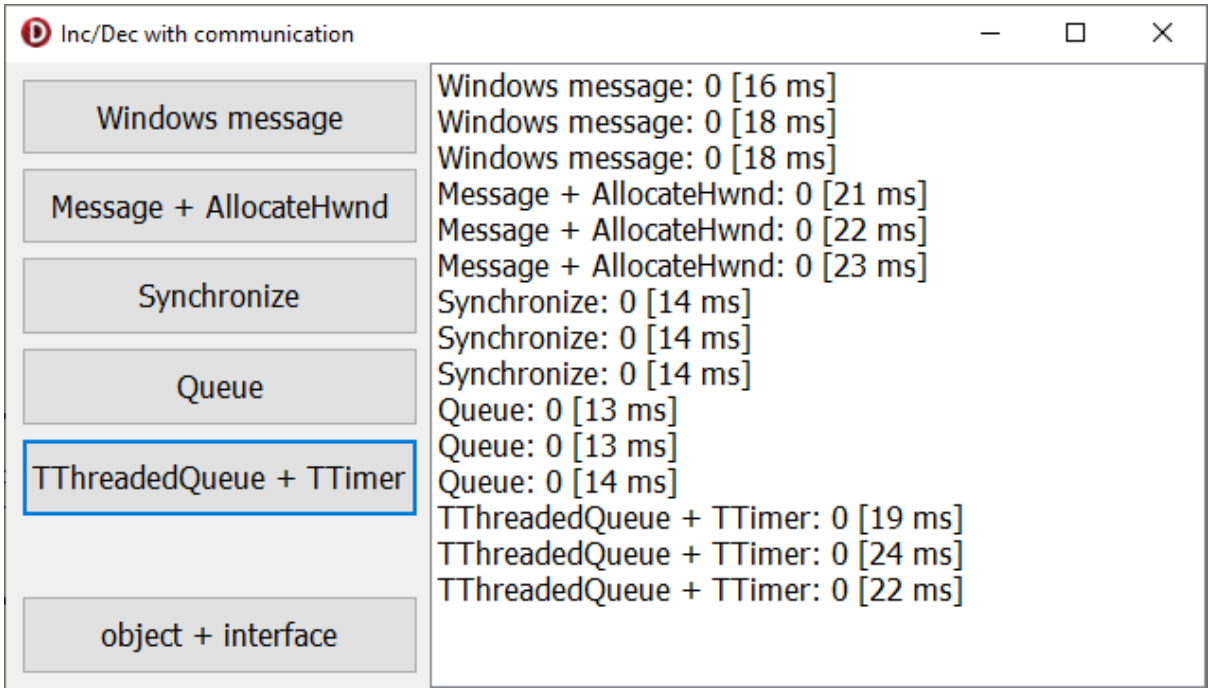

Call Stack - Thread task 1 (62...
:77b8393c ntdll.NtWaitForAlertByThreadId + 0xc
:77b3bfbe ;
:77b3beb5 ;
:77b5b3f1 ;
:77b5b315 ntdll.RtlEnterCriticalSection + 0x45
System.SyncObjs.TCriticalSection.Acquire
:00509378 TCriticalSection.Acquire + $10
deadlock1.TfrmDeadlock.btnTask1Click$ActRec.$0$Bod
System.Threading.TTask.Execute
System.Threading.TTask.InternalExecute($2DEC2F0)
System.Threading.TTask.InternalWork(???)
System.Threading.TTask.ExecuteWork


Inc/Dec
Single threaded
Multithreaded
MT with locking
MT with mutex
MT with TMonitor
MT with spinlock
MT with interlocked
Single threaded: 0 [75 ms]
Multithreaded: 8012044 [122 ms]
Critical section: 0 [1017 ms]
Mutex: 0 [78377 ms]
TMonitor: 0 [700 ms]
Spinlock: 0 [493 ms]
Interlocked: 0 [407 ms]

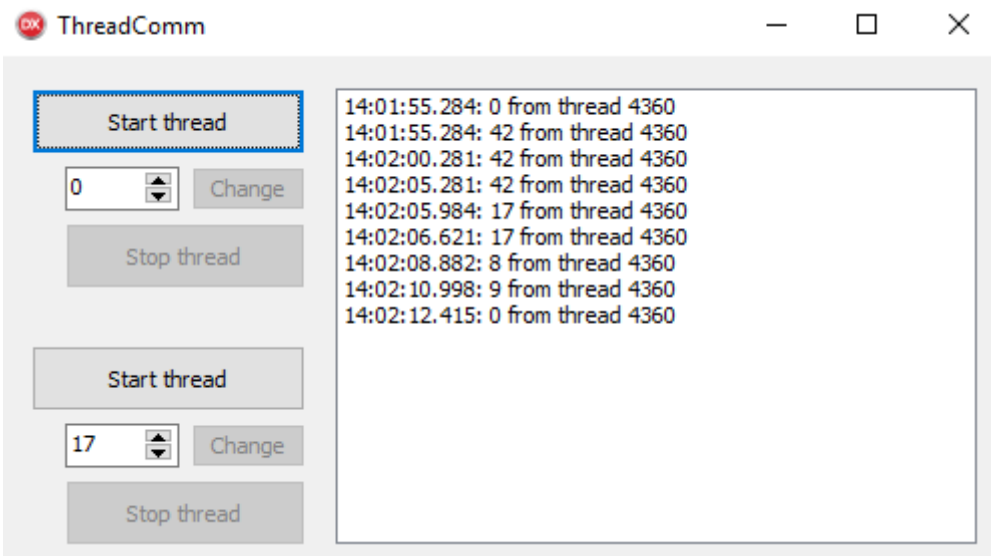

Read-write locking
TCriticalSection
TLightweightMREW
TCSSync: Writer 7 Readers 7 7 7 7 7
TMREWSync: Writer 18 Readers 35 35 35 35 35 35


Shared list
Shared lists
Locked lists
MREW lists
TThreadList
TCriticalSection / elapsed time = 7877
TCriticalSection / elapsed time = 7942
TCriticalSection / elapsed time = 7910
MREW / elapsed time = 2317
MREW / elapsed time = 2337
MREW / elapsed time = 2321
TThreadList / elapsed time = 991
TThreadList / elapsed time = 1018
TThreadList / elapsed time = 964


Read/Write
Read/Write
Read/Write with locking
R/W with interlocked
0000000000000000, 0000000077777777, 7777777700000000, 7777777777777777
Total time: 329
0000000077777777, 7777777700000000
Locked total time: 1572
0000000077777777, 7777777700000000
Interlocked total time: 1212

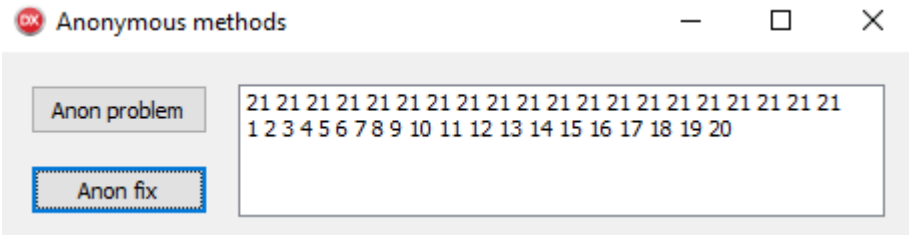

Inc/Dec with communication
Windows message
Message + AllocateHwnd
Synchronize
Queue
TThreadedQueue + TTimer
object + interface
Windows message: 0 [16 ms]
Windows message: 0 [18 ms]
Windows message: 0 [18 ms]
Message + AllocateHwnd: 0 [21 ms]
Message + AllocateHwnd: 0 [22 ms]
Message + AllocateHwnd: 0 [23 ms]
Synchronize: 0 [14 ms]
Synchronize: 0 [14 ms]
Synchronize: 0 [14 ms]
Queue: 0 [13 ms]
Queue: 0 [13 ms]
Queue: 0 [14 ms]
TThreadedQueue + TTimer: 0 [19 ms]
TThreadedQueue + TTimer: 0 [24 ms]
TThreadedQueue + TTimer: 0 [22 ms]

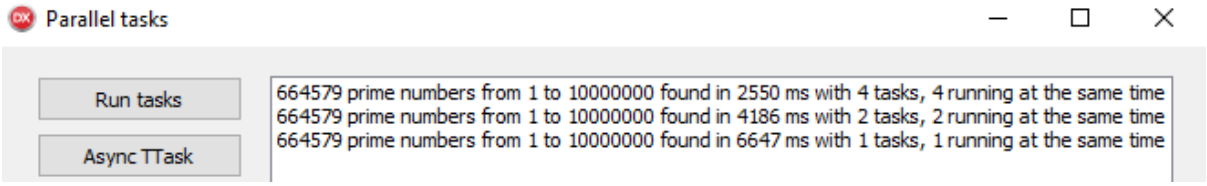
Chapter 8: Working with Parallel Tools

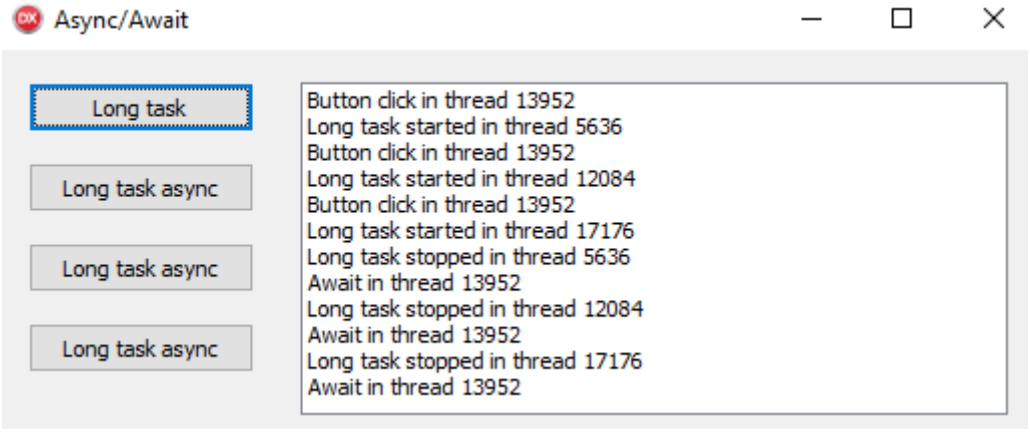


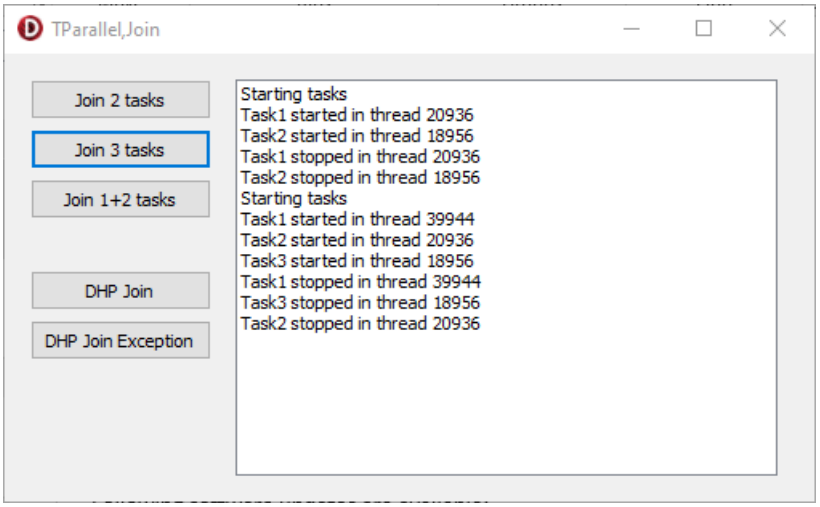


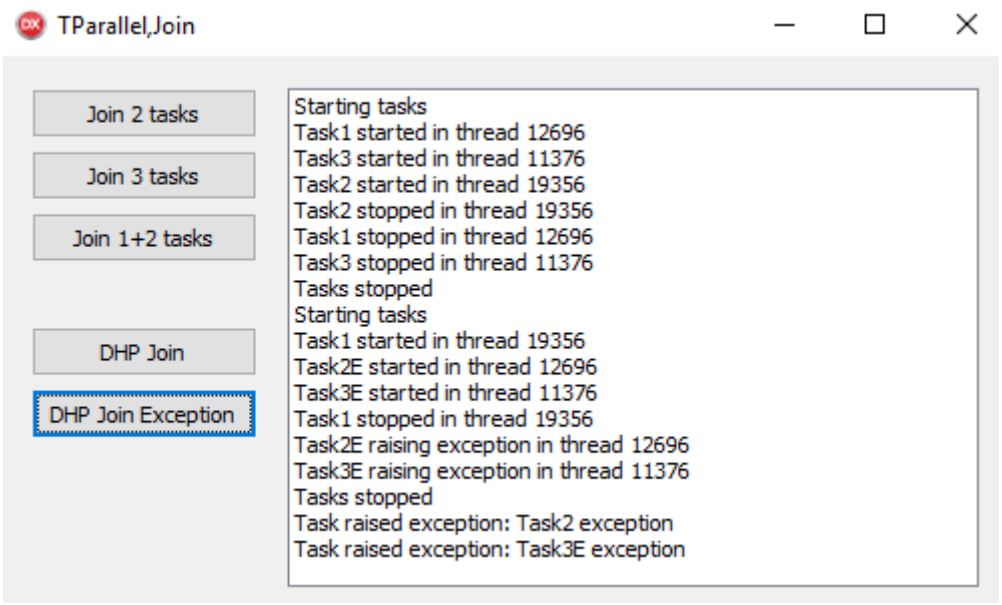
Chapter 9: Exploring Parallel Practices

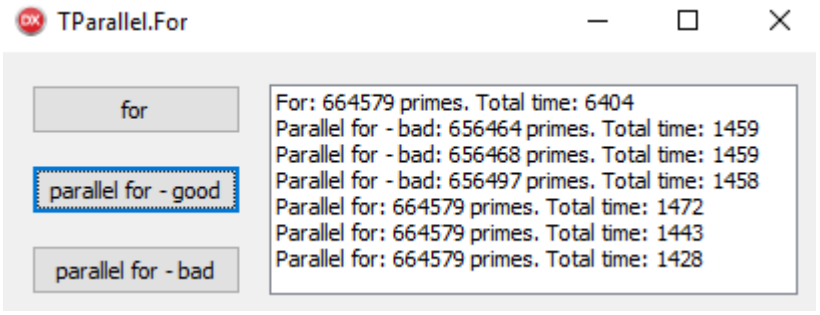


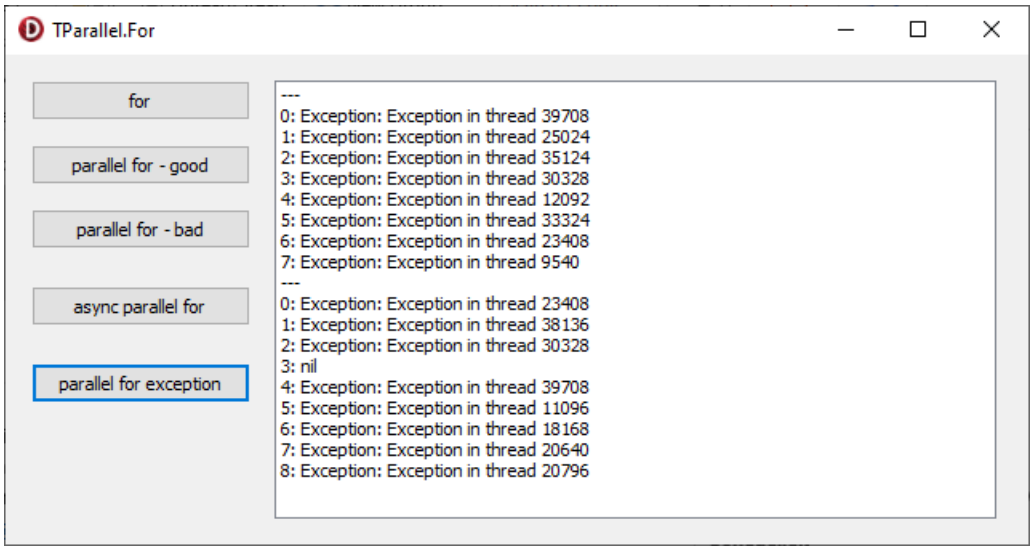






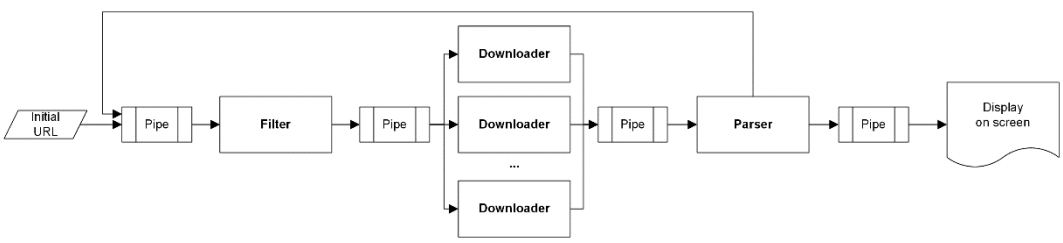

TParallel.Join
Join 2 tasks
Join 3 tasks
Join 1+2 tasks
DHP Join
DHP Join Exception
Starting tasks
Task1 started in thread 12696
Task3 started in thread 11376
Task2 started in thread 19356
Task2 stopped in thread 19356
Task1 stopped in thread 12696
Task3 stopped in thread 11376
Tasks stopped
Starting tasks
Task1 started in thread 19356
Task2E started in thread 12696
Task3E started in thread 11376
Task1 stopped in thread 19356
Task2E raising exception in thread 12696
Task3E raising exception in thread 11376
Tasks stopped
Task raised exception: Task2 exception
Task raised exception: Task3E exception

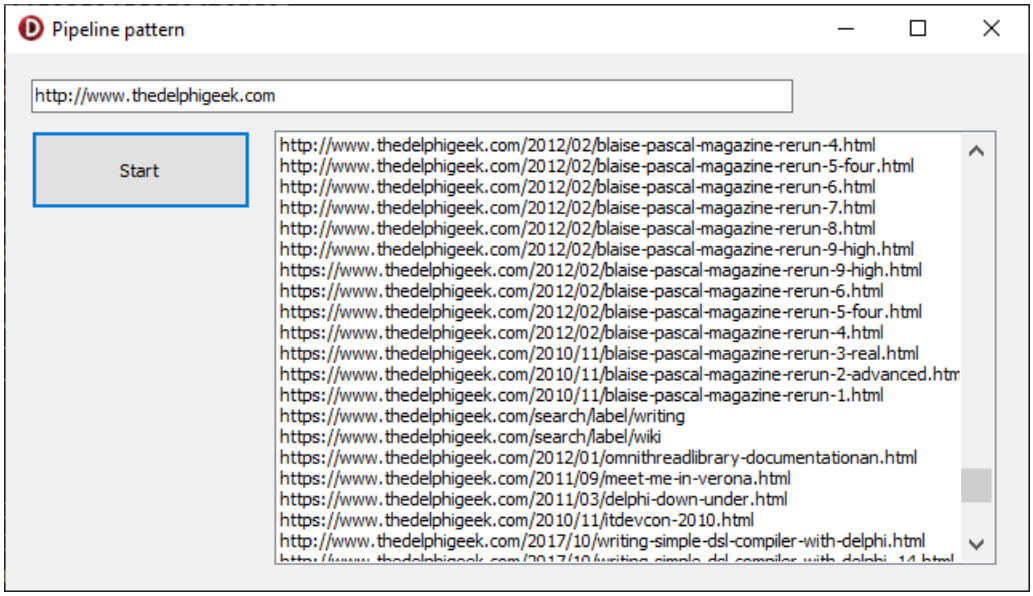

TParallel.For
for
parallel for - good
parallel for - bad
For: 664579 primes. Total time: 6404
Parallel for - bad: 656464 primes. Total time: 1459
Parallel for - bad: 656468 primes. Total time: 1459
Parallel for - bad: 656497 primes. Total time: 1458
Parallel for: 664579 primes. Total time: 1472
Parallel for: 664579 primes. Total time: 1443
Parallel for: 664579 primes. Total time: 1428


TParallel.For
for
parallel for - good
parallel for - bad
async parallel for
parallel for exception

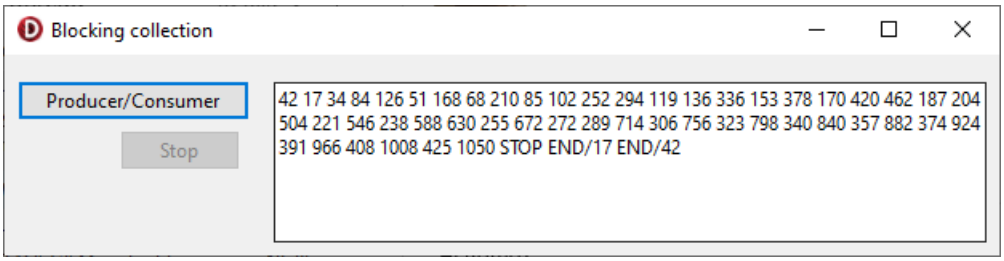
0: Exception: Exception in thread 39708
1: Exception: Exception in thread 25024
2: Exception: Exception in thread 35124
3: Exception: Exception in thread 30328
4: Exception: Exception in thread 12092
5: Exception: Exception in thread 33324
6: Exception: Exception in thread 23408
7: Exception: Exception in thread 9540

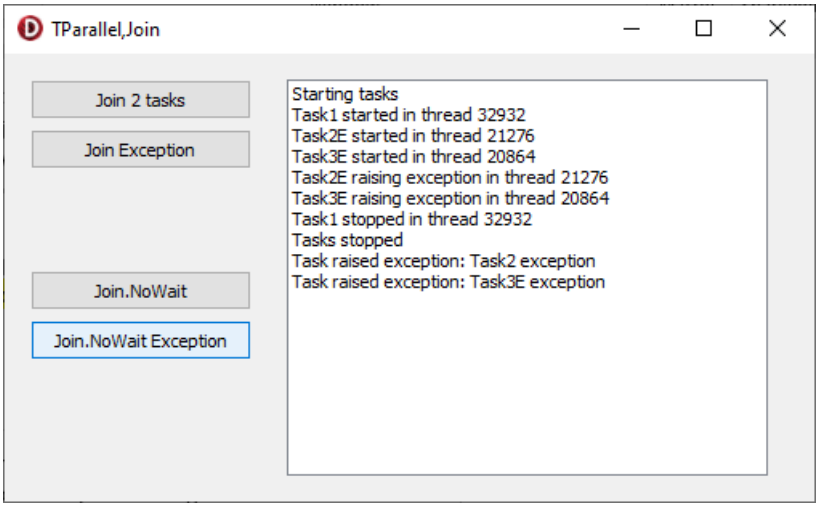
0: Exception: Exception in thread 23408
1: Exception: Exception in thread 38136
2: Exception: Exception in thread 30328
3: nil
4: Exception: Exception in thread 39708
5: Exception: Exception in thread 11096
6: Exception: Exception in thread 18168
7: Exception: Exception in thread 20640
8: Exception: Exception in thread 20796

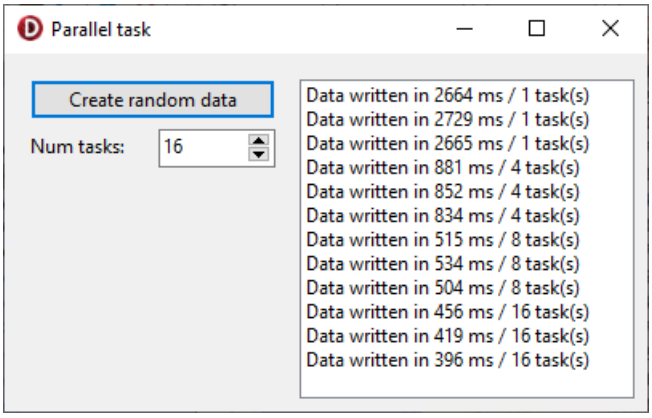

Initial URL
Pipe
Filter
Pipe
Downloader
Downloader
...
Downloader
Pipe
Parser
Pipe
Display on screen

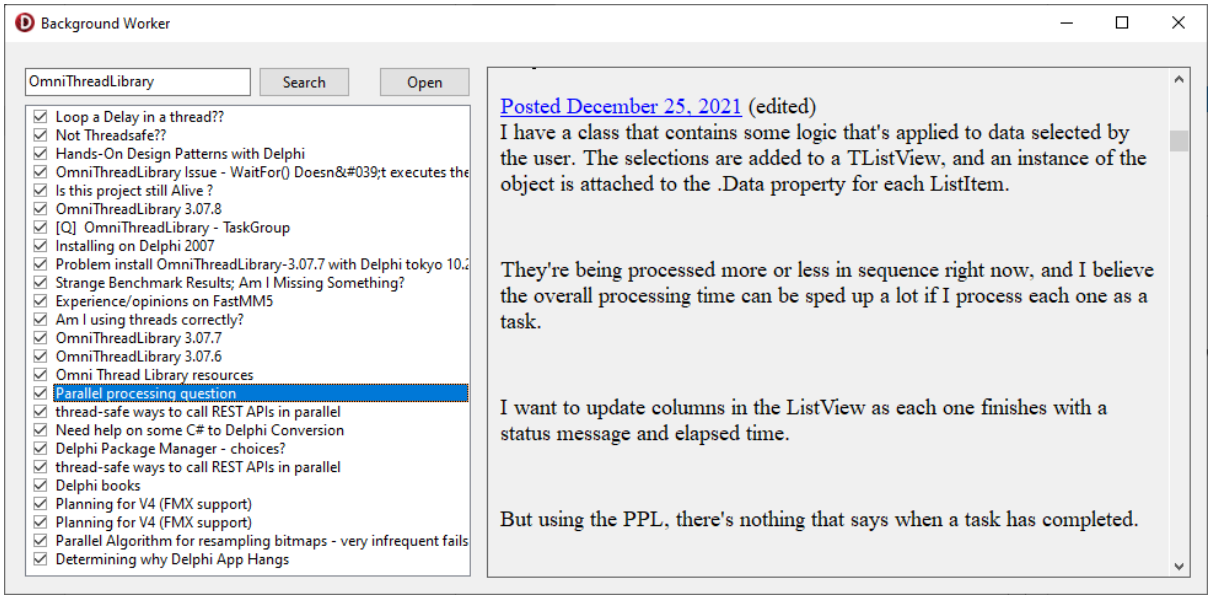

Pipeline pattern
http://www.thedelphigeek.com
Start
http://www.thedelphigeek.com/2012/02/blaise-pascal-magazine-rerun-4.html
http://www.thedelphigeek.com/2012/02/blaise-pascal-magazine-rerun-5-four.html
http://www.thedelphigeek.com/2012/02/blaise-pascal-magazine-rerun-6.html
http://www.thedelphigeek.com/2012/02/blaise-pascal-magazine-rerun-7.html
http://www.thedelphigeek.com/2012/02/blaise-pascal-magazine-rerun-8.html
http://www.thedelphigeek.com/2012/02/blaise-pascal-magazine-rerun-9-high.html
https://www.thedelphigeek.com/2012/02/blaise-pascal-magazine-rerun-9-high.html
https://www.thedelphigeek.com/2012/02/blaise-pascal-magazine-rerun-6.html
https://www.thedelphigeek.com/2012/02/blaise-pascal-magazine-rerun-5-four.html
https://www.thedelphigeek.com/2012/02/blaise-pascal-magazine-rerun-4.html
https://www.thedelphigeek.com/2010/11/blaise-pascal-magazine-rerun-3-real.html
https://www.thedelphigeek.com/2010/11/blaise-pascal-magazine-rerun-2-advanced.htm
https://www.thedelphigeek.com/2010/11/blaise-pascal-magazine-rerun-1.html
https://www.thedelphigeek.com/search/label/writing
https://www.thedelphigeek.com/search/label/wiki
https://www.thedelphigeek.com/2012/01/omnithreadlibrary-documentationan.html
https://www.thedelphigeek.com/2011/09/meet-me-in-verona.html
https://www.thedelphigeek.com/2011/03/delphi-down-under.html
https://www.thedelphigeek.com/2010/11/itdevcon-2010.html
http://www.thedelphigeek.com/2017/10/writing-simple-dsl-compiler-with-delphi.html

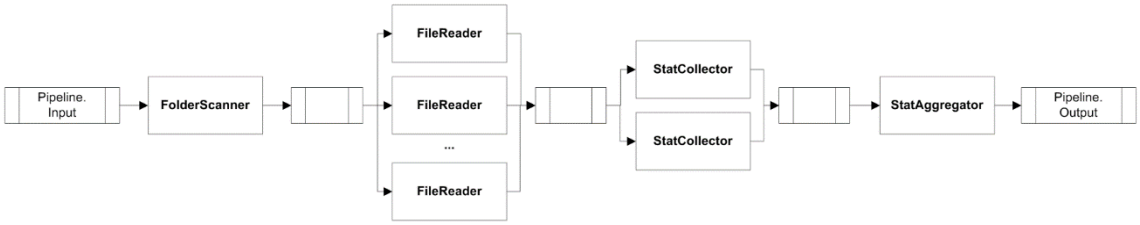
Chapter 10: More Parallel Patterns

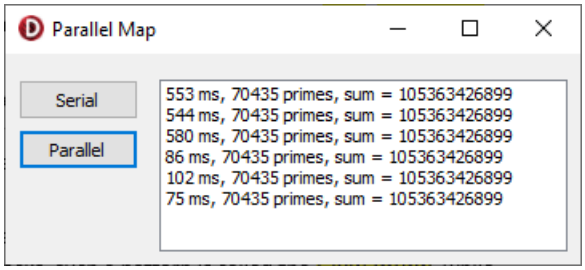


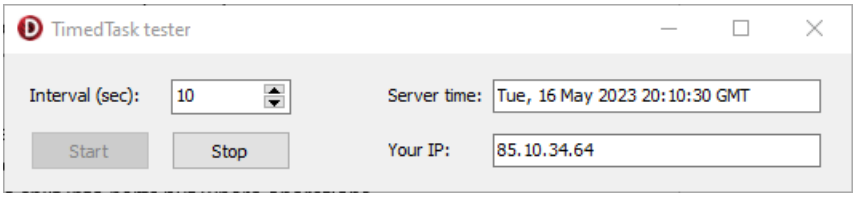





Background Worker
OmniThreadLibrary
Search
Open
Loop a Delay in a thread??
Not Threadsafe??
Hands-On Design Patterns with Delphi
OmniThreadLibrary Issue - WaitFor() Doesn't executes the
Is this project still Alive ?
OmniThreadLibrary 3.07.8
[Q] OmniThreadLibrary - TaskGroup
Installing on Delphi 2007
Problem install OmniThreadLibrary-3.07.7 with Delphi tokyo 10.2
Strange Benchmark Results; Am I Missing Something?
Experience/opinions on FastMM5
Am I using threads correctly?
OmniThreadLibrary 3.07.7
OmniThreadLibrary 3.07.6
Omni Thread Library resources
Parallel processing question
thread-safe ways to call REST APIs in parallel
Need help on some C# to Delphi Conversion
Delphi Package Manager - choices?
thread-safe ways to call REST APIs in parallel
Delphi books
Planning for V4 (FMX support)
Planning for V4 (FMX support)
Parallel Algorithm for resampling bitmaps - very infrequent fails
Determining why Delphi App Hangs
Posted December 25, 2021 (edited)
I have a class that contains some logic that's applied to data selected by the user. The selections are added to a TListView, and an instance of the object is attached to the .Data property for each ListItem.
They're being processed more or less in sequence right now, and I believe the overall processing time can be sped up a lot if I process each one as a task.
I want to update columns in the ListView as each one finishes with a status message and elapsed time.
But using the PPL, there's nothing that says when a task has completed.


Pipeline. Input
FolderScanner
FileReader
FileReader
...
FileReader
StatCollector
StatCollector
StatAggregator
Pipeline. Output


Parallel Map
Serial
Parallel
553 ms, 70435 primes, sum = 105363426899
544 ms, 70435 primes, sum = 105363426899
580 ms, 70435 primes, sum = 105363426899
86 ms, 70435 primes, sum = 105363426899
102 ms, 70435 primes, sum = 105363426899
75 ms, 70435 primes, sum = 105363426899


TimedTask tester
Interval (sec):
10
Server time:
Tue, 16 May 2023 20:10:30 GMT
Start
Stop
Your IP:
85.10.34.64

Chapter 11: Using External Libraries

No-images…

Chapter 12: Best Practices

No-images…