

C++ Unterstützung für CADUL unter RMOS3

Spezifikation

Version: V1.2

Datum: 20.9.2001

Inhaltsverzeichnis

1 Einführung.....	3
2 Compilerschalter	3
2.1 kein NOALIGN, kein namespace std	3
2.2 kein NOALIGN, mit namespace std	3
2.3 mit NOALIGN, kein namespace std	3
2.4 mit NOALIGN, mit namespace std.....	3
3 Linkeranweisung.....	5
3.1 kein NOALIGN, kein namespace std	5
3.2 kein NOALIGN, mit namespace std	5
3.3 mit NOALIGN, kein namespace std	5
3.4 mit NOALIGN, mit namespace std.....	5
4 Lieferumfang.....	7
5 Beispielprogramm	8

1 Einführung

Diese Spezifikation beschreibt die C++ Unterstützung für den CADUL-Compiler ablauffähig unter RMOS3. Unterstützt werden Exceptionhandling, STL und Streaming. Im bestehenden RMOS-Baum müssen neue Headerdateien und Bibliotheken eingespielt werden. Abhängig vom Schalter ALIGN und NAMESPACE müssen verschiedene Bibliotheken verwendet werden.

Die C++ Erweiterungen sind in der Bibliothek i386Cxx.LIB abgelegt, die stl- und streaming-Erweiterungen sind in der Bibliothek i387FTxx.LIB abgelegt. Bestimmte Klassen in der Bibliothek i387FTxx.lib benötigen Coprozessor-Unterstützung. Die Bibliothek i386Cxx.LIB benötigt keine Coprozessor-Unterstützung. Eine Coprozessor-Emulation wird nicht unterstützt.

Die Methode new() fordert Speicher aus dem RMOS Pool an, ohne einen Deskriptor zu erzeugen (Flat Deskriptor). Damit kann es zu keinem Engpaß an Deskriptoren kommen.

Hinweis: Es wird darauf hingewiesen, daß der CADUL-Compiler offiziell kein namespace std unterstützt!

2 Compilerschalter

Zum Übersetzen wird der Cxx386-Compiler verwendet. Der Schalter -VANSI darf nicht gesetzt sein. Wenn kein Exceptionhandling bzw. kein RTTI gewünscht wird, reicht es den Schalter -VCXX zu setzen. Die Dateien libc.sub und libx.sub werden aus dem CADUL-Verzeichnis geholt. Die Headerdateien werden zuerst im bisherigen Verzeichnis INC und dann im neuen Verzeichnis INCPLUS gesucht.

Folgende Compilerschalter müssen beim Übersetzen der C++ Dateien gesetzt werden:

2.1 kein NOALIGN, kein namespace std

```
-VCOMPACT -VSUBSYS=libc.sub,libx.sub -I%RBASE%\INC -I%RBASE%\INCPLUS -VNEARFAR  
-VROM -VFIXEDPARAMS -VSYMUPPER -VCXX=exceptions,rtti,instantiate=used  
-VFARCR -D_ARCHITECTURE_=386 -VLISTING -VBUILTIN -VNDP
```

2.2 kein NOALIGN, mit namespace std

```
-VCOMPACT -VSUBSYS=libc.sub,libx.sub -I%RBASE%\INC -I%RBASE%\INCPLUS -VNEARFAR  
-VROM -VFIXEDPARAMS -VSYMUPPER -VCXX=exceptions,rtti,namespaces,instantiate=used  
-D_HAS_NAMESPACE=1  
-VFARCR -D_ARCHITECTURE_=386 -VLISTING -VBUILTIN -VNDP
```

2.3 mit NOALIGN, kein namespace std

```
-VCOMPACT -VSUBSYS=libc.sub,libx.sub -I%RBASE%\INC -I%RBASE%\INCPLUS -VNEARFAR  
-VROM -VFIXEDPARAMS -VSYMUPPER -VCXX=exceptions,rtti,instantiate=used  
-VFARCR -D_ARCHITECTURE_=386 -VLISTING -VBUILTIN -VNDP  
-VNOALIGN
```

2.4 mit NOALIGN, mit namespace std

```
-VCOMPACT -VSUBSYS=libc.sub,libx.sub -I%RBASE%\INC -I%RBASE%\INCPLUS -VNEARFAR  
-VROM -VFIXEDPARAMS -VSYMUPPER -VCXX=exceptions,rtti,namespaces,instantiate=used  
-VFARCR -D_ARCHITECTURE_=386 -VLISTING -VBUILTIN -VNDP
```

`-D_HAS_NAMESPACE=1 -VNOALIGN`

3 Linkeranweisung

Die Linkeranweisung sieht wie folgt aus:

```
@ECHO OFF
ECHO %RBASE%\LIB\CADUL\CSTART.OBJ > LNK.CMD
ECHO TEST.OBJ >>LNK.CMD
ECHO %RBASE%\LIB\CADUL\I387FTxx.LIB >>LNK.CMD
ECHO %RBASE%\LIB\CADUL\I386CXxx.LIB >>LNK.CMD
ECHO %RBASE%\LIB\CADUL\RM3CRIF.LIB >>LNK.CMD
ECHO %RBASE%\LIB\CADUL\RM3HLI.LIB >>LNK.CMD
ECHO %RBASE%\SYSTEM\PC\RM3NUC.GAT >>LNK.CMD
ECHO -SEGSIZE STACK+=4000 >>LNK.CMD
ECHO ON
LINK386 -CF LNK.CMD -LOADABLE -SU -CPLUS FAR -OJ B.386 -PR TEST.MP1 -TD .
@ECHO OFF
DEL LNK.CMD
```

Wichtig sind die Schalter `-CPLUS` und `-SU`. Die C++ Erweiterungen sind in der Bibliothek `I386CXxx.LIB` abgelegt, die `stl`- und `streaming`-Erweiterungen sind in der Bibliothek `I387FTxx.LIB` abgelegt. Bestimmte Klassen in der Bibliothek `I387FTxx.lib` benötigen Coprozessor-Unterstützung. Die Bibliothek `I386CXxx.LIB` benötigt keine Coprozessor-Unterstützung.

Abhängig von dem Schalter `ALIGN` und `NAMESPACE` müssen folgende Bibliotheken gebunden werden:

3.1 kein NOALIGN, kein namespace std

Datei `test00.bat`:

```
...
ECHO %RBASE%\LIB\CADUL\I387FT00.LIB >>LNK.CMD
ECHO %RBASE%\LIB\CADUL\I386CX00.LIB >>LNK.CMD
```

3.2 kein NOALIGN, mit namespace std

Datei `test01.bat`:

```
...
ECHO %RBASE%\LIB\CADUL\I387FT01.LIB >>LNK.CMD
ECHO %RBASE%\LIB\CADUL\I386CX00.LIB >>LNK.CMD
```

3.3 mit NOALIGN, kein namespace std

Datei `test10.bat`:

```
...
ECHO %RBASE%\LIB\CADUL\I387FT10.LIB >>LNK.CMD
ECHO %RBASE%\LIB\CADUL\I386CX10.LIB >>LNK.CMD
```

3.4 mit NOALIGN, mit namespace std

Datei `test11.bat`:

```
...
ECHO %RBASE%\LIB\CADUL\I387FT11.LIB >>LNK.CMD
```

```
ECHO  %RBASE%\LIB\CADUL\I386CX10.LIB    >>LNK.CMD
```

4 Lieferumfang

Die bisherigen Headerdateien limits.h, locale.h, math.h, new.h, stddef.h und stdlib.h im RMOS-Verzeichnis INC werden ausgetauscht. Die neuen C++ Headerdateien werden in das neue RMOS-Verzeichnis INCPLUS kopiert.

Die neuen Bibliotheken i386Cxx.lib und i387FTxx.lib werden in das RMOS-Verzeichnis LIB/CADUL kopiert.

Das C++ Beispielprogramm wird in das Verzeichnis EXAMPLES/CPLUS kopiert.

5 Beispielprogramm

```
/*--FILE HEADER-----
;
;   filename      : test.C
;   version       : 0001
;   contents      : Test task to test the C++ runtime library
;   author        : B.N.
;   date          : Aug 31, 2001
;
;--FUNCTION AND RMOS ENVIRONMENT-----
;
;   function      : Test task to test the C++ runtime library
;
;   RMOS3
;
;--END OF FILE HEADER-----
;
;--MODIFICATION HEADER-----
;
;   Aug 31, 2001  B.N.  0001  original issue
;
;--END OF MODIFICATION HEADER-----*/

#include <stdio.h>
#include <stdlib.h>
#include <exception.h>
#include <new.h>
#include <typeinfo.h>
#include <stdexcept.h>
#include <iostream.h>
#include <fstream.h>
#include <algorithm>
#include <vector>
#include <iterator>

void _FAR error(void)
{
    cout << "Error\n";
    exit(1);
}

/*****

                        Application task
                        =====

*****/
void _FAR main (void)
{
    int i, number;
    char buffer[10];

    printf("Hello\n");

    // test the streaming
    cout << "Input number: " << endl;
    cin >> number;
    cout << "Input number was : " << number << endl;
    ifstream in("c:/c.bat", ios::in);
```



```
ofstream out("c:/cc.bat", ios::out);
if (!in) cout << "error1\n";
if (!out) cout << "error2\n";
while (in.read(buffer,10))
{
    cout << "write ";
    out.write(buffer,10);
}
cout << endl;
set_new_handler(error);

// test the new function (now with flat alloc!!)
for (i=0; i<10000; i++)
{
    int *p =new int[10];
}

// test the exception handling
try {
    throw 2;
}
catch (const char *p) {
    printf("catch -%p-\n");
}
catch (const int i) {
    printf("catch -%d-\n",i);
}

// test the stl
int a[] = {100,200,320,1,5,6};
vector< int > v;
// copy array int 'a' to container 'v'
copy(a,a+6,back_inserter< vector<int> >(v));

sort( a, (a+6));
printf("sorted numbers: %d, %d, %d, %d, %d, %d\n",
    a[0], a[1], a[2], a[3], a[4], a[5]);
printf("size 'v' %d, last number is %d\n", v.size(), *(v.end()-1));
}
```