```
//=======================================
// Compiler: MSL
// Project:  TestIO
// Bestand:  FileReadSync.h
// Implemen: FileReadSync.cp
//=======================================

#pragma once

int main();

bool Test();

bool Test(z2 invoer_refNum,
          z4 invoer_lengte);

bool Test(z2 invoer_refNum,
          z4 invoer_lengte,
          p_char buffer);

bool Lees(c_z4 lengte,
          r_ParamBlockRec pb);
```

```
//=======================================
// Compiler: MSL
// Project:  TestIO
// Bestand:  FileReadSync.cp
//=======================================

#include "FileReadSync.h"

c_z4 buffer_grootte=20*1024*1024;

int main()
{
    cout << "Start van het programma.";
    cout << endl;

    cout << "Typ Command-Q om te stoppen.";
    cout << endl;

    c_n8 start=Tijd::Geef_microseconden();
    c_bool ok=Test();
    c_n8 einde=Tijd::Geef_microseconden();
    c_n8 duur=einde-start;
    cout << "Duur=";
    cout << duur;
    cout << endl;

    if (!ok)
        {
        cout << "Het programma mislukt.";
        cout << endl;
        }

    cout << "Einde van het programma.";
    cout << endl;

    return 0;
}

bool Test()
{
    // 1. Maak invoer spec.
    FSSpec invoerSpec;
    OSErr err=Files::OSERR_FSMakeFSSpec(0,
                                        0,
                                        "\pTemp:Invoer:Inside Macintosh.img",
```

```
                                        invoerSpec);
    if (err!=osErr_noErr)
        {
        cout << "Pad van invoer is niet gevonden.";
        cout << endl;
        return false;
        }

    // 2. Open invoer.
    z2 invoer_refNum;
    err=Files::OSERR_FSpOpenDF(invoerSpec,
                               permission_rd,
                               invoer_refNum);
    if (err!=osErr_noErr)
        {
        cout << "Open data fork mislukt.";
        cout << endl;
        return false;
        }

    // 3. Zoek invoer lengte.
    z4 invoer_lengte;
    err=Files::OSERR_GetEOF(invoer_refNum,
                            invoer_lengte);
    if (err!=osErr_noErr)
        {
        cout << "Lees data fork lengte mislukt.";
        cout << endl;
        return false;
        }

    // 4. Test.
    Test(invoer_refNum,
         invoer_lengte);

    // 5. Sluit invoer.
    err=Files::OSERR_FSClose(invoer_refNum);
    if (err!=osErr_noErr)
        {
        cout << "Sluit data fork mislukt.";
        cout << endl;
        return false;
        }

    return true;
```

```
}

bool Test(c_z2 invoer_refNum,
          c_z4 invoer_lengte)
{
    // 1. Maak buffer.
    p_char buffer;
    OSErr err=Memory::OSERR_NewPtr(buffer_grootte,
                                   buffer);
    if (err!=osErr_noErr)
        {
        cout << "Te weinig geheugen voor buffer.";
        cout << endl;
        return false;
        }

    // 2. Test.
    Test(invoer_refNum,
         invoer_lengte,
         buffer);

    // 3. Wis buffer.
    err=Memory::OSERR_DisposePtr(buffer);
    if (err!=osErr_noErr)
        {
        cout << "Wis buffer pointer mislukt.";
        cout << endl;
        return false;
        }

    return true;
}

bool Test(c_z2 invoer_refNum,
          c_z4 invoer_lengte,
          p_char buffer)
{
    ParamBlockRec pb;
    IOParam &in=pb.ioParam;
    Blok::Zero(sizeof(IOParam),
               &in);
    in.ioRefNum=invoer_refNum;
    in.ioBuffer=buffer;

    z4 aBlok;
```

```
    z4 overschot;
    aBlok=invoer_lengte/buffer_grootte;
    overschot=invoer_lengte-aBlok*buffer_grootte;

    while (aBlok)
        {
        if (!Lees(buffer_grootte,
                  pb))
            {
            cout << "Lees blok mislukt.";
            cout << endl;
            return false;
            }
        aBlok--;
        }
    if (overschot)
        {
        if (!Lees(overschot,
                  pb))
            {
            cout << "Lees blok mislukt.";
            cout << endl;
            return false;
            }
        }

    return true;
}

bool Lees(c_z4 lengte,
          r_ParamBlockRec pb)
{
    pb.ioParam.ioReqCount=lengte;
    c_OSErr err=Files::OSERR_PBReadSync(pb);
    if (err!=osErr_noErr)
        {
        cout << "Lezen mislukt.";
        cout << endl;
        return false;
        }
    return true;
}
```