

ML64-Fastpath DLL Framework – Standaard Documentpagina

ML64-Fastpath DLL Framework

Deze documentpagina bevat het volledige, compacte en direct inzetbare multi-export DLL-framework met Fortran 95, ML64-assembly, C-ABI-contracten en Rocket COBOL-aanroepvoorbeelden.

1. Projectstructuur

```
mini-fastpath\  
core_exports.f90  
fastpath.asm  
core.def  
core.h  
test_c.c  
prog.cbl  
build.bat
```

2. Fortran 95 module - core_exports.f90

```

module core_exports
use iso_c_binding
implicit none

interface
subroutine FILL_VALUES_FAST(arr, n, factor) bind(C, name="FILL_VALUES_FAST")
use iso_c_binding
real(c_double) :: arr(*)
integer(c_int), value :: n
real(c_double), value :: factor
end subroutine FILL_VALUES_FAST
end interface

contains

function SCALE(x, factor) result(res) bind(C, name="SCALE")
real(c_double), value :: x, factor
real(c_double) :: res
res = x * factor
end function SCALE

function FILL_VALUES(arr, n, factor) result(err) bind(C, name="FILL_VALUES")
real(c_double), intent(out) :: arr(*)
integer(c_int), value :: n
real(c_double), value :: factor
integer(c_int) :: err
integer :: i

if (n <= 0) then
err = 1
return
end if

if (n > 1000_c_int) then
call FILL_VALUES_FAST(arr, n, factor)
err = 0
return
end if

do i = 1, n
arr(i) = dble(i) * factor
end do

err = 0
end function FILL_VALUES

end module core_exports

```

3. ML64-fastpath routine - fastpath.asm

```
option casemap:none

.code

FILL_VALUES_FAST PROC

push rbx

xor r8d, r8d ; i = 0
test edx, edx
jle done

loop_start:
inc r8d ; i++
cvtsi2sd xmm0, r8d ; xmm0 = (double)i
mulsd xmm0, xmm2 ; xmm0 *= factor

mov rax, rcx ; base pointer
mov rbx, r8
dec rbx ; i-1
shl rbx, 3 ; *8 bytes
add rax, rbx
movsd qword ptr [rax], xmm0

cmp r8d, edx
jl loop_start

done:
pop rbx
ret

FILL_VALUES_FAST ENDP

END
```

4. DEF-file - core.def

```
LIBRARY core
EXPORTS
SCALE
FILL_VALUES
```

5. C-header - core.h

```
#ifndef CORE_H
#define CORE_H

double SCALE(double x, double factor);
int FILL_VALUES(double* arr, int n, double factor);

#endif
```

6. C-testprogramma - test_c.c

```
#include <stdio.h>
#include "core.h"

int main() {
    double arr[5];
    int err = FILL_VALUES(arr, 5, 2.0);

    printf("err = %d\n", err);
    for (int i = 0; i < 5; i++)
        printf("%f\n", arr[i]);

    return 0;
}
```

7. Rocket COBOL test - prog.cbl

IDENTIFICATION DIVISION.

PROGRAM-ID. PROG.

DATA DIVISION.

WORKING-STORAGE SECTION.

01 WS-LEN PIC S9(9) COMP-5 VALUE 5.

01 WS-FACTOR COMP-2 VALUE 2.0.

01 WS-ERR PIC S9(9) COMP-5.

01 WS-ARR.

05 WS-VAL OCCURS 5 TIMES COMP-2.

01 I PIC S9(9) COMP-5.

PROCEDURE DIVISION.

CALL "FILL_VALUES"

USING WS-VAL WS-LEN WS-FACTOR

RETURNING WS-ERR

END-CALL

DISPLAY "ERR = " WS-ERR

PERFORM VARYING I FROM 1 BY 1 UNTIL I > WS-LEN

DISPLAY "VAL(" I ") = " WS-VAL(I)

END-PERFORM

STOP RUN.

8. Buildscript - build.bat

```
@echo off
setlocal

rem ML64 → OBJ
ml64 /c /Fo fastpath.obj fastpath.asm

rem Fortran → OBJ
ifx /c /MD core_exports.f90 /object:core_exports.obj

rem DLL bouwen
link /DLL /OUT:core.dll core_exports.obj fastpath.obj /DEF:core.def

rem C test
cl /MD test_c.c core.lib /Fe:test_c.exe

rem COBOL test
cobol prog.cbl
link prog.obj core.lib /OUT:prog.exe

echo Klaar. Zorg dat core.dll naast test_c.exe en prog.exe staat.
endlocal
```

9. Samenvatting

Dit framework biedt:

- Fortran 95 exports met C-ABI
- Automatische ML64-fastpath voor grote arrays
- Stabiele DLL-exports via DEF
- C-header als contract
- C- en COBOL-testprogramma's
- Eén buildscript dat alles bouwt

Het project is direct compileerbaar en vormt een solide basis voor verdere uitbreiding.