IBM XL Fortran for AIX

The final clause in OpenMP 3.1

By: Dorra Bouchiha

Level: Intermediate

May 2012

Contents

IBM XL Fortran for AIX 1

About this Tutorial 3

Objectives 3

Prerequisites 3

System Requirements 3

Glossary 3

Start the Terminal Emulator to AIX System 4

Get Started with IBM XL Fortran Compiler 4

Get Started with the OpenMP final clause: 5

Conclusion 8

Trademarks 8

Resources 8

Before you start

About this Tutorial

This demo illustrates the use of the new final clause to the task construct in the IBM XL Fortran compiler. The principles demonstrated here also apply to C and C++ programs using the IBM XL C/C++ compilers.

Objectives

- Use IBM XL Fortran Compiler to illustrate the use of the new final clause to the task construct
- Total time: 20 minutes

Prerequisites

- Basic Unix skills
- Basic Source code compile and build experience
- Basic Fortran knowledge
- Basic OpenMp knowledge

System Requirements

http://www.ibm.com/software/awdtools/fortran/xlfortran/aix/sysreq

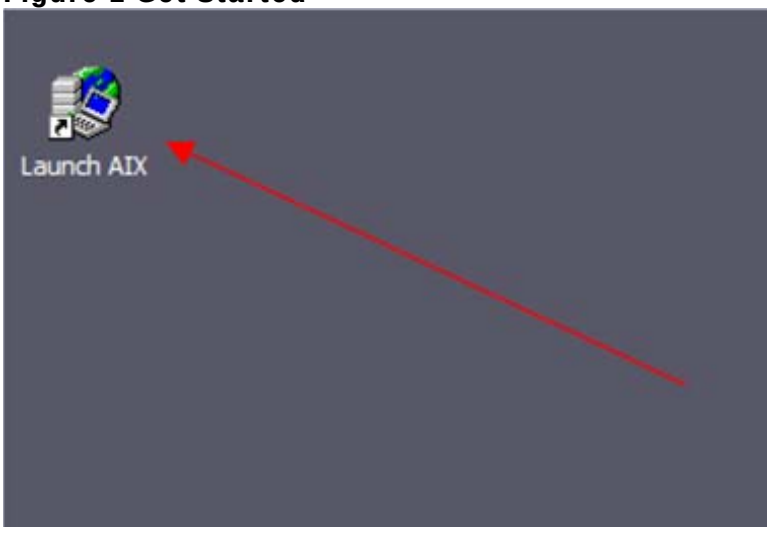
Glossary

IBM XL Fortran Compiler: The IBM® XL Fortran compiler offers advanced compiler and optimization technologies and is built on a common code base for easier porting of your applications between platforms. It complies with the latest Fortran international standards and industry specifications and supports a large array of common language features.

Getting Started

Start the Terminal Emulator to AIX System

Figure 1 Get Started

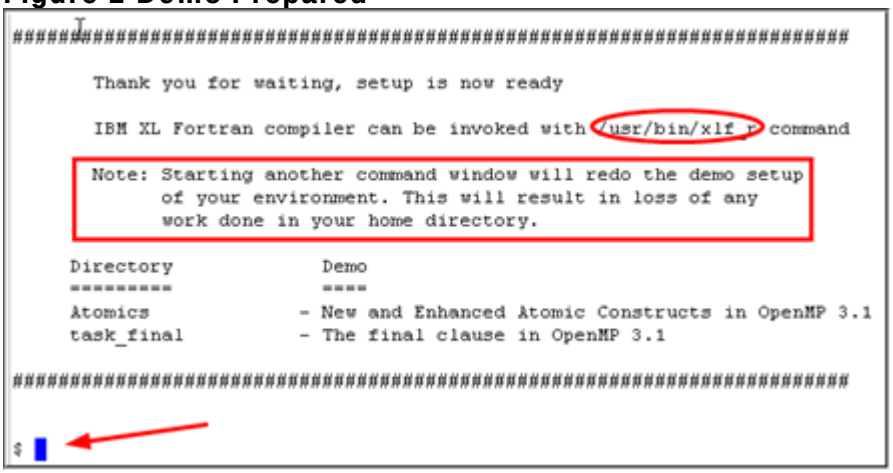


Double click the “Launch AIX” icon on the desktop (See Figure 1) to start the character terminal to AIX system.

Get Started with IBM XL Fortran Compiler

A successful login will result with the user presented with a menu of the demos hosted on the server. Type 19 and press Enter to select the “Fortran Exploring new OMP3.1 constructs” demo.

Figure 2 Demo Prepared



On the terminal window you will see important information and the directory path to the compiler invocation command (See Figure 2 Demo Prepared *oval red*).

Note: Starting another command window will start the demonstration setup of your environment. This will result in loss of any work done in your home directory. This will impact any progress you have made on demo steps going forward.

This demo does not require more than one terminal window. However, if you prefer more than one terminal window then you may open them before going forward.

The terminal window is now ready for commands. The directory OMP31_F_demo/task_final contains the source code for the demo. Type the ls command to see the directory content (See Figure 3 Contents).

Command:

ls

Figure 3 Contents

```
$ ls
task_final.f95
$
```

The demo contains one sample program written in Fortran 95 and is compliant with OpenMP 3.1 specifications.

Get Started with the OpenMP final clause:

Part 1: Using OMP3.1 final clause

Steps:

The demo sample task_final.f95 illustrates the basic use of the **final** clause to the **task** construct. The final clause forces all its generated child tasks to become final and included. An included task is a task for which execution is sequentially included in the generating task region. The final clause was introduced in OpenMP 3.1 to reduce the overhead of task switching between threads and allow for additional optimization.

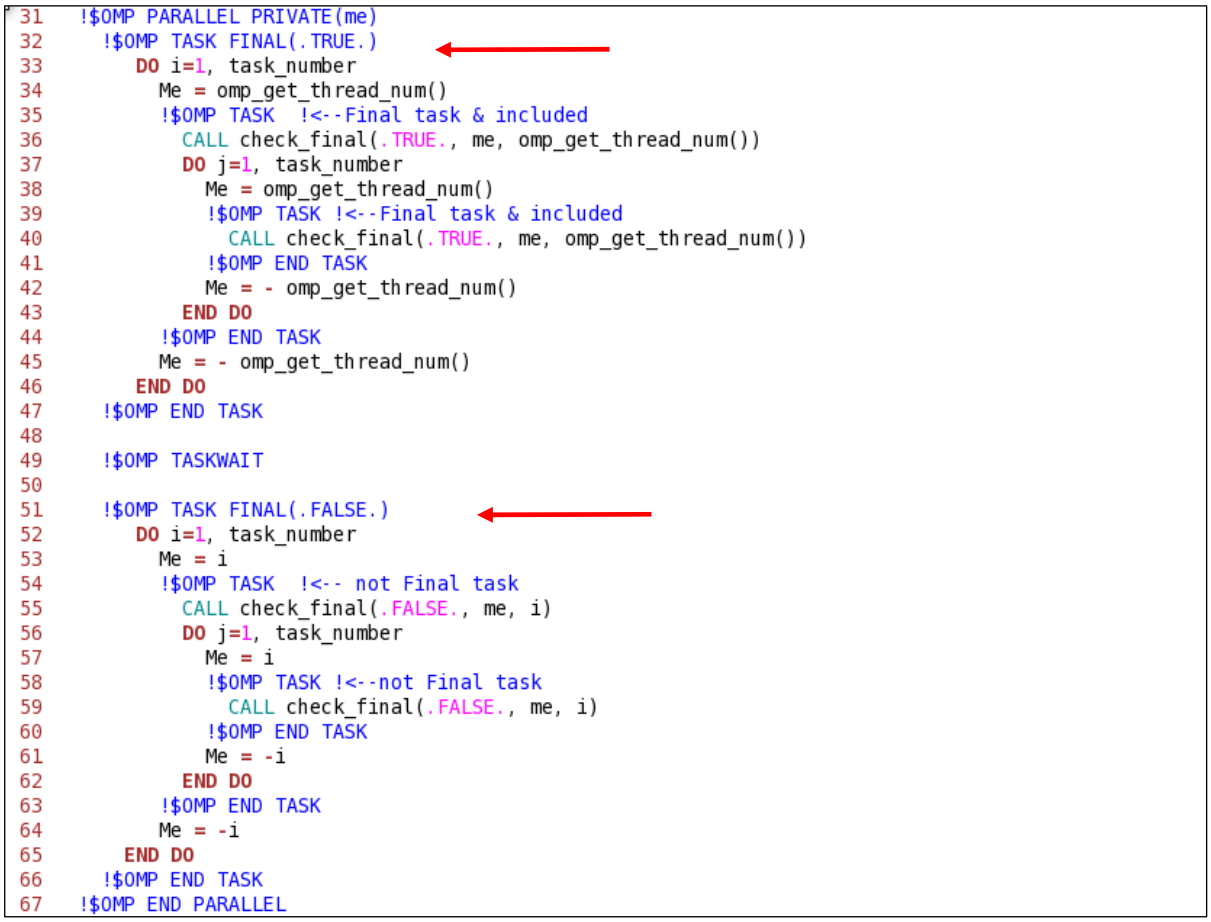
1. Compile and run task_final.f95 program.

   View the source code of task_final.f95 (See figure 4)

   Command:

vi task_final.f95

Figure 4 Program Source Code



```
31   !$OMP PARALLEL PRIVATE(me)
32     !$OMP TASK FINAL(.TRUE.)
33        DO i=1, task_number
34          Me = omp_get_thread_num()
35          !$OMP TASK  !<--Final task & included
36            CALL check_final(.TRUE., me, omp_get_thread_num())
37            DO j=1, task_number
38              Me = omp_get_thread_num()
39              !$OMP TASK !<--Final task & included
40                CALL check_final(.TRUE., me, omp_get_thread_num())
41              !$OMP END TASK
42              Me = - omp_get_thread_num()
43            END DO
44          !$OMP END TASK
45          Me = - omp_get_thread_num()
46        END DO
47     !$OMP END TASK
48
49     !$OMP TASKWAIT
50
51     !$OMP TASK FINAL(.FALSE.)
52        DO i=1, task_number
53          Me = i
54          !$OMP TASK  !<-- not Final task
55            CALL check_final(.FALSE., me, i)
56            DO j=1, task_number
57              Me = i
58              !$OMP TASK !<--not Final task
59                CALL check_final(.FALSE., me, i)
60              !$OMP END TASK
61              Me = -i
62            END DO
63          !$OMP END TASK
64          Me = -i
65       END DO
66     !$OMP END TASK
67   !$OMP END PARALLEL
```

When the condition on the final clause is true (line 32), the encountering thread executes the associated task region and nested task regions. No other thread from the same team is allowed to execute a task region that is final and owned by another thread.

The subroutine check_final verifies that the task is executed immediately by the same thread when the condition on the final clause is true. Also, the call to the omp_in_final() routine returns true as the routine is called in a final task region.

When the final task condition is set to false, the nested task is not necessarily executed by the same thread and can deferred. The task may be added to the pool of tasks waiting to be executed by the next available thread.

Compile task_final.f95 (See figure 5)

Command:

xlf_r -qsmp=omp -c task_final.f95

Figure 5 Compile

```
$ xlf_r -qsmp=omp -c task_final.f95
** task_final   === End of Compilation 1 ===
1501-510  Compilation successful for file task_final.f95.
$
```

Link task_final.f95 (See figure 6)

Command:

xlf_r -qsmp=omp task_final.o

Figure 6 Link

```
$ xlf_r -qsmp=omp task_final.o
$
```

Run task_final.f95 (See Figure 7).

Command:

./a.out

Figure 7 Run

```
$ ./a.out
$
```

What you have learned

In this exercise you learned how to:

- Use the IBM XL Fortran for AIX compiler to compile programs.
- Use final clause to the task construct

Conclusion

This concludes the tutorial on using the IBM XL Fortran compiler. This tutorial has shown how to use the final clause to the task construct.

Trademarks

Resources

Learn

Overview of the IBM XL C/C++ and XL Fortran Compiler Family
http://www.ibm.com/support/docview.wss?uid=swg27005175

Getting Started with XL Fortran for AIX, V14.1
http://www.ibm.com/support/docview.wss?uid=swg27024216

Compiler Reference - XL Fortran for AIX, V14.1
http://www.ibm.com/support/docview.wss?uid=swg27024215

Language Reference - XL Fortran for AIX, V14.1
http://www.ibm.com/support/docview.wss?uid=swg27024218

Optimization:

Optimization and Programming Guide - XL Fortran for AIX, V14.1
http://www.ibm.com/support/docview.wss?uid=swg27024219

Code Optimization with the IBM XL Compilers
http://www.ibm.com/support/docview.wss?uid=swg27005174