Using OpenMP - The Examples

User's Guide

Version 1

April 2, 2009

1. Introduction

Thank you for your interest in these examples. There are a total of 41 examples in source format available. All of these are introduced and described in the book “Using OpenMP” [1].

The name of the corresponding source file has the following structure: fig<figure-number-in-book>-descriptive-text.<ext> The extension <ext> is either .c or .f90. For example: fig4.16-sections.c.

With the exception of the example given in Figure 6.21 there are no comments in the source. The examples are not only very simple, they are also discussed in detail in the book.

Although at times tempting to do, there are no changes compared to what is shown and discussed in the book.

In general, the goal was to go for simplicity. Each source can be used standalone. There is no need to have any special library, or tool. All that is needed is an OpenMP compliant compiler. Actually, all programs compile and execute without OpenMP activated, but of course there is no parallel execution.

Each program sets the number of threads from within the source. For this to work, dynamic thread adjustment is disabled at run time. A warning message is issued in case this should fail.

2. Installation

Installation is meant to be straightforward.

After unzipping the distribution directory, the following subdirectories should be present:

```
chapter3
chapter4
chapter6
chapter7
include
UsersGuide
```

This guide is located in directory “UsersGuide”. Directories “chapter3”, “chapter4”, “chapter6” and “chapter7” will be referred to as “the chapter directories” in the remainder of this document. They contain the sources with the respective examples, as well as a make file called “Makefile”.

If the “make” command is issued in one of the chapter directories, the following message appears:

```
$ make
Makefile not configured yet ..... but this is really easy to do

Remove or comment out this line in file Makefile:
   include make.inc

Next, select which compiler you want to use by activating
one of the following lines in file Makefile:
```

```
#include ../include/gnu-compilers.h
#include ../include/ibm-compilers.h
#include ../include/intel-compilers-linux.h
#include ../include/intel-compilers-macos.h
#include ../include/pgi-compilers.h
#include ../include/sun-studio-compilers.h
```

There has to be an appropriate include file for the compiler in directory “include”. In case this file does not exist yet, please check Appendix A for directions how to create one.

> Please note that the above listed configuration files are made available on a best effort basis. In some cases, testing has been limited. Please report any problems through the appropriate forum on http://www.openmp.org. Also note there are currently no configuration files for Windows.

In case an existing file can be used, or if one is newly created, simply include it in the make file in the respective chapter directories.

The examples are now ready to be built and executed.

3. Building and running the examples

The make files excel in simplicity. The goal was to supply make files that are as portable and easy as can be.

Each make file supports the following commands:

```
make help (the default) - gives an overview of the commands, plus the examples that can be built and run
make build              - builds all examples in this directory
make run                - runs all examples in this directory
make clean              - removes the object files and executables
```

Individual examples can be built and run by the following commands:

```
make fig<figure-number-in-book>.exe     - builds the specific example
make run-fig<figure-number-in-book>.exe - runs the specific example
```

4. Example

This example illustrates how to build and run a single program:[1]

```
$ cd Using-OpenMP-Examples-Distr
$ ls
chapter3/  chapter4/  chapter6/  chapter7/  include/
$ cd chapter4
$ make fig4.16.exe
cc -c -xO3 -xopenmp -xloopinfo fig4.16-sections.c
cc -o fig4.16.exe fig4.16-sections.o -xO3 -xopenmp -xloopinfo
$ make run-fig4.16.exe
In funcA: this section is executed by thread 0
In funcB: this section is executed by thread 1
$
```

The first make command compiles the source and links the object file to build an executable. The second make command runs the program. Note that there is no need to set the number of threads through environment variable `OMP_SET_NUM_THREADS`. This is handled within the source.

[1] *In this particular case the Sun Studio compiler is used. Depending on what compiler is used, the compile and link line may look differently.*

5. Feedback

The recommend method to provide feedback is to use the forum "Using OpenMP - The Examples" on http://www.openmp.org. This forum can be used to ask questions, start discussions and share results with the OpenMP community.

6. Acknowledgements

The following people contributed to this effort. I would like to thank them for their help, their time and patience with me.

Barbara Chapman, Lei Huang and Vijay Anand Kandiraju (University of Houston, TX, USA). Their assistance testing early versions of these examples against various compilers has been very helpful.

Thomas Nau at the University of Ulm in Germany for testing these examples against a variety of Solaris releases, servers and compilers.

Jakub Jelinek (RedHat), Larry Meadows (Intel), Michael Wolfe (Portland Group), Guansong Zhang and Kelvin Li (IBM) for providing the compiler related info, as well as for testing the examples.

Richard Friedman (Sun Microsystems) for setting up and maintaining the forum for these examples on the OpenMP web site.

7. Disclaimer

Care has been taken to test these examples against various compilers and Operating Systems, but this has been on a best effort basis only. If you find a flaw, please let us know through the forum.

These examples are not designed for verification of OpenMP compliance, nor correctness of an implementation.

8. References

[1] "Using OpenMP" - Barbara Chapman, Gabriele Jost, Ruud van der Pas, MIT Press, 2008, ISBN-10: 0-262-53302-2, ISBN-13: 978-0-262-53302-7

Appendix A - Defining a compiler include file

In case the include file for a compiler and/or Operating System does not exist yet, one has to create it. The name for this file is arbitrary, but make sure to have the corresponding include line in the make file in the various chapter directories.

In the include file, the following make variables need to be set:

`CC` Set this to point to the (path and) name of the C compiler
`FTN` Set this to point to the (path and) name of the Fortran compiler
`CFLAGS` Compiler option(s) for serial performance on the C compiler (could be blank)
`FFLAGS` Compiler option(s) for serial performance on the Fortran compiler (could be blank)
`LDFLAGS` Loader options (could be left blank if not needed)
`C_OMP` Compiler option(s) to activate OpenMP on the C compiler (will be added to `CFLAGS`)
`FTN_OMP` Compiler option(s) to activate OpenMP on the Fortran compiler (will be added to `FFLAGS`)

These variables are used in the make file in the various chapters. Once they has been set to the appropriate value(s), there should be no need for further changes.

It assumed that the C compiler supports the C99 standard. It may be needed to use a special compiler option for this.

The Fortran sources use the Fortran 90/95 formatting rules and have file name extension .f90. On many compilers this extension is interpreted to imply that the source is in free format. If this is not the case, please check the documentation of your compiler how to set this.