服务器安装（centos）

2019 年 10 月 15 日

0:53

- 参照官网　见附录 pdf
- https://www.jianshu.com/p/3471fc8f5e5f
- 环境变量最终的样子

```
# .bashrc

# Source global definitions
if [ -f /etc/bashrc ]; then
        . /etc/bashrc
fi

# Uncomment the following line if you don't like systemctl's auto-paging feature:
# export SYSTEMD_PAGER=
# User specific aliases and functions
export PATH=/mnt/zxw/anaconda3/bin:$PATH
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/usr/local/cuda-9.0/lib64
export PATH=$PATH:/usr/local/cuda-9.0/bin
export CUDA_HOME=$CUDA_HOME:/usr/local/cuda-9.0

##wrf
export DIR=/home/zxw/wrf/Build_WRF/LIBRARIES
export CC=gcc
export CXX=g++
export FC=gfortran
export FCFLAGS=-m64
export F77=gfortran
export FFLAGS=-m64
export PATH=$DIR/netcdf/bin:$PATH
export NETCDF=$DIR/netcdf
export PATH=$DIR/mpich/bin:$PATH
export LDFLAGS=-L$DIR/grib2/lib
export CPPFLAGS=-I$DIR/grib2/include
export JASPERLIB=$DIR/grib2/lib
export JASPERINC=$DIR/grib2/include
```

- 1.检测编译环境并测试

```
which gfortran
which cpp
which gcc
gfortran --version
gcc --version
g++ --version
```

#缺啥装啥 yum install

#测试（与官网无差别）

mkdir Build_WRF　#放安装的压缩文件

mkdir TESTS　　#放测试程序

cd TESTS/
wget http://www2.mmm.ucar.edu/wrf/OnLineTutorial/compile_tutorial/tar_files/Fortran_C_tests.tar
ls
tar -xf Fortran_C_tests.tar

- #Fixed Format Fortran Test:

gfortran TEST_1_fortran_only_fixed.f
ls
./a.out

- #Free Format Fortran:

```
gfortran TEST_2_fortran_only_free.f90
ls
 ./a.out
```

```
Assume Fortran 2003: has FLUSH, ALLOCATABLE, derived
type, and ISO C Binding
SUCCESS test 2 fortran only free format
```

- #c 测试

```
 gcc TEST_3_c_only.c
 ./a.out
```

```
SUCCESS test 3 c only
```

```
gcc -c -m64 TEST_4_fortran+c_c.c
gfortran -c -m64 TEST_4_fortran+c_f.f90
gfortran -m64 TEST_4_fortran+c_f.o TEST_4_fortran+c_c.o
./a.out
```

```
C function called by Fortran
Values are xx = 2.00 and ii = 1
SUCCESS test 4 fortran calling c
```

#接下来测试下 csh，perl，sh 是否可行。

```
./TEST_csh.csh
./TEST_perl.pl
./TEST_sh.sh
```

- 2.安装依赖库

首先在 Build_WRF 文件夹下面创建一个 LIBRARIES 的文件夹。然后下载所需的依赖库。（我开始忘记建 libraries QAQ 影响不大 只是看起来乱了一点）

和官网的区别 mpich 版本 需要下载对应 centos 版本 官网是 ubuntu 版本

mpich-3.0.4#这个有问题

netcdf-4.1.3

Jasper-1.900.1

libpng-1.2.50

zlib-1.2.7

mkdir LIBRARIES

cd LIBRARIES

#(ubuntu 用注释的这条)

#wget http://www2.mmm.ucar.edu/wrf/OnLineTutorial/compile_tutorial/tar_files/mpich-3.0.4.tar.gz
wget http://www.mpich.org/static/downloads/3.3.1/mpich-3.3.1.tar.gz
wget http://www2.mmm.ucar.edu/wrf/OnLineTutorial/compile_tutorial/tar_files/netcdf-4.1.3.tar.gz
wget http://www2.mmm.ucar.edu/wrf/OnLineTutorial/compile_tutorial/tar_files/jasper-1.900.1.tar.gz
wget http://www2.mmm.ucar.edu/wrf/OnLineTutorial/compile_tutorial/tar_files/libpng-1.2.50.tar.gz
wget http://www2.mmm.ucar.edu/wrf/OnLineTutorial/compile_tutorial/tar_files/zlib-1.2.7.ta

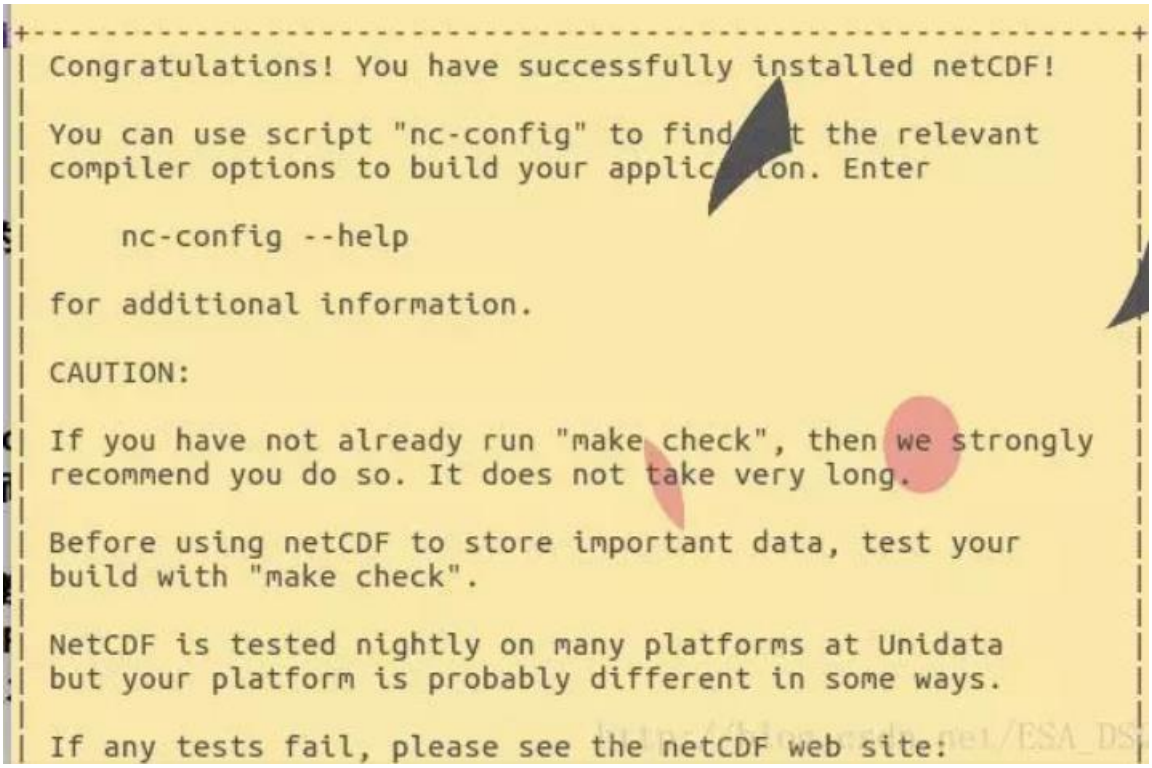
- netcdf 安装

  #改标红的这项 为你的目录

  export DIR=/home/zxw/wrf/Build_WRF/LIBRARIES
  export CC=gcc

```
export CXX=g++
export FC=gfortran
export FCFLAGS=-m64
export F77=gfortran
export FFLAGS=-m64

tar zxvf netcdf-4.1.3.tar.gz
cd netcdf-4.1.3
./configure --prefix=$DIR/netcdf --disable-dap \
 --disable-netcdf-4 --disable-shared
make
make install
export PATH=$DIR/netcdf/bin:$PATH
export NETCDF=$DIR/netcdf
```

- cd ..
- 成功标志
- 

mpich 安装

```
tar xzvf mpich-3.3.1.tar.gz
cd mpich-3.3.1/
./configure --prefix=$DIR/mpich
```

```
make
make install
export PATH=$DIR/mpich/bin:$PATH
```

```
cd ..
```

- ## zlib 安装

```
export LDFLAGS=-L$DIR/grib2/lib
export CPPFLAGS=-I$DIR/grib2/include
tar xzvf zlib-1.2.7.tar.gz
cd zlib-1.2.7
./configure --prefix=$DIR/grib2
make
make install
```

```
cd ..
```

- ## libpng 安装

```
tar xzvf libpng-1.2.50.tar.gz
cd libpng-1.2.50
./configure --prefix=$DIR/grib2
make
make install
```

```
cd ..
```

- ## jasper 安装

```
tar xzvf jasper-1.900.1.tar.gz
cd jasper-1.900.1
./configure --prefix=$DIR/grib2
make
make install
```

cd ..

- 库测试（与官网无差别，记得下载到 TEST 目录下）

cd LIBRARIES

wget https://link.jianshu.com/?t=http://www2.mmm.ucar.edu/wrf/OnLineTutorial/compile_tutorial/tar_files/Fortran_C_NETCDF_MPI_tests.tar

- Test #1: Fortran + C + NetCDF

tar -xf Fortran_C_NETCDF_MPI_tests.tar
cp ${NETCDF}/include/netcdf.inc .
gfortran -c 01_fortran+c+netcdf_f.f
gcc -c 01_fortran+c+netcdf_c.c
gfortran 01_fortran+c+netcdf_f.o 01_fortran+c+netcdf_c.o -L${NETCDF}/lib -lnetcdff -lnetcdf
./a.out

```
C function called by Fortran
Values are xx = 2.00 and ii = 1
SUCCESS test 1 fortran + c + netcdf
```

- Test #2: Fortran + C + NetCDF + MPI

cp ${NETCDF}/include/netcdf.inc .

mpif90 -c 02_fortran+c+netcdf+mpi_f.f mpicc -c

02_fortran+c+netcdf+mpi_c.c mpif90 02_fortran+c+netcdf+mpi_f.o \

02_fortran+c+netcdf+mpi_c.o \ -L${NETCDF}/lib -lnetcdff -lnetcdf

mpirun ./a.out

```
C function called by Fortran
Values are xx = 2.00 and ii = 1
status = 2
SUCCESS test 2 fortran + c + netcdf + mpi
```

- 3.Building WRFV3

cd ../Build_WRF

wget http://www2.mmm.ucar.edu/wrf/src/WRFV3.9.1.1.TAR.gz

gunzip WRFV3.9.1.1.TAR.gz

tar -xf WRFV3.9.1.1.TAR

cd WRFV3

./configure

```
checking for perl5... no
checking for perl... found /usr/bin/perl (perl)
Will use NETCDF in dir: /home/Build_WRF/LIBRARIES/netcdf
HDF5 not set in environment. Will configure WRF for use without.
PHDF5 not set in environment. Will configure WRF for use without.
Will use 'time' to report timing information
$JASPERLIB or $JASPERINC not found in environment, configuring to build without grib2 I/O...
------------------------------------------------------------------------
Please select from among the following Linux x86_64 options:

  1. (serial)   2. (smpar)   3. (dmpar)   4. (dm+sm)   PGI (pgf90/gcc)
  5. (serial)   6. (smpar)   7. (dmpar)   8. (dm+sm)   PGI (pgf90/pgcc): SGI MPT
  9. (serial)  10. (smpar)  11. (dmpar)  12. (dm+sm)   PGI (pgf90/gcc): PGI accelerator
 13. (serial)  14. (smpar)  15. (dmpar)  16. (dm+sm)   INTEL (ifort/icc)
                                         17. (dm+sm)   INTEL (ifort/icc): Xeon Phi (MIC architecture)
 18. (serial)  19. (smpar)  20. (dmpar)  21. (dm+sm)   INTEL (ifort/icc): Xeon (SNB with AVX mods)
 22. (serial)  23. (smpar)  24. (dmpar)  25. (dm+sm)   INTEL (ifort/icc): SGI MPT
 26. (serial)  27. (smpar)  28. (dmpar)  29. (dm+sm)   INTEL (ifort/icc): IBM POE
 30. (serial)               31. (dmpar)                PATHSCALE (pathf90/pathcc)
 32. (serial)  33. (smpar)  34. (dmpar)  35. (dm+sm)   GNU (gfortran/gcc)
 36. (serial)  37. (smpar)  38. (dmpar)  39. (dm+sm)   IBM (xlf90_r/cc_r)
 40. (serial)  41. (smpar)  42. (dmpar)  43. (dm+sm)   PGI (ftn/gcc): Cray XC CLE
 44. (serial)  45. (smpar)  46. (dmpar)  47. (dm+sm)   CRAY CCE (ftn $(NOOMP)/cc): Cray XE and XC
 48. (serial)  49. (smpar)  50. (dmpar)  51. (dm+sm)   INTEL (ftn/icc): Cray XC
 52. (serial)  53. (smpar)  54. (dmpar)  55. (dm+sm)   PGI (pgf90/pgcc)
 56. (serial)  57. (smpar)  58. (dmpar)  59. (dm+sm)   PGI (pgf90/gcc): -f90=pgf90
 60. (serial)  61. (smpar)  62. (dmpar)  63. (dm+sm)   PGI (pgf90/pgcc): -f90=pgf90
 64. (serial)  65. (smpar)  66. (dmpar)  67. (dm+sm)   INTEL (ifort/icc): HSW/BDW
 68. (serial)  69. (smpar)  70. (dmpar)  71. (dm+sm)   INTEL (ifort/icc): KNL MIC
 72. (serial)  73. (smpar)  74. (dmpar)  75. (dm+sm)   FUJITSU (frtpx/fccpx): FX10/FX100 SPARC64 IXfx/Xl
```

按需要 这里选 34 和 1

注释：**serial**（单核） means single processor （考虑到大家机器的实际情况，请大家选择这项）

smpar （多核公用内存）means Symmetric Multi-Processing/Shared Memory Parallel (OpenMPI)

dmpar（多核分布式内存） means Distributed Memory Parallel (MPI)

dm+sm means Distributed Memory with Shared Memory (for example, MPI across nodes with OpenMP within a node)

dm 和 **sm** 都需要安装 **mpi** 才能实现多核的并行运算（后面看附录二）

一般选第一个 按需要

```
./compile em_real >& log.compile
```

等待一会》》》》》》

- Once your configuration is complete, you should have a `configure.wrf` file, and you are ready to compile. To compile WRFV3, you will need to decide which type of case you wish to compile. The options are listed below:

  `em_real` (3d real case)
  `em_quarter_ss` (3d ideal case)
  `em_b_wave` (3d ideal case)
  `em_les` (3d ideal case)
  `em_heldsuarez` (3d ideal case)
  `em_tropical_cyclone` (3d ideal case)
  `em_hill2d_x` (2d ideal case)
  `em_squall2d_x` (2d ideal case)
  `em_squall2d_y` (2d ideal case)
  `em_grav2d_x` (2d ideal case)
  `em_seabreeze2d_x` (2d ideal case)
  `em_scm_xy` (1d ideal case)

```
ls -ls main/*.exe
```

出现

```
38892 -rwxr-xr-x 1 root 8079 39822032 10月  13 17:16 main/ndown.exe
38768 -rwxr-xr-x 1 root 8079 39694920 10月  13 17:16 main/real.exe
38388 -rwxr-xr-x 1 root 8079 39306160 10月  13 17:16 main/tc.exe
42360 -rwxr-xr-x 1 root 8079 43374736 10月  13 17:16 main/wrf.exe
```

```
SUCCESS test 1 fortran only fixed format
```

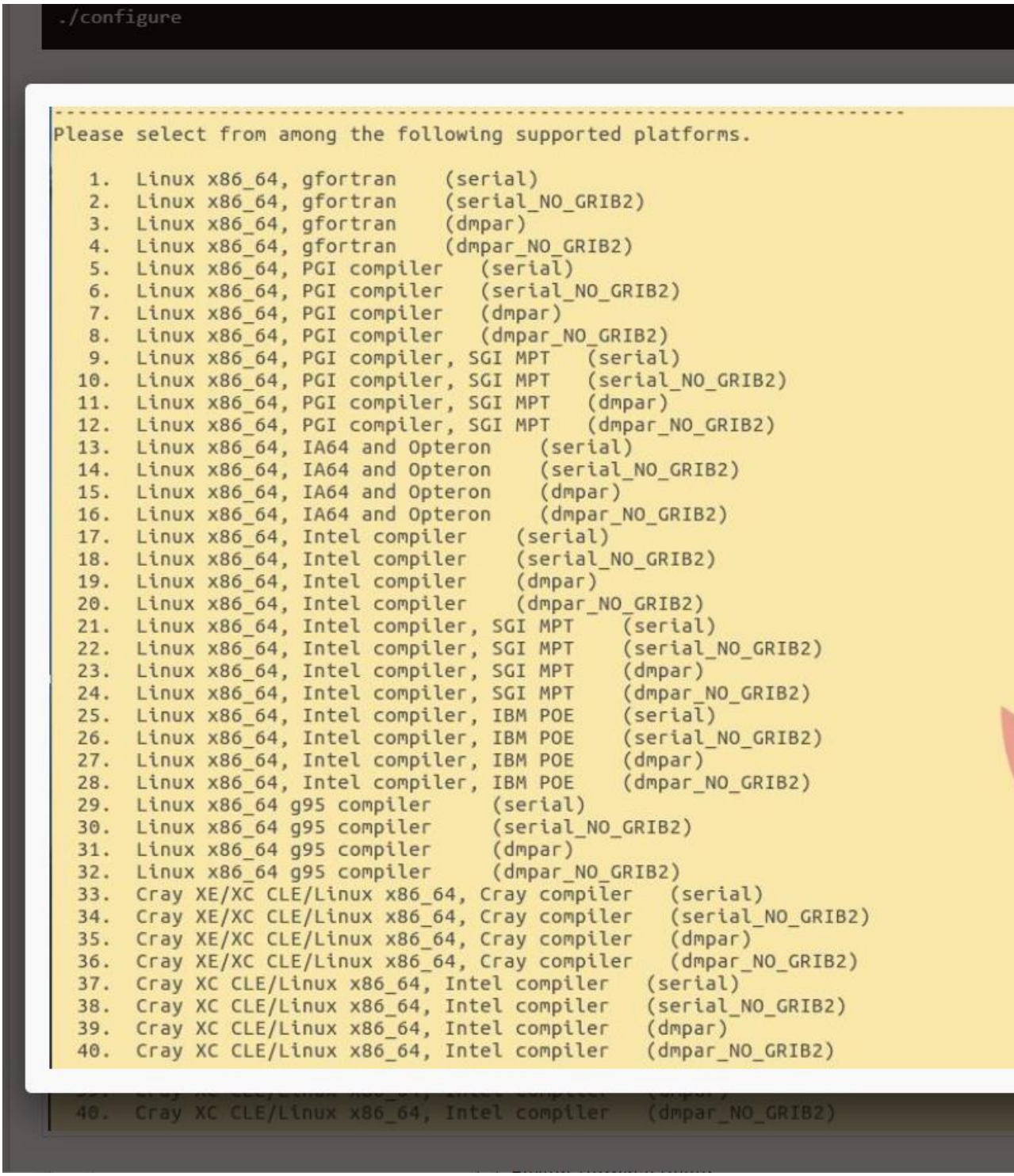
- ## 4. 编译 WPS

wget http://www2.mmm.ucar.edu/wrf/src/WPSV3.9.1.TAR.gz

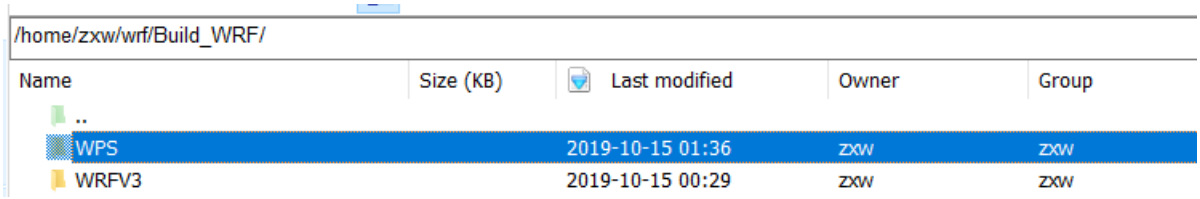
cd ..

```
gunzip WPSV3.9.1.TAR.gz
tar -xf WPSV3.9.1.TAR
cd WPS
./clean
export JASPERLIB=$DIR/grib2/lib
export JASPERINC=$DIR/grib2/include
./configure
```

官网建议选 1，我开始选其他报错



compile 之前先看结构 WPS 是不是与 WRF3 并列



如果是 略过这一步 不是的话 需要修改

/home/zxw/wrf/Build_WRF/WPS/configure.wps

框中部分改为相对于 configure.wps 文件的 WRF3 的相对路径 因为 WPS 的编译需要

wrf 的 I/O 接口（详情见手册解释）

- the `metgrid.exe` and `geogrid.exe` programs rely on the WRF model's I/O libraries. There is a line in the `configure.wps` file that directs the WPS build system to the location of the I/O libraries from the WRF model:

  ```
  WRF_DIR = ../WRFV3
  ```

  Above is the default setting. As long as the name of the WRF model's top-level directory is "WRFV3" and the WPS and WRFV3 directories are at the same level (which they should be if you have followed exactly as instructed on this page so far), then the existing default setting is correct and there is no need to change it. If it is not correct, you must modify the configure file and then save the changes before compiling.

```
# configure.wps
#
# This file was automatically generated by the configure script in the
# top level directory. You may make changes to the settings in this
# file but be aware they will be overwritten each time you run configure.
# Ordinarily, it is necessary to run configure once, when the code is
# first installed.
#
# To permanently change options, change the settings for your platform
# in the file arch/configure.defaults, the preamble, and the postamble -
# then rerun configure.
#

.SUFFIXES: .F .f .c .o

SHELL              =       /bin/sh

NCARG_LIBS    = -L$(NCARG_ROOT)/lib -lncarg -lncarg_gks -lncarg_c \
        -lX11 -lXext -lpng -lz -lcairo -lfontconfig -lpixman-1 \
        -lfreetype -lexpat -lpthread -lbz2 -lXrender -lgfortran -lgcc

NCARG_LIBS2   = # May be overridden by architecture specific value below

FDEFS     = -DUSE_JPEG2000 -DUSE_PNG

# Listing of options that are usually independent of machine type.
# When necessary, these are over-ridden by each architecture.

ARFLAGS     =

PERL      = perl

RANLIB      = echo

WRF_DIR     = ../WRFV3

WRF_INCLUDE     =       -I$(WRF_DIR)/external/io_netcdf \
```

./compile >& log.compile

ls -ls *.exe

```
[zxw@30 WPS]$ ls -ls *.exe
0 lrwxrwxrwx 1 zxw zxw 23 Oct 15 01:36 geogrid.exe -> geogrid/src/geogrid.exe
0 lrwxrwxrwx 1 zxw zxw 23 Oct 15 01:36 metgrid.exe -> metgrid/src/metgrid.exe
0 lrwxrwxrwx 1 zxw zxw 21 Oct 15 01:36 ungrib.exe -> ungrib/src/ungrib.exe
```

- 5 Static Geography Data

下载 WPS_GEOG

http://www2.mmm.ucar.edu/wrf/users/download/get_sources_wps_geog.html

Wget

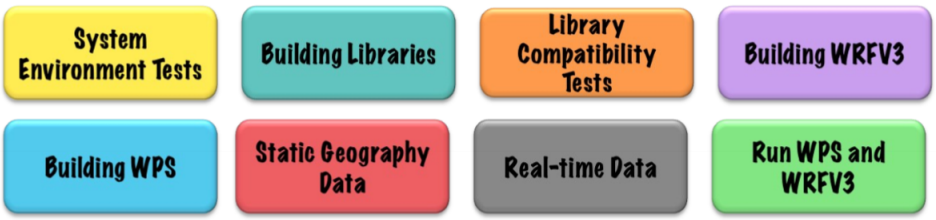
mv geog WPS_GEOG

<<ch10-part2-compiling_wrf.pdf>>

How to Compile WRF: The Complete Process

This page is meant to provide guidance through the steps of compiling WRF. It will take a beginning user through the processes of ensuring th environment is set up correctly, to testing the components and their compatibility with each other, then to installing WRFV3 and WPS, and fina guidance for preparing to run WPS and then WRFV3.

Click on a tab below for quick navigation. If you are a beginner, it is recommended to start at the beginning and follow through each step.



**__IMPORTANT NOTES: PLEASE READ BEFORE CONTINUING!__

- In order to use personal machines, you must have all the pre-required programs and compilers built, as well as their functionality/compatibi through testing. We cannot be responsible or provide assistance for the installation of Linux, Linux utilities, or the compilers.

- We are attempting to walk you through the steps for building necessary libraries (netCDF, MPICH, JasPer, Libpng, and Zlib); however, if yo errors, we cannot be responsible for helping to correct the errors, as these are related to your particular system, and are not supported by ou You will need to contact someone in your systems administation office, or go to the library websites to contact someone in their support group

- All of the examples given here are in tcsh. If you are very familiar with another shell (e.g., bash), and feel comfortable making the necessar the commands, then feel free to use your other shell. If not, however, we recommend using tcsh.

System Environment Tests

1. First and foremost, it is very important to have a gfortran compiler, as well as gcc and cpp. To test whether these exist on the system, type the following:

   - `which gfortran`
   - `which cpp`
   - `which gcc`

   If you have these installed, you should be given a path for the location of each.

   We recommend using gfortran version 4.4.0 or later. To determine the version of gfortran you have, type:

   ```
   gcc --version
   ```

2. Create a new, clean directory called `Build_WRF`, and another one called `TESTS`.

3. There are a few simple tests that can be run to verify that the fortran compiler is built properly, and that it is compatible with the C compiler. Below is a tar file that contains the tests. Download the tar file and place it in the `TESTS` directory.

   Fortran and C Tests Tar File

   To unpack the tar file, type:

   ```
   tar -xf Fortran_C_tests.tar
   ```

   There are 7 tests available, so start at the top and run through them, one at a time.

   Test #1: Fixed Format Fortran Test: `TEST_1_fortran_only_fixed.f`

   Type the following in the command line:

   ```
   gfortran TEST_1_fortran_only_fixed.f
   ```

   Now type: