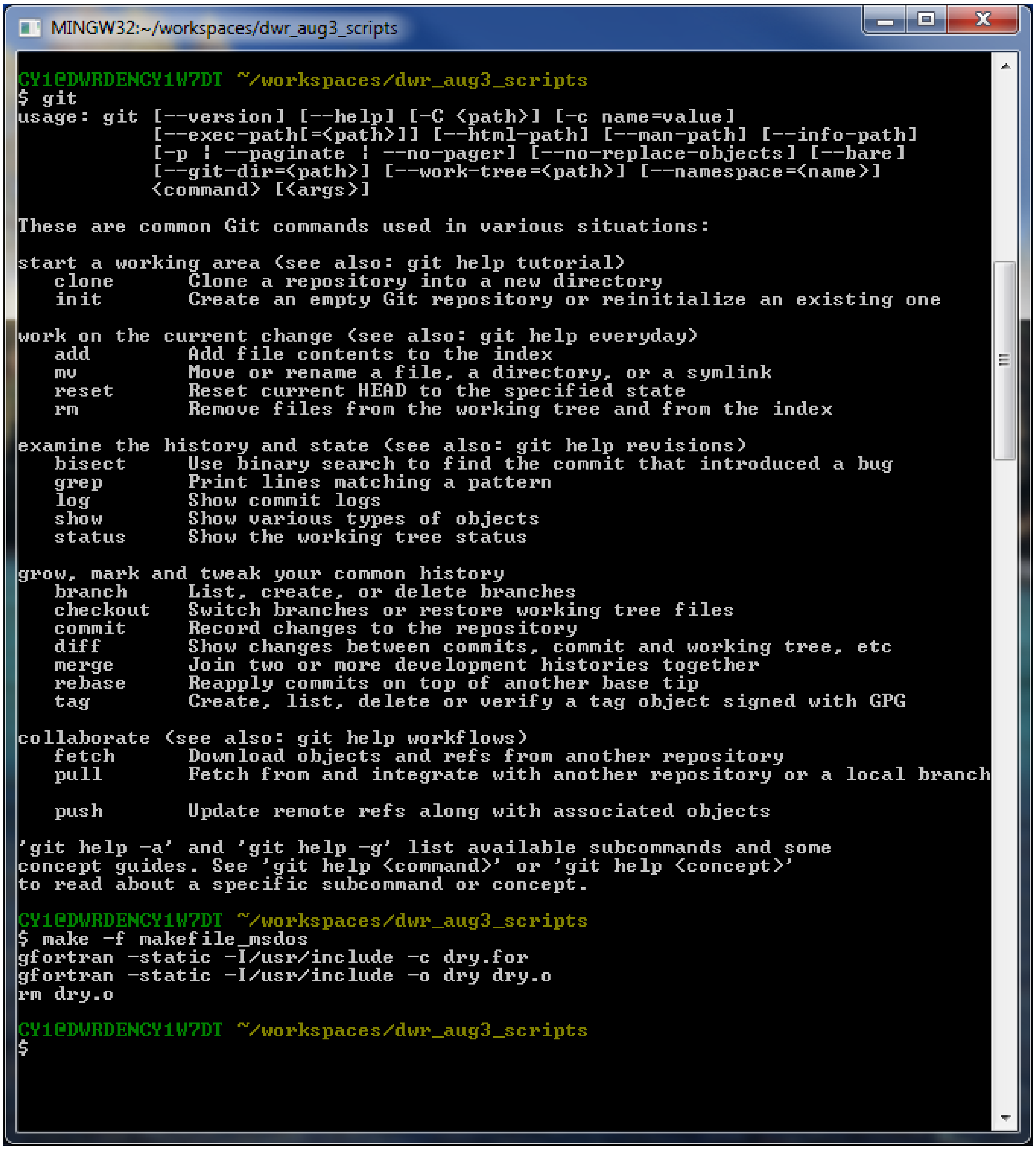
How to Compile a Fortran Program on Windows Using Free Software MinGW

MinGW, a contraction of "Minimalist GNU for Windows", is a minimalist development environment for native Microsoft Windows applications.

1. On Google Search Page, search MinGw
2. On Google Chrome http://www.mingw.org/ Page, Download Installer
3. A file called mingw-get-setup.exe is downloaded to your machine
4. Install it using default options
5. Detailed information on instalation can be found at http://www.mingw.org/wiki/Getting_Started
6. On MinGW Installation Manager window, select the following packages (Mark for Installation)
    a. Mingw-developer-toll
    b. Mingw32-base
    c. Mingw32-gcc-fortan
    d. Msys-base
7. When you have completed selectin and marking, open the Installation menu (on the menu bar), and select the ***Apply Changes*** operation
8. The click the ***Apply*** button to commit them
9. After installing you should ...
    a. Open a Windows Explore window and locate your installation directory (i.e., C:\MinGW)
    b. Below your installation directory, you should find, you should find a directory named "msys" and below which you should find subdirectories "1.0" and "etc"
    c. Within the "etc" directory, there is a file named "fstab"
    d. Open "fstab" with a text editor
    e. Edit the "fstab" and ensure that it contains one line, which reads: "C:\MinGW   /mingw (ensure that there is at least one space, or tab, before the "/mingw" entry
    f. Before you save the file, ensure that there is at least one blank line at the bottom, below all of the entries that may exist, then save and close the file
    g. Create a shortcut for "WinGW Shell" on your desktop and this should invoke the C:\MinGW\msys\1.0\msys.bat script.
    h. Double click this shortcut will then open a command window with the correct environment set up for you, including the correct path references, allowing you to run any of the MinGW applications with that command window
    i. Your MinGW installations should now be ready to use
    j. Download "git" from https://git-scm.com/download/win. For this installation, the latest (2.13.0) 64-bit version of Git for Windows was selected
    k. Installation of Git and it can be found at https://git-scm.com/book/en/v2 for the book
10. Make a directory under C:\MinGW\msys\1.0\home\CY1\workspaces\dwr_aug3_scripts (on Chunming Yu's computer)
11. Save all necessary files (including Fortran files and make files, etc.,) from Jim Brannon in above directory

12. Any Fortran code file can be edited using any test editor and save the change
13. On “MinGW Shell“, go to directory C:\MinGW\msys\1.0\home\CY1\workspaces\dwr_aug3_scripts
14. For compile the new changed Fortran file, type “make –f makefile_msdos”, the updated Fortran code will be compiled (if no changes in Fortran file, the command will not carried out.

Note: The programs were compiled using mingw32-gcc-fortran. Although this is a 32-bit compiler, programs compiled with this software will run on both the 32 and 64-bit versions of Windows.



```
CY1@DWRDENCY1W7DT ~/workspaces/dwr_aug3_scripts
$ git
usage: git [--version] [--help] [-C <path>] [-c name=value]
           [--exec-path[=<path>]] [--html-path] [--man-path] [--info-path]
           [-p | --paginate | --no-pager] [--no-replace-objects] [--bare]
           [--git-dir=<path>] [--work-tree=<path>] [--namespace=<name>]
           <command> [<args>]

These are common Git commands used in various situations:

start a working area (see also: git help tutorial)
   clone      Clone a repository into a new directory
   init       Create an empty Git repository or reinitialize an existing one

work on the current change (see also: git help everyday)
   add        Add file contents to the index
   mv         Move or rename a file, a directory, or a symlink
   reset      Reset current HEAD to the specified state
   rm         Remove files from the working tree and from the index

examine the history and state (see also: git help revisions)
   bisect     Use binary search to find the commit that introduced a bug
   grep       Print lines matching a pattern
   log        Show commit logs
   show       Show various types of objects
   status     Show the working tree status

grow, mark and tweak your common history
   branch     List, create, or delete branches
   checkout   Switch branches or restore working tree files
   commit     Record changes to the repository
   diff       Show changes between commits, commit and working tree, etc
   merge      Join two or more development histories together
   rebase     Reapply commits on top of another base tip
   tag        Create, list, delete or verify a tag object signed with GPG

collaborate (see also: git help workflows)
   fetch      Download objects and refs from another repository
   pull       Fetch from and integrate with another repository or a local branch

   push       Update remote refs along with associated objects

'git help -a' and 'git help -g' list available subcommands and some
concept guides. See 'git help <command>' or 'git help <concept>'
to read about a specific subcommand or concept.

CY1@DWRDENCY1W7DT ~/workspaces/dwr_aug3_scripts
$ make -f makefile_msdos
gfortran -static -I/usr/include -c dry.for
gfortran -static -I/usr/include -o dry dry.o
rm dry.o

CY1@DWRDENCY1W7DT ~/workspaces/dwr_aug3_scripts
$
```