UNIX and netCDF commands
(in appendix, netcdf and idl installation)

Purpose

In this document, you will learn how to work on a UNIX environment with the most used command lines and how to manage with the netCDF files which are commonly used in oceanography.
In appendix A, the step by step installation of the netCDF library is explained depending on your system.
In appendix B, the step by step installation of the IDL library is explained depending on your system.

Linux command lines

If you have never used a **UNIX** environment, well it's not that bad but you will quickly see that everything is done through a terminal, navigating in your computer folders, reading and editing files, compiling and executing codes, plotting results, etc...

Basic commands

Once you have opened a terminal, you will be located in your home directory. To what is the exact path where you are, use the command **pwd** :

```
pwd
```

It should display a path like /home/userAccount

To know which are the files and folders contained in your current path, use **ls** :

```
ls
```

It should display the list of all the files and folders like Documents

To move inside the folder Documents, use **cd** :

```
cd Documents
```

To move outside this folder, it is symbolized by **..** :

```
cd ..
```

Wherever you are, you can easily go back to your home directory, using equivalent commands :

```
cd $HOME
cd ~
cd
```

To copy, move, link, remove a file fileA, you can respectively use the following commands :

```
cp fileA ../fileB
mv fileA ../fileB
ln -s fileA ../fileB
rm fileA
```

HOME is an environment variable that you can print using the command **echo** followed by **$** :

```
echo $HOME
```

To display on the screen the content of a file readme.txt, you can use the command **cat** :

```
cat readme.txt
```

To redirect the output displayed on the screen into a log file, you can use different commands :

```
cat readme.txt > result.out        (to redirect and overwrite the log file result.out)
cat readme2.txt >> result.out      (to redirect and append in the log file result.out)
cat readme | less                  (to display on the screen page by page)
cat readme.txt | tee result.out    (to print on the screen and redirect in the log file result.out)
```

A few keys combinations are commonly used to work in a terminal :

```
Ctrl+C          (to quit a running application)
Ctrl+Shift+C    (to copy the selection)
Ctrl+Shift+V    (to paste the selection)
```

Common applications

The following applications are not all installed by default on your computer. You can usually install it from the official packages of your operating system distribution using apt-get for ubuntu, yum for fedora, port for mac os x.

```
apt-get install package   (ubuntu)
yum install package       (fedora)
port install package      (mac os x)
```

To read a pdf file, you can either use commands like okular, evince or acroread :

```
okular manual.pdf
```

To display/edit a picture, you can use commands like eog, shotwell, gimp :

```
eog figure.jpg
```

To edit a text file, you can use graphical editors like gedit, nedit :

```
gedit readme.txt
```

To edit a text file, you can also use integrated edited like vi (version vim.tiny) :

```
vi readme.txt
```

The usual vi commands are :

```
i           to get in writing mode
esc         to get in reading mode
:q          to quit
:q!         to quit without saving changes
:wq         to quit saving changes
:300        to go to line 300
/author     to find occurences of the word author
:n          to go to next occurrence
:N          to go to the previous occurences
:%s/X/Y/g       to replace all the occurences of X by Y
yy          to copy a line (adding a number before the command to specify the number of lines to handle)
dd          to cut a line (adding a number before the command to specify the number of lines to handle)
pp          to paste the buffer of the copy/cut commands
```

Advanced settings

The user environment can be set up in different manners, first thing is to know in which environment you are working, it is defined via the environment variable SHELL :

```
echo $SHELL
```

If it display /bin/csh, you are using csh environment
If it display /bin/bash, you are using bash environment

The following commands will depend you environment, csh or bash. Your environment is defined by environment variable and aliases. It will be defined in a hidden file stored in your home directory :

```
ls ~/.cshrc
or
ls ~/.bashrc
```

When you change something in this environment file, you have to source the file to propagate the modifications in the environment, otherwise open a new terminal, it will automatically source the file :

```
source ~/.cshrc
or
source ~/.bashrc
```

The syntax to defined the aliases will change depending on csh or bash environment :

```
alias idlww3 idl -rt=rt_visumain.sav     (for csh)
alias idlww3="idl -rt=rt_visumain.sav"  (for bash)
```

The syntax will also change for the environment variables defined in the environment file :

```
setenv OASISDIR /home/$USER/work_oasis3-mct2.0  (for csh)
export OASISDIR=/home/$USER/work_oasis3-mct2.0  (for bash)
```

The main environment variables are the path definitions of the bin folders :

```
setenv PATH $PATH:/usr/local/bin   (for csh)
export PATH=$PATH:/usr/local/bin  (for bash)
```

... and the library folders :

```
setenv LD_LIBRARY_PATH $LD_LIBRARY_PATH:/usr/local/lib   (for csh)
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/usr/local/lib  (for bash)
```

netCDF format

The most common format for a file is the **text file** which is called the **ASCII format** (with extensions .doc, .txt) that you can read with common editors (like word, gedit). The problem with these ASCII files is that they take a lot of space so if you are storing model data or measurements in ASCII format it can fill up your disk really quickly ! The alternative is to store your data in a **binary fil**e like **netCDF format** (with extensions .nc) which will use much less space than the ASCII format thanks to the way that each value is written into the file. On the other hand, you can't read this format with previously cited text editors, for this you have to use specific tools to handle your netCDF files, the easy ones are ncdump and ncview.

The usual structure of a netCDF file is divided in 2 parts, the header and the data. The header is divided into 3 sections which are dimensions, variables, global attributes. Each variable is defined by one or many dimensions and is usually described by one or many attributes.

Here is an example of a netCDF header :

netcdf myfile {

dimensions:
longitude = 89 ;
latitude = 94 ;
time = UNLIMITED ; // (1 currently)

variables:
hs:long_name = "significant height of wind and swell waves" ;
hs:standard_name = "sea_surface_wave_significant_height" ;
hs:units = "m" ;
hs:_FillValue = -32767s ;
hs:scale_factor = 0.002f ;
hs:add_offset = 0.f ;
hs:valid_min = 0 ;
hs:valid_max = 32000 ;

// **global attributes:**
:WAVEWATCH_III_version_number = "6.00" ;
:latitude_resolution = "0.1250000" ;
:longitude_resolution = "0.1250000" ;
:southernmost_latitude = "-38.90000" ;
:northernmost_latitude = "-27.27500" ;
:westernmost_longitude = "12.00000" ;
:easternmost_longitude = "23.00000" ;
:minimum_altitude = "-12000 m" ;
:maximum_altitude = "9000 m" ;
:altitude_resolution = "n/a" ;
:start_date = "2009-09-01 00:00:00" ;
:stop_date = "2009-09-01 00:00:00" ;

Let's now read all the netCDF file using ncdump :

```
ncdump myfile.nc
```

It will transcript all the content of your binary file into text.

Well, you didn't had time to read all the content, you can either show it on your screen page by page:

```
ncdump myfile.nc | less
```

or redirecting all the transcripted content into a text file:

```
ncdump myfile.nc > myfile.txt
```

or both:

```
ncdump myfile.nc | tee myfile.txt
```

If you want to see the content of the header, just use the option '-h':

```
ncdump -h myfile.nc
```

To go further, you choose to display only specific variable values like the time using the option “-v”:

```
ncdump myfile.nc -v time
```

It will give you integer of floating values, not really human readable, so if your time variable has the attribute 'units' correctly defined, you can use the option '-t' to convert the values in dates :

```
ncdump myfile.nc -v time -t
```

Now you can visualize your variables using ncview:

```
ncview myfile.nc
```

More information:
Mickael Accensi (Mickael.Accensi-at-ifremer.fr)

Appendix A - How to install the netCDF library

installation on Ubuntu

To install the netCDF library, you need to have root permissions :

```
sudo apt-get install libhdf5-serial-dev
sudo apt-get install libnetcdf-dev
sudo apt-get install libnetcdff-dev  (only needed for ubuntu version 16.06 and further)
sudo apt-get install netcdf-bin
sudo apt-get install netcdf-dbg
sudo apt-get install netcdf-doc
```

Check that your installation is well done using the nf-config command :

```
nc-config --all
```

It will display all the dependencies and compilers used to compile the netCDF library, check that your fortran compiler (--fc) is well defined, it should be gfortran or ifort or mpif90.

--fc -> gfortran
--fflags -> -g -O2 -I/usr/include
--flibs -> -L/usr/lib -lnetcdff -lnetcdf
--has-f77 -> yes
--has-f90 -> yes

You can also instead extra packages to handle the netCDF files :

```
sudo apt-get install nco
sudo apt-get install ncview
```

installation on Fedora

```
sudo yum install netcdf-fortran-devel.x86_64
sudo ln -s /usr/lib64/gfortran/modules/netcdf.mod   /usr/include
```

The netcdf library should be installed in /usr/lib64/gfortran/modules/
You may need to add this path to the environment variable LD_LIBRARY_PATH

installation on Mac OS X

If you don't have it, you will have to install Macports, depending on your Mac distribution, the installation is explained on this webpage https://www.macports.org/install.php

Once Macports is installed, you can install the netcdf library :

```
sudo port install netcdf
sudo port install gcc48 +gfortran
sudo port install netcdf-fortran
```

The gfortran should be installed in /opt/local/bin/gfortran-mp-4.8. You have to make link it in your local bin folder or add it in your PATH environment variable :

```
sudo ln -s /opt/local/bin/gfortran-mp-4.8 /opt/local/bin/gfortran
or
PATH=$PATH:/opt/local/bin/gfortran-mp-4.8
```

installation on OpenSuse

download the netcdf fortran devel depending on your operating system

for the version 42.1 :

```
wget --cut-dirs=10 ftp://ftp.pbone.net/mirror/ftp5.gwdg.de/pub/opensuse/repositories/home:/NfG84:/cluster/openSUSE_Leap_42.1/x86_64/netcdf-fortran-devel-4.4.2-6.1.x86_64.rpm
```

for other versions, the download links are available at this address :
http://rpm.pbone.net/index.php3?stat=3&limit=1&srodzaj=3&dl=50&search=netcdf-fortran-devel

install the package :

```
sudo zypper install netcdf-fortran-devel-4.4.2-6.1.x86_64.rpm
or
sudo rpm -i netcdf-fortran-devel-4.4.2-6.1.x86_64.rpm
```

Appendix B – How to install the IDL library

Installation on Ubuntu

Prerequisites :

```
apt-get update
apt-get upgrade
apt-get install libxp6
```

If it don't find the libxp6 library, you will need to get it from the archive repos.

```
wget -mnH --cut-dirs=5 http://archive.ubuntu.com/ubuntu/pool/main/libx/libxp/libxp6_1.0.2-1ubuntu1_amd64.deb
sudo dpkg -i libxp6_1.0.2-1ubuntu1_amd64.deb
```

Download the IDL sources :

```
cd /tmp
wget -mnH --cut-dirs=5 ftp://ftp.ifremer.fr/ifremer/ww3/COURS/WAVE_DATA/IDL81/linux
```

Define the destination directory where you will install the IDL softwave, usually it is in /usr/local/itt

```
sudo mkdir /usr/local/itt
```

Run the installation script :

```
cd /tmp/linux
chmod 775 install.sh
sudo ./install.sh
```

Do you accept all of the terms of the preceding license agreement ? (y/n):

```
> y
```

Please enter the directory to contain IDL 8.1 (e.g. "/usr/local/itt ")

```
> /usr/local/itt
```

Do you want to install DICOM Network Services? (y/n)

```
> n
```

Install the above configuration? (y/n):

```
> y
```

Execute the install command now? (y/n):

```
> y
```

Do you wish to continue? (y/n):

```
> y
```

Create the symbolic links described in option 1 above? (y/n):

```
> y
```

Do you want to run the License Wizard? (y/n):

```
> n
```

Installation on Mac OS X

Download the IDL sources :
ftp://ftp.ifremer.fr/ifremer/ww3/COURS/WAVE_DATA/IDL81/mac

Installation on windows

Download the IDL sources :
ftp://ftp.ifremer.fr/ifremer/ww3/COURS/WAVE_DATA/IDL81/windows