**WRF**

**Compilação C e Fortran**

Na pasta “install\_wrf”, torne executável os arquivos .sh: **chmod +x \*.sh**

Execute-os na ordem como usuário sudo:

user@name: **su –**

user@name: **(password)**

1) user@name: **./l\_dpcpp-cpp-compiler\_p\_2023.2.1.8.sh**

Quando abrir a tela do intel oneapi dê: **next>> Accept & customize >> next>>**

Na tela de “change installation directory” coloque o local de instalação como: **/usr/local/intel/oneapi**

**next >> install >> next >>** marca a ultima caixinha **“I do not consent to the collection of my information” >> begin installation**

**close**

2) user@name: **./l\_fortran-compiler\_p\_2023.2.1.8.sh**

Quando abrir a tela do intel oneapi dê: **next>> Accept & customize >> next>>**

Na tela de “change installation directory” coloque o local de instalação como: **/usr/local/intel/oneapi**

**next >>** marca a ultima caixinha **“I do not consent to the collection of my information” >> begin installation**

**close**

3) user@name**: source /usr/local/intel/oneapi/setvars.sh**

4) user@name**: icc** (se tiver compilado o C vai aparecer uma mensagem caso não tenha instalado vai aparecer **“-bash: icc: comando não encontrado”**)

5) user@name**: ifort** (se tiver compilado o Fortran vai aparecer uma mensagem caso não tenha instalado vai aparecer **“-bash: ifort: comando não encontrado”**)

6) Na pasta “install\_wrf” dê: nano instala\_libs\_intel.sh

Verifique se o caminho do source está correto:

source /usr/local/intel/oneapi/setvars.sh

abaixo da linha: mv \*.tar.\* $DEST

ligue as flags colocando o valor = 1 (assim, todas que estiverem com o valor 1 serão instaladas e compiladas, caso não queira instalar alguma, coloque o valor 0)

ZLIB\_FLAG = 1

...

NETCDF\_F\_FLAG = 1

Salve (Ctrl+o) e saia (Ctrl+x)

7) user@name: ./instala\_libs\_intel.sh

**Baixando o WRF, WPS e Geog**

Escolha a pasta onde vai instalar o WRF e crie uma pasta com a versão:

user@name: mkdir WRFv4.6.1

user@name: cd WRFv4.6.1

user@name: /WRFv4.6.1# git clone <https://github.com/wrf-model/WRF.git>

user@name: /WRFv4.6.1# git clone <https://github.com/wrf-model/WPS.git>

user@name: /WRFv4.6.1# wget https://www2.mmm.ucar.edu/wrf/src/wps\_files/geog\_low\_res\_mandatory.tar.gz

user@name: /WRFv4.6.1# tar zxvf geog\_high\_res\_mandatory.tar.gz

user@name: /WRFv4.6.1# mkdir DOMAINS

user@name: /WRFv4.6.1# cd DOMAINS

user@name: /WRFv4.6.1/DOMAINS# mkdir GRADE PREPROC RUN POSPROC (organização para boas práticas)

**Compilando WRF**

Na pasta “install\_wrf” dê: source setenv.sh

Volte para pasta do WRFv4.6.1

user@name: /WRFv4.6.1# cd WRF

user@name: /WRFv4.6.1/WRF# ./configure

Enter selection (1-83): **16**

Compile for nesting? (1=basic, 2=present moves, 3=vortex following) [default 1]: **Enter**

user@name: /WRFv4.6.1/WRF# ./compile em\_real

caso apareça a mensagem:

**-bash: ./compile: cannot execute: required file not found**

user@name: /WRFv4.6.1/WRF# apt install csh,

user@name: /WRFv4.6.1/WRF# ./compile em\_real

**Compilando WPS**

user@name: /WRFv4.6.1/WPS# ./configure

Enter selection [1-40] : 3

Configuration successful. To build the WPS, type: compile

user@name: /WRFv4.6.1/WPS# ./compile

**Instalação do anaconda3**

Execute como usuário sudo:

user@name: **su –**

user@name: **(password)**

vai até o diretório da pasta “install\_wrf” dê: **./anaconda3-2023.09-0-Linux-x86\_64.sh**

leia o contrato (enter)

no final vai aparecer a mensagem:

“Do you accept the license terms: [yes|no]” digite “yes”

[no] >>> yes

Anaconda will now be installed into this location: /root/anaconda3

* Press ENTER to confirm the location
* Press CTRL-C to abort the installation
* Or specify a different location below

(ao final dessa mensagem voce vai colocar o mesmo caminho onde ta instalado o intel/oneapi)

[/root/anaconda3] >>> /usr/local/anaconda3

...

You can undo this by running ‘conda init –reverse $SHELL’? [yes|no]

[no] >>> yes

…

Thank you for installing Anaconda3!

user@name: nano ~/.bashrc

copie o bloco >>> conda initialize <<<

user@name: nano /etc/bash.bashrc

cole no final o bloco >>> conda initialize <<<

e adicione:

export PATH=/usr/local/opengrads:$PATH

source /usr/local/ferret/bin/ferret\_paths\_template.sh

ctrl+o e ctrl+x

teste o grads e o ferret:

user@name: grads

user@name: ferret

user@name: go tutorial

Instalando o grads e o ferret:

Volte na pasta “install\_wrf”:

Descompacte o .tar: tar zxfv opengrads.tar.gz

Descompacte o .tar: tar zxfv ferret7.6.tar.gz

Execute como usuário sudo:

user@name: **su –**

user@name: **(password)**

user@name: mv ferret /usr/local

user@name: mv opengrads /usr/local