

Fortran Skeleton Package (FSP)

Nicolas.Grima(at)univ-brest.fr

24 janvier 2013

Table des matières

1 Introduction.....	3
2 Télécharger et installer FSP.....	3
3 Mode rapide : utilisation du « Graphique User Interface ».....	4
4 FSP à la main (mode débutant).....	9
4.1 Sans NetCDF.....	9
4.2 Avec NetCDF.....	13
4.3 À vous de développer votre application.....	17
5 FSP en détail (mode initié).....	17
5.1 Les autotools.....	17
5.1.a la configuration.....	17
5.1.b la compilation.....	17
5.1.c l'installation.....	17
5.2 Les routines Fortran.....	17
5.2.a le programme principal.....	17
5.2.b les modules.....	17
5.2.c Le fichier namelist.....	17
5.2.d Écrire rapidement des fichiers NetCDF.....	17
5.3 Distribuer mon application.....	17

1 Introduction

Le développement de codes sources informatiques nécessite généralement une base commune de routines et un mode de compilation spécifique au langage. C'est le cas pour le langage Fortran qui sera dans ce document notre langage de référence. Un des objectifs du FSP est d'avoir une sorte de modèle, ou de squelette, à partir duquel on va pouvoir facilement développer une application en Fortran.

Dans le paquet FSP on trouvera un ensemble d'outils et de routines dont l'objectif est de simplifier la phase de mise en place d'un programme informatique en Fortran.

Ce paquet inclut tout les outils standards nécessaires, généralement appelés « autotools », pour une simplification de la compilation et l'installation de l'application. Il inclut également des routines (modules Fortran) de base qui serviront de références pour la précision des variables, la définition des constantes, la gestion de la mémoire dynamique et l'écriture simplifiée de fichiers au format NetCDF. Une convention d'écriture du code source en langage Fortran est également proposée, n'entraînant évidemment aucune obligation.

L'auteur de ce paquet espère qu'il permettra de simplifier la phase du début de développement d'un code en langage Fortran.

2 Télécharger et installer FSP

Le nom du paquet est de la forme :

```
fsp-x.x.x.tar.gz
```

où

```
x.x.x
```

représente le numéro de version.

Après avoir récupéré le paquet FSP, il faut le décompresser et le désarchiver.

Pour cela on utilise la commande suivante :

```
tar xzvf fsp-x.x.x.tar.gz
```

vous devriez obtenir à l'écran la séquence suivante :

```
fsp-0.0.0/  
fsp-0.0.0/confdb/  
fsp-0.0.0/confdb/config.guess  
fsp-0.0.0/confdb/config.sub  
fsp-0.0.0/confdb/install-sh  
fsp-0.0.0/confdb/missing  
fsp-0.0.0/confdb/mkinstalldirs  
fsp-0.0.0/README  
fsp-0.0.0/configure.ac  
fsp-0.0.0/aclocal.m4  
fsp-0.0.0/Makefile.am  
fsp-0.0.0/Makefile.in  
fsp-0.0.0/configure  
fsp-0.0.0/AUTHORS  
fsp-0.0.0/COPYING  
fsp-0.0.0/ChangeLog  
fsp-0.0.0/INSTALL  
fsp-0.0.0/NEWS
```

```
fsp-0.0.0/src/  
fsp-0.0.0/src/Makefile.am  
fsp-0.0.0/src/Makefile.in  
fsp-0.0.0/src/mod_precision.f90  
fsp-0.0.0/src/mod_configure.f90  
fsp-0.0.0/src/mod_cst.f90  
fsp-0.0.0/src/mod_lun.f90  
fsp-0.0.0/src/mod_memory.f90  
fsp-0.0.0/src/mod_inout.f90  
fsp-0.0.0/src/mod_namelist.f90  
fsp-0.0.0/src/prg_fsp.f90  
fsp-0.0.0/src/mod_netcdf_write_fast.f90  
fsp-0.0.0/src/namelist  
fsp-0.0.0/src/model_fortran/  
fsp-0.0.0/src/model_fortran/Makefile.am  
fsp-0.0.0/src/model_fortran/Makefile.in  
fsp-0.0.0/src/model_fortran/mod_00_fortran_model.f90  
fsp-0.0.0/src/options/  
fsp-0.0.0/src/options/Makefile.am  
fsp-0.0.0/src/options/Makefile.in  
fsp-0.0.0/src/options/Makefile.am_fundamental  
fsp-0.0.0/src/options/Makefile.am_netcdf  
fsp-0.0.0/src/options/mod_inout.f90_fundamental  
fsp-0.0.0/src/options/mod_inout.f90_netcdf  
fsp-0.0.0/doc/  
fsp-0.0.0/doc/Makefile.am  
fsp-0.0.0/doc/Makefile.in
```

Le paquet FSP est désormais installé sur votre machine. Placez vous maintenant dans le répertoire `fsp-0.0.0`.

```
cd fsp-0.0.0
```

3 Mode rapide : utilisation du « Graphique User Interface »

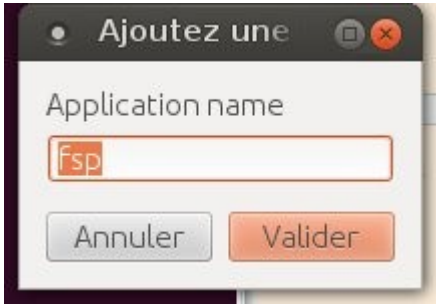


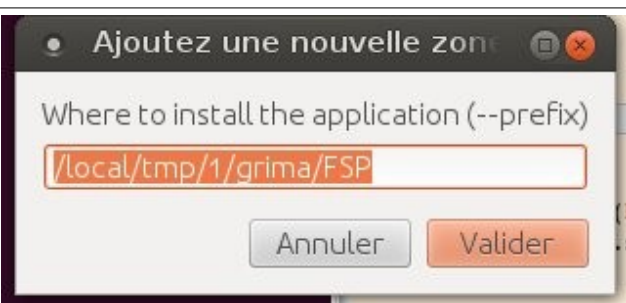
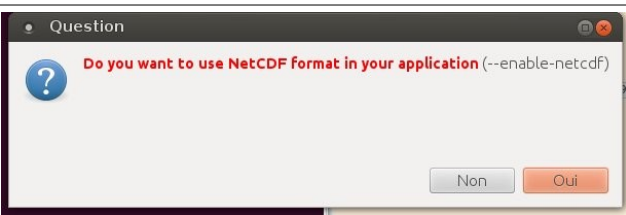
On suppose que vous avez suivi toutes les étapes du chapitre 2 et qu'elles se sont correctement déroulées.

Dans ce chapitre on va utiliser une interface graphique pour mettre en place votre future application à partir du paquet FSP.

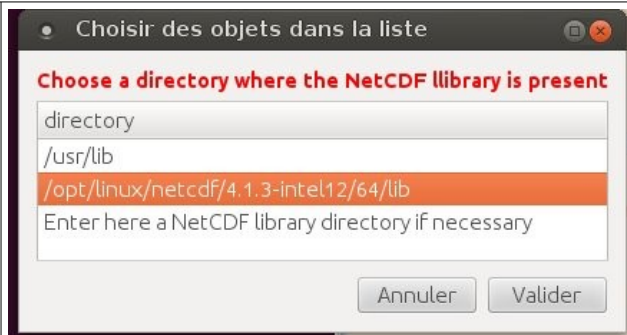
Placez vous dans le répertoire `fsp-0.0.0/gui`.

Puis soumettez la commande `./fsp_gui.sh`.

Vous devriez obtenir une succession de boites de dialogue qui vous demanderons de rentrer ou de choisir des options pour votre future application.

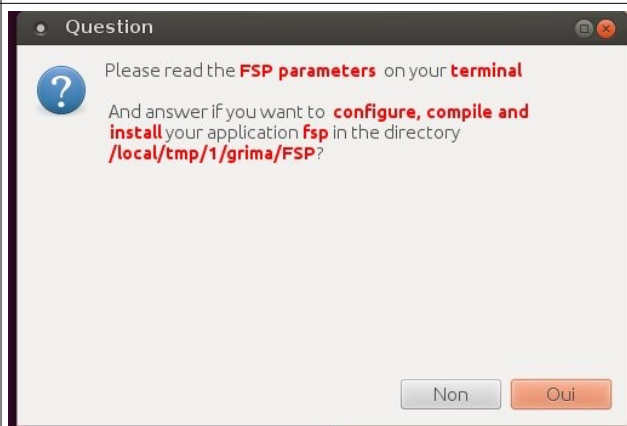
<p>1/ Entrer le nom de la future application [fsp].</p>	
<p>2/ Entrer votre adresse mèl [author@mail.org].</p>	
<p>3/ Entrer l'URL du site web associé à l'application [http://www.fsp.org].</p>	
<p>4/ Entrer le nom du répertoire où sera installé l'exécutable de l'application [<code>`pwd` ; cd ../../</code>].</p>	
<p>5/ Ici vous déciderez si votre application utilisera la librairie NetCDF pour écrire et lire des fichiers à ce format. Répondre oui si vous comptez utiliser la librairie NetCDF, non dans le cas inverse [oui].</p>	

5bis/ Si vous comptez utiliser la bibliothèque NetCDF alors une fenêtre de dialogue vous demandera le répertoire où se trouve cette bibliothèque parmi un choix que fsp_gui.sh aura trouvé (regarde dans LD_LIBRARY_PATH). Si aucun répertoire correspond, vous pouvez le rentrer à la main en remplaçant la ligne « Enter here ... necessary » par le nom du répertoire que vous avez choisi. ATTENTION : généralement la bibliothèque NetCDF doit avoir été compilé avec le même compilateur Fortran que votre future application.



6/ A partir de ce point, vous devez lire sur votre terminal l'ensemble des paramètres que vous avez sélectionné ou défini. Ensuite vous pouvez, soit stopper ici l'outil GUI et prendre la main sur le reste des commandes, soit vous laisser l'outil GUI lancer la configuration (./configure), la compilation (make) et l'installation (make install).

Il est recommandé de répondre oui à cette question si les paramètres affichés vous paraissent corrects [oui].



Voici au final ce qui devrait s'afficher sur le terminal où vous avez soumis l'outil fsp_gui.sh. En vert les paramètres que vous avez sélectionné ou entré. Puis ensuite l'ensemble des commandes de configuration, compilation et installation. À la fin, si tout c'est correctement déroulé, l'exécutable du squelette de votre application doit être disponible dans le répertoire que vous avez défini à l'étape .

```
*****
*   Fortran Skeleton Package Gui results   *
*****
Application Name      : fsp
Application Version    : 0.0.0
Author Mail           : author@mail.org
Application URL        : http://www.fsp.org
--prefix              : /local/tmp/1/grima/FSP
--enable-netcdf        : yes
  - NETCDF_LIB         : /opt/linux/netcdf/4.1.3-intel12/64/lib
  - NETCDF_INC         : /opt/linux/netcdf/4.1.3-intel12/64/include
*****

Creating the configure.ac file:

Creating the prg_fsp.f90 file:

Creating Makefile.am files:

Creating configure file:

setenv NETCDF_LIB /opt/linux/netcdf/4.1.3-intel12/64/lib
setenv NETCDF_INC /opt/linux/netcdf/4.1.3-intel12/64/include

./configure --enable-netcdf
make clean
make install

*****
* Your application fsp should be installed
* in the directory /local/tmp/1/grima/FSP/bin.
* You can go to this directory and submit ./prg_{applicationName}
* or make a symlink to it, or enter /local/tmp/1/grima/FSP
* in you PATH environment variable.
*****

Please check the file sfp_gui.log...

xncd1 /local/tmp/1/grima/FSP/fsp-0.0.0/gui > █
```

Vous retrouverez l'ensemble des sorties à l'écran et le détail des sorties des commandes dans le

fichier `fsp_gui.log` qui devra être présent dans le répertoire `fsp-0.0.0/gui`.

Il est possible de se rendre dans le répertoire d'installation et de soumettre l'application :

```
xncd1 /local/tmp/1/grima/FSP/fsp-0.0.0/gui > cd /local/tmp/1/grima/FSP/bin
xncd1 /local/tmp/1/grima/FSP/bin > fsp

=====
= -o0)   fsp v0.0.0      (0o- =
=====

=====
= Machine precision =
=====
-          iprec:          4
-          rprec:          8
-    range(iprec):          9
-    range(rprec):        307
- Precision(rprec):        15
- Spacing(rprec):  2.220446049250313E-016
- EPSILON(rprec):  2.220446049250313E-016

=====
= NAMELIST =
=====
--- Successful Opening ---

- Reading FSP item:
- ifsp   =          15
- rfsp   =  3.140000000000000
- lfsp   = T
- cfsp   =fortran skeleton package

- Reading PARAM item:
- iparam =          10
- rparam = 14.30000000000000
- lparam = F
- cparam =an example of namelist char parameter

=====
= Memory Allocations =
=====
- dummy_i2D
- dummy_r2D

=====
= Memory DeAllocations =
=====
- dummy_i2D
- dummy_r2D

Maximum of allocated memory is (in octets):          2400

=====
= -o0)   fsp v0.0.0      (0o- =
=====
```


À vous maintenant de vous rendre dans le répertoire `fsp-0.0.0/src` pour éditer le fichier programme (`prg_fsp.f90`) et dit écrire votre code source, vos modules en Fortran.

Si vous créez de nouveaux fichiers il vous faudra modifier le fichier `Makefile.am` (lien symbolique) du répertoire `fsp-0.0.0/src`, afin d'y ajouter vos modules avec les autres modules

Si vous modifiez le fichier `Makefile.am` il sera nécessaire de retourner dans le répertoire `fsp-0.0.0` et de relancer la commande `./configure`.

Il est fortement conseiller de lire les chapitres suivants pour bien comprendre les mécanismes de la configuration et de la compilation.

4 FSP à la main (mode débutant)

On suppose que vous avez suivi toutes les étape du chapitre 2 et qu'elles se sont correctement déroulée.

Dans ce chapitre on va rapidement tester si le paquet est correctement installé et qu'il fonctionne sur votre ordinateur et son environnement.

Dans les chapitres suivants nous détaillerons tout ce qui ce cache derrière chacune des commandes suivantes et la façon dont on pourra les influencer.

4.1 Sans NetCDF

La première commande à soumettre est :

```
./configure
```

Cette commande va inspecter votre ordinateur à l'aide des informations contenues dans la variable d'environnement «`PATH` ». elle va entre autre chercher le compilateur fortran disponible.

Voici une illustration de ce que vous devriez obtenir, avec évidemment des résultats différents pour les variables correspondant à votre environnement.

```
$> ./configure
checking for a BSD-compatible install... /usr/bin/install -c
checking whether build environment is sane... yes
checking for a thread-safe mkdir -p... /bin/mkdir -p
checking for gawk... gawk
checking whether make sets $(MAKE)... yes
checking build system type... x86_64-unknown-linux-gnu
checking host system type... x86_64-unknown-linux-gnu
checking for ifort... no
checking for gfortran... gfortran
checking whether the Fortran compiler works... yes
checking for Fortran compiler default output file name... a.out
checking for suffix of executables...
checking whether we are cross compiling... no
checking for suffix of object files... o
checking whether we are using the GNU Fortran compiler... yes
checking whether gfortran accepts -g... yes
configure:  -> NORMAL MODE
checking for Fortran flag to compile .f90 files... none
checking for Fortran flag needed to accept free-form source... none
checking for Fortran compiler can handle Fortran-90 code... yes
```

```

configure: generating makefiles
configure: creating ./config.status
config.status: creating Makefile
config.status: creating doc/Makefile
config.status: creating src/Makefile
config.status: creating src/options/Makefile
config.status: creating src/model_fortran/Makefile

Configuration Parameters:
-----
fsp's version..... 0.0.0
prefix..... ../..
FC..... gfortran
FCFLAGS..... -g -O2 -frecord-marker=4
FCFLAGS_f90.....
LDFLAGS..... -g -O2 -frecord-marker=4
LIBS.....
ENABLE_OPTIMIZATION.. normal
ENABLE_NETCDF..... no
ENABLE_PROFILING..... no
ENABLE_QUAD..... no
host..... x86_64-unknown-linux-gnu

Mon Jan 21 15:31:58 CET 2013

*****
*   Fortran Skeleton Package (fsp)'s configuration is completed   *
*   You are now ready to run 'make clean', 'make' and 'make install'*
*****

```

On suppose que tout c'est bien déroulé, si ce n'est pas le cas reportez aux chapitres suivants sur les autotools.

À partir d'ici, les fichier « Makefile » on été généré avec les informations nécessaires pour que la compilation se déroule correctement. Le compilateur Fortran a été sélectionné (FC).

Entrez les commande suivante pour compiler l'application et l'installer dans un répertoire bin qui se trouvera dans le répertoire au dessus de celui d'où vous venez d'exécuter la commande ./configure.

```

$> make clean
$> make install
$> ls -rtl ../bin

```

Vous devriez avoir dans le répertoire ../bin le programme prg_fsp et un fichier namelist. (nous détaillerons plus loin le rôle de ce fichier).

```

-rwxr-xr-x 1 login lpo 44211 month 21 15:45 prg_fsp
-rwxr-xr-x 1 login lpo   626 month 21 15:45 namelist

```

Félicitation, vous venez d'installer le squelette de votre future application Fortran.

Si Vous désiriez l'installer dans un autre répertoire, il suffit d'ajouter une option (--prefix) à la

commande `./configure` pour lui indiquer le répertoire où seront installés les fichiers.

Voici un exemple (on suppose que vous êtes dans le répertoire `fsp-x.x.x`) :

```
$> ./configure --prefix=/home/login
... (informations volontairement coupées)
Configuration Parameters:
-----
fsp's version..... 0.0.0
prefix..... /home/login
FC..... gfortran
FCFLAGS..... -g -O2 -frecord-marker=4
FCFLAGS_f90.....
LDFLAGS..... -g -O2 -frecord-marker=4
LIBS.....
ENABLE_OPTIMIZATION.. normal
ENABLE_NETCDF..... no
ENABLE_PROFILING..... no
ENABLE_QUAD..... no
host..... x86_64-unknown-linux-gnu

Mon Jan 21 15:56:23 CET 2013

*****
* Fortran Skeleton Package (fsp)'s configuration is completed *
* You are now ready to run 'make clean', 'make' and 'make install'*
*****
```

On peut constater que le paramètre `prefix` a maintenant la valeur `'/home/login'`.

Vous n'avez plus qu'à soumettre les commandes suivantes pour que l'application soit installée dans le répertoire `'/home/login/bin'`.

```
$> make clean
$> make install
$> ls -rtl /home/login/bin
-rwxr-xr-x 1 login lpo      44211 month 21 15:56 prg_fsp
-rwxr-xr-x 1 login lpo      626 month 21 15:56 namelist
(et peut être d'autres applications...)
```

Plaçons nous dans le répertoire où le programme `fsp` (votre future application).

```
$> cd /home/login/bin
```

Soumettons le programme :

```
$> ./prg_fsp
=====
= -o0)      prg_fsp   v0.0.0      (0o- =
=====

=====
= Machine precision =
=====
```

```

-      iprec:      4
-      rprec:      8
-      range(iprec):      9
-      range(rprec):      307
- Precision(rprec):      15
- Spacing(rprec):  2.22044604925031308E-016
- EPSILON(rprec):  2.22044604925031308E-016

=====
= NAMELIST =
=====
--- Successful Opening ---

- Reading FSP item:
- ifsp   =      15
- rfsp   =   3.140000000000000001
- lfsp   = T
- cfsp   =fortran skeleton package

- Reading PARAM item:
- iparam =      10
- rparam =  14.300000000000000001
- lparam = F
- cparam =an example of namelist char parameter

=====
= Memory Allocations =
=====
- dummy_i2D
- dummy_r2D

=====
= Memory DeAllocations =
=====
- dummy_i2D
- dummy_r2D

Maximum of allocated memory is (in octets):      2400

=====
= -o0)      prg_fsp   v0.0.0      (0o- =
=====

```

Le programme a bien fonctionné.

Listons les fichiers dans ce répertoire :

```

$> ls -rtl
-rwxr-xr-x 1 login lpo 44211 month 21 16:24 prg_fsp
-rwxr-xr-x 1 login lpo   648 month 21 16:33 namelist
-rw-r--r-- 1 login lpo   546 month 21 16:33 memory.log

```

Un nouveau fichier a été créé avec le contenu des allocations et déallocation dynamique de mémoire :

```

$> more memory.log

```

Total	Real	Integer	Array size	Array type	Array Name	Routine call
0	0	0	0	-----		
1200	0	1200	1200	real array_i2D	sub_memory_allocate_i2D	
2400	1200	1200	1200	integer array_r2D	sub_memory_allocate_r2D	
1200	1200	0	-1200	real array_i2D	sub_memory_deallocate_i2D	
0	0	0	-1200	integer array_r2D	sub_memory_deallocate_r2D	

4.2 Avec NetCDF

Le paquet FSP est fourni de base avec la possibilité d'utiliser le format NetCDF¹ comme format d'écriture des données dans des fichiers respectant ce format (fichiers qui généralement se terminent par .nc).

Nous supposons ici qu'a été installer correctement la librairie NetCDF avec le même compilateur que celui que vous allez utiliser pour compiler votre application.

Si vous êtes sur une distribution Linux et en particulier Ubuntu², il est préférable dans un premier temps d'utiliser gfortran³ comme compilateur et la librairie NetCDF qui s'installe par défaut avec gcc⁴. Dans ce cas, le paquet FSP suppose par défaut que la librairie NetCDF est installée dans le répertoire /usr/lib et les fichiers include dans le répertoire /usr/inc.

On peut le vérifier rapidement, à l'aide des commande suivantes :

```
$> which gfortran
/usr/bin/gfortran
$> ls -rtl /usr/lib/libnetcdf*
...
-rw-r--r-- 1 root root 538222 nov. 7 2011 /usr/lib/libnetcdf.a
-rw-r--r-- 1 root root 1770536 nov. 7 2011 /usr/lib/libnetcdf.a
...
$> ls -rtl /usr/include/netcdf*
...
-rw-r--r-- 1 root root 309303 nov. 7 2011 /usr/include/netcdf.mod
...
```

On suppose que nous sommes dans le répertoire fsp-x.x.x et on soumet la commande :

```
$> ./configure --enable-netcdf
```

Voici un résultat que l'on peut obtenir à l'écran :

```
...
Configuration Parameters:
-----
fsp's version..... 0.0.0
prefix..... ../..
```

1 <http://www.unidata.ucar.edu/software/netcdf/>

2 <http://www.ubuntu-fr.org/>

3 <http://gcc.gnu.org>

4 <http://gcc.gnu.org>

```

FC..... gfortran
FCFLAGS..... -g -O2 -frecord-marker=4 -I/usr/include
FCFLAGS_f90.....
LDFLAGS..... -g -O2 -frecord-marker=4
LIBS..... -L/usr/lib -lnetcdff -lnetcdf
ENABLE_OPTIMIZATION.. normal
ENABLE_NETCDF..... yes
  -NETCDF_LIB..... /usr/lib
  -NETCDF_INC..... /usr/include
ENABLE_PROFILING..... no
ENABLE_QUAD..... no
host..... x86_64-unknown-linux-gnu

Mon Jan 21 16:16:49 CET 2013

*****
*   Fortran Skeleton Package (fsp)'s configuration is completed   *
*   You are now ready to run 'make clean', 'make' and 'make install'*
*****

```

On constate que la librairie NetCDF a été pris en compte et est indiqué à l'écran où elle est située ainsi que le fichier include (netcdf.mod).

On soumet ensuite les mêmes commandes que précédemment :

```

$> make clean
$> make install

```

On peut évidemment mettre plusieurs options à la suite de la commande ./configure :

```
./configure --prefix=/home/login --enable-netcdf
```

On obtient dans ce cas :

```

...
Configuration Parameters:
-----
fsp's version..... 0.0.0
prefix..... /home/login
FC..... gfortran
FCFLAGS..... -g -O2 -frecord-marker=4 -I/usr/include
FCFLAGS_f90.....
LDFLAGS..... -g -O2 -frecord-marker=4
LIBS..... -L/usr/lib -lnetcdff -lnetcdf
ENABLE_OPTIMIZATION.. normal
ENABLE_NETCDF..... yes
  -NETCDF_LIB..... /usr/lib
  -NETCDF_INC..... /usr/include
ENABLE_PROFILING..... no
ENABLE_QUAD..... no
host..... x86_64-unknown-linux-gnu

```

```
Mon Jan 21 16:20:24 CET 2013
```

```
*****
*   Fortran Skeleton Package (fsp)'s configuration is completed   *
*   You are now ready to run 'make clean', 'make' and 'make install'*
*****
```

Plaçons nous tout d'abord dans le répertoire des fichiers source de l'application :

```
$> cd src
```

Éditons le fichier `prg_fsp.f90` pour retirer les commentaires devant les lignes représentées ci-dessous :

```
...
!=====!
!- NETCDF -!
!=====!
!!$ CALL sub_netcdf_write_fast(dummy_i2D)
!!$ CALL sub_netcdf_write_fast      ( &
!!$     dummy_r2D                    , & ! 2D array of reals.
!!$     variable_name                = 'dummy_r2D' , & ! netcdf variable name (opt).
!!$     netcdf_file_name             = 'dummy_r2D.nc', & ! netcdf file name (opt).
!!$     nb_netcdf_file_name = 0      )    ! add number in the netcdf file name (opt).
...

```

Votre code doit désormais être de la forme :

```
...
!=====!
!- NETCDF -!
!=====!
CALL sub_netcdf_write_fast(dummy_i2D)
CALL sub_netcdf_write_fast      ( &
    dummy_r2D                    , & ! 2D array of reals.
    variable_name                = 'dummy_r2D' , & ! netcdf variable name (opt).
    netcdf_file_name             = 'dummy_r2D.nc', & ! netcdf file name (opt).
    nb_netcdf_file_name = 0      )    ! add number in the netcdf file name (opt).
...

```

Restons dans le même répertoire et relançons la compilation et l'installation. On peut facilement le faire en soumettant seulement la commande suivante :

```
$> make install
```

Plaçons nous dans le répertoire où se trouve le programme `prg_fsp` (votre future application).

```
$> cd /home/login/bin
```

Soumettons le programme :

```
$> ./prg_fsp
=====
= -o0)      prg_fsp      v0.0.0      (0o- =
=====
```



```

=====
= Machine precision =
=====
-      iprec:      4
-      rprec:      8
-      range(iprec):      9
-      range(rprec):      307
- Precision(rprec):      15
- Spacing(rprec):  2.22044604925031308E-016
- EPSILON(rprec):  2.22044604925031308E-016

=====
= NAMELIST =
=====
--- Successful Opening ---

- Reading FSP item:
- ifsp =      15
- rfsp =  3.140000000000000001
- lfsp = T
- cfsp =fortran skeleton package

- Reading PARAM item:
- iparam =      10
- rparam =  14.30000000000000001
- lparam = F
- cparam =an example of namelist char parameter

=====
= Memory Allocations =
=====
- dummy_i2D
- dummy_r2D

=====
= Memory DeAllocations =
=====
- dummy_i2D
- dummy_r2D

Maximum of allocated memory is (in octets):      2400

=====
= -o0)      prg_fsp      v0.0.0      (0o- =
=====

```

Le programme a bien fonctionné et cette fois ci en utilisant la bibliothèque NetCDF.
Listons les fichiers dans ce répertoire :

```

$> ls -rtl
-rwxr-xr-x 1 login lpo 72044 month 21 17:01 prg_fsp
-rwxr-xr-x 1 login lpo   652 month 21 17:01 namelist
-rw-r--r-- 1 login lpo  1308 month 21 17:01 netcdf_file.nc
-rw-r--r-- 1 login lpo   546 month 21 17:01 memory.log

```

```
-rw-r--r-- 1 login lpo 1308 month 21 17:01 dummy_r2D.0000.nc
```

Deux nouveaux fichiers au format NetCDF viennent d'être créés, ils contiennent chacun un tableau à deux dimensions, d'entiers pour le fichier `netcdf_file.nc` et de réels pour le fichier `dummy_r2D.0000.nc`.

4.3 À vous de développer votre application

Le programme peut maintenant être modifier à votre guise pour en faire votre application.

Pour cela, placez vous dans le répertoire `fsp-x.x.x/src` et éditez le fichier `prg_fsp.f90`. C'est le programme principal à partir duquel vous allez appeler les modules que vous développerez. Dans la mesure du possible essayer dans ce programme principal de ne faire apparaître que la structure algorithmique de votre application. Les calculs, lectures de fichiers, etc devons idéalement être écrits dans des fichiers modules et appelés par le programme principale via des routines ou fonctions.

Inspirez vous des routines déjà présentes dans le répertoire `src`, comme les modules `mod_memory.f90`, `mod_netcdf_write_fast.f90`, `mod_namelist.f90`, etc...

vous trouverez un modèle de fichier Fortran module dans le répertoire `src/model_fortran` :

```
mod_00_fortran_model.f90
```

Il est évidemment préférable de lire les chapitres suivants si vous n'êtes pas familier avec les autotools et le développement de programme Fortran, le fichier `namelist`, le format NetCDF, etc...

Bon développement :-)

5 FSP en détail (mode initié)

5.1 Les autotools

5.1.a la configuration

5.1.b la compilation

5.1.c l'installation

5.2 Les routines Fortran

5.2.a le programme principal

5.2.b les modules

5.2.c Le fichier `namelist`

5.2.d Écrire rapidement des fichiers NetCDF

5.3 Distribuer mon application

Avec les « autotools » il est très simple de mettre à disposition votre application sous la forme d'un fichier archivé et compressé. Pour cela il suffit de soumettre la commande suivante, dans le répertoire où se trouve le fichier `configure` :

```
$> make clean
```

```
$> make dist-gzip
...
$> ls -rtl
...
-rw-r--r-- 1 me lpo 111612 month 21 17:11 votre-application-x.x.x.tar.gz
```

Vous obtiendrez dans le répertoire un fichier avec les extensions `votre_application-x.x.x.tar.gz`. Vous pouvez maintenant mettre à disposition ce fichier archive pour que d'autres personnes utilisent votre programme.