cours 9

Erreurs

1 Methodologie

Les compilateurs sous unix ne detectent pas des incoherences entre les arguments d'une subroutine et la facon dont elle est appelee. Pour minimiser ces erreurs a l'execution qui se traduisent le plus souvent par un core-dump sans autre explication, il est essentiel de copier-coller la ligne de definition de la subroutine pour generer la ligne d'appel. Cela permettra d'eviter les permutations de variables..

Utiliser aussi les fichiers include pour eviter d'editer le meme bloc common plusieurs fois.

2 Erreurs d'algorithme

- Initialisez toutes les variables. Assurez vous que les variables dans les subroutines et les boucles sont bien reinitialisees entre chaque usage. Fixez les constantes par l'instruction data (compilation) et les variables a l'execution. S'il y a un debugger utilisez le.
- Attention aux erreurs d'une unite. Assurez vous que les instructions sont effectuées le bon nombre de fois et que les tests de comparaison vont au bon endroit lors d'egalites.
- Attention aux depassements de bornes dans les tableaux. Il existe une option de compilation pour tester ca.
- Ne mettez pas plusieurs instructions de sortie dans une boucle. Mettez les tests de sortie ensemble et pres du haut de la boucle.
- Testez votre programme a ses bords internes, Ceci doit etre fait avant l'execution et comme un test. Demandez vous si chaque boucle peut etre faite 0 fois.
- Attention aux divisions par 0. Ainsi le resultat de division de 2 entiers doit toujours etre teste avant d'etre utilise comme diviseur (exemple: echantillonage pour un probleme d'evolution).
- Attention aux depassements de capacite lors de certains calculs. Rappelez vous l'exemple du C_n^p.
- Attention aux programmes qui bouclent. Il faut toujours prevoir la fin de l'execution.
- Pas de boucles sur des indices reels. Ne testez pas des reels pour l'egalite.
- Pour eviter les erreurs de lecture. Utilisez des entrees-sorties en format libre et respectez le type de donnees. Si le programme doit lire 3 entiers et que vous lui donnez 2 reels et un entier, il vous donne une erreur.

- Attention aux erreurs lors de sous programmes. Il peut manquer des parametres, ou ceux ci sont du mauvais type. Attention aux passages de tableaux. Pour des matrices il faut toujours passer les dimensions maximales. Attention aux modifications de constantes par un sous-programme.

3 Erreurs de syntaxe

Le plus souvent le compilateur donne des explications. IL suffit de se reporter a la ligne indiquee pour decouvrir le probleme, parenthese non fermee, fonction non definie, ...

D'autres erreurs sont plus difficiles a detecter. Ainsi des boucles non fermees ou des structures if,elseif,else,endif non redigees correctement. On s'attachera a soigner ces elements du programme.

On testera la syntaxe de chaque subroutine separement en utilisant
f77 -c sub.f

4 Principales erreurs de make

Don't know how to make target
Le Makefile ne contient pas de regles implicites ou explicites de construction de target.

Too many comman lines for name
Il y a plus d'une seule definition de name, une seule est possible.

target **up to date**
l'objet a creer est cree, pas d'action de make.

Must be a separator on rules line n
Il manque un TAB a la ligne n.

Warning: macro **changed after being used**
Une macro a ete redfinie apers avoir ete utilisee. Les consequences ne sont pas claires. A ne pas faire donc!

5 Erreurs de compilation et d'edition de liens

- flag : arg missing
- flag : too small
Il manque la specification - flag à la commande f77.

bad flag mauvaise option

file : Bad magic number
Le fichier n'a pas le bon type.
exemple: linker un programme à un fichier texte

symbol: multiply defined
Le symbole a ete defini dans plusieurs fichiers.
exemple: link de deux subroutines portant le meme nom.

Read failed Un des fichiers avec lequel on linke est vide.

Undefined: sub Il manque la subroutine sub. Il faut alors la chercher dans un autre objet ou dans une bibliotheque.

6 Erreurs de calcul

Certaines operations donnent des resultats qui ne peuvent etre representes sous forme de nombres. On affecte alors a la variable concernee les valeurs ci-dessous en donnant un message.

Inf $+\infty$ Overflow
-Inf $-\infty$ Underflow
NaN not a number, Invalid operation, operation illegale (ex $\sqrt{-1}$)

Ce quantites obeissent a des regles d'operation bien precises Inf $\pm$ x = Inf
Inf - Inf = NaN
Inf * 0 = NaN

N'importe quelle operation sur un NaN donne un NaN

Exemple

```
program tst4
parameter(n=39)
real x , y , z

y=0.
x = 1. / y
write(*,*)'1/0  =', x

x = 10.**n
write(*,*)'10**n =', x

y = -10.**n
write(*,*)'-10**n =', y

z= x + y
write(*,*)'10**n-10**n =',z

x = 10.**(-2*n)
write(*,*)'10**(-2*n) =', x
```

```
      stop
      end

---------------execution-----------------------
tst4
 1/0  =  INF
 10**n =  INF
 -10**n = -INF
 10**n-10**n =  NAN
 10**(-2*n) =  0.
```

7 Erreurs d'execution

100 *Error in format* Une entree sortie illegale s'est produite durant l'execution.

101 *Illegal unit number* Le programme utilise un numero d'unité exterieur a [0,99].

102 *Formatted I/O not allowed* Le programme a essaye d'ecrire en formatte dans un fichier declare **'form'=unformatted**.

103 *Unformatted I/O not allowed* Le programme a essaye d'ecrire en non formatte dans un fichier declare **'form'=formatted**.

104 *Direct I/O not allowed* Le programme a essaye d'ecrire en acces direct dans un fichier declare **'access'=sequential**.

105 *Sequential I/O not allowed* Le programme a essaye d'ecrire de façon sequentielle dans un fichier declare **'access'=direct**.

107 *Cant backspace file* Un backspace a ete tente sur un fichier ou cette operation est interdite (eg un terminal).

108 *Off beginning of record* Essai de lire ou ecrire avant un bloc.

110 *Off end of record* Essai de lire ou ecrire apres un bloc.

112 *Incomprehensible list input* L'entree d'une serie d'arguments ne correspond pas a ce qu'attend le programme.

117 *Status='new' file exists* Le fichier declare 'new' existe.

118 *Status='old' file does not exist* Le fichier declare 'old' n'existe pas.

121 *Illegal argument* L'argument lors d'une entree/sortie est illegal.

8 Les signaux UNIX

Ces signaux peuvent etre envoyes au programme par l'utilisateur. Ils peuvent aussi etre envoyes par le programme au terminal pour indiquer un deroulement anormal.

Nom du signal	numero	action par defaut	signification
SIGINT	2	Terminer	Interrompre l'execution. Envoye au programme par **control C**
SIGQUIT	3	Core dump	Quitte
SIGILL	4	Core dump	Instruction illegale
SIGFPE	8	Core dump	Floating point exception
SIGKILL	9	Terminer	Kill. Le programme est arrete le plus rapidement possible. envoye par **kill -9**.
SIGBUS	10	Core dump	Bus error. Le programme a essaye d'acceder une zone memoire illegalement.
SIGSEV	11	Core dump	Segmentation violation. Le programme a essaye d'acceder une zone memoire illegalement.
SIGSYS	12	Core dump	Mauvais argument lors d'un appel systeme
SIGPIPE	13	Terminer	Le programme a essaye d'ecrire dans un "pipe" qui ne debouchait pas.
SIGTERM	15	Terminer	Un autre programme a stoppe le programme.
SIGSTOP	17	Stop	Stop (ne peut etre intercepte ou ignore) Le programme continuera lorsqu'il recevra le signal **CONTINUE**.
SIGCONT	19	si tourne, ne fait rien si stoppe, continue	Continue l'execution apres un SIGSTOP. Ce signal est envoye en tapant **fg** ou **bg** au clavier.
SIGTTIN	21	Stop	Le programme a essaye de lire au clavier en position **bg**. Ceci l'arrete et on peut le ramener en tache frontale avec **fg**. On peut alors lui entrer des donnees.