

Durée : 4 jours

Réf : MOD-FORTRAN

Modernisation de codes FORTRAN

Cette formation présente les éléments à modifier dans un code Fortran ancien (77 ou 95)

Elle passe en revue les anciennes fonctionnalités afin de les convertir dans un Fortran moderne (2008, 2018, 2023).

Objectifs

- Savoir remplacer les constructions du Fortran 77 ou 90
- Maîtriser les nouvelles fonctionnalités
- Savoir optimiser un code Fortran

Public

- Chercheurs, Ingénieurs
- Développeurs, Chefs de projet

Pré requis

- Avoir des connaissances de Fortran 77 ou 90/95

Évaluation des acquis

- TP et QCM pendant la formation

Méthodes et moyens pédagogiques

- Travaux pratiques
- Pédagogie active basée sur des exemples, des démonstrations, des partages d'expériences, des cas pratiques et une évaluation des acquis tout au long de la formation
- Un poste de travail par personne
- Support électronique

Points forts

- Groupe de 4 personnes maximum
- Alternance théorie/pratique
- Assistance après la formation

Modernisation de code FORTRAN

1 Environnement préalable

- définition de cas tests ou de validations
- utilisation d'un système de gestion de version (Git)
- utilisation d'un système de build (CMake, fpm, ...)
- outils d'analyse (Fortitude, ...)
- outils de tests unitaires
- support des normes suivant les compilateurs

2 Références de bonnes pratiques

- Fortran modernisation (<https://github.com/codee-com/fortran-modernization>)

3 Améliorations de la lisibilité

- utilisation du format libre
- forcer le typage explicite de toutes les variables (IMPLICIT NONE)
- typage au plus près (BLOCK)
- suppression des GOTO calculés, IF arithmétique et ASSIGN
- remplacement des boucles DO ... étiquette numérique par des DO END DO avec des étiquettes nommées
- remplacement des GOTO pour quitter une boucle par des CYCLE/EXIT
- suppression des GOTO par des structures : IF THEN, DO END ENDO, FUNCTION, SUBROUTINE
- suppression des ENTRY et des retours alternatifs des sous programmes

4 Précision des calculs

- définition des précisions (KIND)
- utilisation des suffixes de type dans les constantes

5 Améliorations de sécurité des fonctions et sous programmes

- déclaration explicite des variables (INTENT IN, OUT, INOUT)
- tableaux en paramètre (suppression des assume size et des tailles en paramètres, utilisation des assume shape, de SIZE, UBOUND, SHAPE)
- définition d'interface (utilisation de modules et de sous modules (IMPLICIT NONE EXTERNAL)
- suppression des COMMONS, BLOCK DATA, INCLUDE (utilisation des modules)

6 Amélioration des performances

- parcours d'un tableau (indice le plus à gauche dans la boucle la plus interne, section de tableaux)
- utilisation des fonctions intrinsèques sur les tableaux
- utilisation des opérateurs sur les tableaux
- utilisation de DO CONCURRENT (suppression des FOR ALL)
- utilisation des fonctions ELEMENTAL sur les tableaux

7 Améliorations de la structuration des données

- suppression des équivalences (pointeur, ASSOCIATE)
- tableaux allocatable

Contact

- 04 58 00 02 22
- contact@webformation.fr

Plus d'infos sur

<https://www.webformation.fr>

Modernisation de code FORTRAN

- pointeurs
- types personnalisés

8 Améliorations des entrées sorties

- Suppression des confusions sur les numéros (NEWUNIT)
- suppression de ERR et END avec étiquettes (IOSTAT, IOMSG)

9 Introduction aux nouveaux modèles de programmation

- programmation parallèle : coarrays :
images, coarrays, team, synchronisation
comparaison avec MPI/OpenMP
- programmation Orientée Objet :
extension des types personnalisés, liaisons de procédure,
polymorphisme, classes abstraites