

Perfectionnement en langage C

Durée: 5 jours

Prix et dates: nous consulter

Public:

Toute personne amenée à programmer, à superviser ou à modifier des logiciels écrits en langage C.

Objectifs:

Compléter des connaissances en langage C par une formation approfondie sur les mécanismes fondamentaux de fonctionnement.

Connaissances préalables nécessaires:

Il est demandé aux participants de connaître les structures et fonctions de base du langage C.

Programme:

- Rappels** : Les variables, type statique.
Variables statiques et variables registres.
Conversions.
- Structures** : Présentation, intérêt des structures.
syntaxe de définition, déclaration et d'accès aux éléments des structures.
Exemples: copie de structures
Structures avancées, unions, tableaux, champs binaires, drapeaux.
- Fonctions de saisie, affichage** : Options avancées de printf.
Mise en forme paramétrée.
Saisie avancée.
Rappel du principe : décomposition du flux d'entrée.
Les types 'ensemble'.
- Les opérateurs** : Rappels sur les opérateurs de calculs.
Opérateurs logiques, opérateurs binaires.
Travaux pratiques :
mise en oeuvre des opérateurs de décalage.
Priorité des opérateurs.

Perfectionnement en langage C

- Fonctions** : Pointeurs sur les fonctions.
Applications aux interpréteurs.
- Allocation dynamique** : Principe d'allocation mémoire.
Syntaxe de malloc et free.
Travaux pratiques de mise en oeuvre.
Fonctions avancées (calloc et realloc) :
intérêt et applications.
- Pratique** : Les listes chaînées, les arbres binaires.
Applications à l'organisation des données.
- Techniques de programmation** : Les phases de compilation :
précompilation , assemblage, édition de liens.
Définition de constantes.
Contrôle de compilation.
Les macro-instructions.
Conventions de nommage.
Comparaison avec les fonctions.
Les fichiers inclus : #include.
- Bibliothèques** : Méthode, syntaxe
Les bibliothèques standards : libc.a, libm.a, libcur.a
Fonctions disponibles dans la bibliothèque mathématique.
- Les entrées/sorties** : Mécanisme de stockage des fichiers.
Méthode d'accès, les descripteurs de fichiers.
Fonctions open/close.
Travaux pratiques :
écriture d'une fonctions permettant de tester l'existence d'un
fichier.
Fonctions read/write.
Mise en oeuvre avec lecture/écriture de structures.
Modes d'ouvertures spécifiques :
avec positionnement dans le fichier, avec création du
fichier, ...
Options : O_TRUNC, o_SYNC, O_NDELAY.
Le type FILE : mise en oeuvre de fprintf, fscanf, fgets, fputs.