AJC FORMATION

CYCLE

57 jours – 399 heures

DEV C++ LINUX EMBARQUE

Filière

AJC FORMATION - 6 rue Rougemont 75009 - 01 81 51 64 85 – formonsnous@ajc-formation.fr
S.A.R.L. au capital de 10 000 € - SIRET : 50096702100030
N° de déclaration d'activité 117542701 75 (ce numéro ne vaut pas agrément de l'Etat)

PROGRAMME DE LA FILIERE

AJC FORMATION - 6 rue Rougemont 75009 - 01 81 51 64 85 – formonsnous@ajc-formation.fr
S.A.R.L. au capital de 10 000 € - SIRET : 50096702100030
N° de déclaration d'activité 117542701 75 (ce numéro ne vaut pas agrément de l'Etat)

FILIERE SOFTWARE DEVELOPER C++

Programme

OBJECTIFS

- Exploiter et administrer l'environnement Linux
- Gérer les exigences et structurer les tests automatisés
- Acquérir les bases de la programmation structurée en langage C
- Apprendre à développer objet en C++
- Être capable en fin de session de concevoir une application sur environnement embarqué (ex : Raspberry Pi) et avec interfaces graphiques
- Acquérir le savoir être du consultant

METHODES ET MOYENS PEDAGOGIQUES

Méthodes pédagogiques. Pour l'ensemble des stagiaires, le cours intégrera les suivantes :

- Alternance d'exercices, cas pratiques, QCM et de notions théoriques, Projet Fil Rouge
- Evaluations

Moyens pédagogiques

- AJC met à la disposition de chaque stagiaire un accès à notre plateforme à distance ainsi qu'éventuellement les logiciels utiles dans le cadre de chaque module
- Les supports de cours seront remis via notre la plate-forme de téléchargement Quest et/ou AJC Classroom

Informations concernant les classes virtuelles

- Pour les formations en classe virtuelle, avec @JC CLASSROOM, vous profiterez des mêmes possibilités et interactions avec votre formateur que lors d'une formation présentielle : votre formation se déroulera en connexion continue 7h/7.
- Vous pourrez échanger directement avec le formateur et l'équipe pédagogique à travers notre système de visioconférence, mais aussi grâce aux forums et chats présents dans @JC CLASSROOM.
- Votre formateur sera à même de vérifier l'avancement de votre travail et de vous évaluer à l'aide d'exercices et de cas pratiques. Cela lui permettra de vous apporter un suivi pédagogique et des conseils personnalisés pendant toute la durée de la formation.
- Notre équipe technique vous enverra les modalités de connexion (accès, identifiants, dates, heures et numéro de la hotline) par mail dès votre inscription.
- Si vous rencontrez un problème de connexion, vous pourrez joindre à tout moment (avant ou même pendant la formation) notre hotline assistance technique au 01 82 83 72 41 ou par mail (hotline@ajc-formation.fr)

PRE-REQUIS

- Des notions d'algorithmie seraient un plus

PARTICIPANTS

- Scientifique ou toute personne en reconversion métier

POSTES VISES

- Développeur C, Développeur C++, Développeur Système embarqué, Ingénieur Développement C++, Développeur Logiciel Embarqué C-C++, Ingénieur Logiciel C++ …

LIEU

- Présentiel et/ou Distanciel

CERTIFICATION / ATTESTATION

- Attestation de formation

Programme - Contenu pédagogique

COMPORTEMENTAL	RÔLE ET COMPORTEMENT DU CONSULTANT OBJECTIF « QUALITÉ » DE LA MISSION	1 jour
FONDAMENTAUX ET NORMES	ALGORITHMIE	2 jours
	DO 178C	1 jour
	GESTION DES EXIGENCES	2 jours
	ROBOT FRAMEWORK	3 jours
UNIX - LINUX	PRESENTATION LINUX	1 jour
	COMMANDES LINUX	3 jours
	SHELL SCRIPT	3 jours
	ADMINISTRATION LINUX	1 jour
	ARCHITECTURE LINUX	1 jour
	SYSTÈME DE FICHIERS	1 jour
DEVELOPPEMENT D'APPLICATIONS	GIT	2 jours
	PROGRAMMATION C	5 jours
	C VERS C++ 11	3 jours
	PROGRAMMATION PYTHON	3 jours
	SQLITE	1 jour
	DEBUG	1 jour
EMBARQUE	LINUX EMBARQUE	5 jours
	RASPBERRY PI	3 jours
FONDAMENTAUX ET NORMES	AGILE SCRUM	1 jour
COMPORTEMENTAL	PRESENTER SES NOUVELLES COMPETENCES	1 jour
PROJET	PROJET FINAL & SOUTENANCE - APPLICATION EMBARQUEE	13 jours

57 JOURS

PROGRAMMES DÉTAILLÉS

COMPORTEMENTAL

AJC FORMATION

DEV C++ LINUX EMBARQUE

ROLE ET COMPORTEMENT DU CONSULTANT

1 jour,

7 heures

PRESENTIEL
et/ou
DISTANCIEL

PROGRAMME DU MODULE

Pourquoi s'intéresser aux Module communication

- Force et faiblesse de son expression orale
- Réactivité et spontanéité dans sa prise de parole
- Apprivoiser son stress pour développer une image cohérente de soi
- Prise de conscience de l'image que l'on véhicule
- Identifier et traiter les agents stressants lors de l'entretien client
- Cerner les croyances limitatives en rapport avec le contexte de la mission

Estime de soi

- Influence sur soi même et sur les autres lors de l'entretien et au cours de la mission
- Identifier et mettre en valeur ses atouts en rapport avec la mission
- Parler de soi en gardant une écoute assertive

Objectif qualités de la mission

- Identifier clairement les attentes et les objectifs du client
- Anticiper les difficultés (objections, déstabilisations, critiques)
- Définir les objectifs qualités en adoptant son rôle et son comportement au contexte de la mission
- Positionnement du consultant vis à vis de client et des collaborateurs au sein de la mission (ex : communication en réunion…)
- Nature et gestion des conflits

OBJECTIFS

- La communication interne et externe au sein de l'entreprise
- Adapter et maîtriser les différents types de communication pour accroître son efficacité personnelle

PROGRAMME DETAILLE

FONDAMENTAUX ET NORMES

DEV C++ LINUX EMBARQUE

ALGORITHMIE

2 jours,
14 heures

PRESENTIEL
et/ou
DISTANCIEL

PROGRAMME DU MODULE

Maîtriser les outils de l'algorithmique

(Schémas de programme, types et structures de données, modules)

Acquérir le s bases des méthodes de programmation structurée nécessaires à l'apprentissage de tout langage de programmation

Apprendre à raisonner sur un algorithme

Maitriser les notions de fichiers

Découvrir et mettre en œuvre la traduction d'un algorithme dans un langage de programmation

OBJECTIFS

- Présenter les principes fondamentaux de la programmation et de l'algorithmique et expliquer les notions communes à tous les langages de programmation
- S'approprier les structures logiques et la démarche de résolution d'un problème de façon structurée et indépendante de toute contrainte matérielle ou logicielle
- Résoudre des problèmes plus ou moins complexes

PROGRAMME DETAILLE

DEV C++ LINUX EMBARQUE

DO 178C

1 jour,
7 heures

PRESENTIEL
et/ou
DISTANCIEL

PROGRAMME DU MODULE

Le contexte réglementaire de la certification

- Les principes généraux et les aspects systèmes
- Certification système vs certification logiciel
- Les niveaux de logiciel
- Les options architecturales

La planification du développement

- La définition du cycle de vie
- Les objectifs et les activités de planification
- Les critères de transition
- Les plans et les standards
- L'environnement (méthodes, outils,...)

Le processus de développement

Les activités de soutien

- La vérification
- La gestion de configuration
- L'assurance qualité

La démonstration de conformité

- Traçabilité des informations
- Couverture des vérifications
- Coordination pour la certification
- Règles de codages
- Audits de code
- Sampling/inspections

Les aspects particuliers

- Les logiciels du commerce
- Les logiciels réutilisés
- Les outils et l'environnement de développement
- Les moyens de conformité spécifiques

OBJECTIFS

- Connaître les exigences applicables aux logiciels aéronautiques critiques
- Connaître la démarche à appliquer pour une certification de logiciel embarqué
- Savoir comment adapter cette démarche dans les cas particuliers prévus par la norme (codage automatique, réutilisation de logiciel...)

PROGRAMME DETAILLE

DEV C++ LINUX EMBARQUE

GESTION DES EXIGENCES

2 jours,
14 heures

PRESENTIEL
et/ou
DISTANCIEL

PROGRAMME DU MODULE

Principes de la gestion des exigences

Procédures et processus

Gestion de projet et des risques

Responsabilités et règles

Définition des exigences

Spécification des exigences

Analyse des exigences

Traçabilité des exigences

Gestion des exigences

Gestion des anomalies

Utilisation des outils

OBJECTIFS

- Appliquer des méthodes structurées et systématiques d'ingénierie des exigences.
- Maitriser l'accroissement de la pertinence des exigences, leur réalisation et leur gestion
- Appliquer des règles pour la rédaction d'exigences dans un langage naturel ,de même que les règles portant sur l'amélioration et les exigences qualité des spécifications

DEV C++ LINUX EMBARQUE

ROBOT FRAMEWORK

3 jours,
21 heures

PRESENTIEL
et/ou
DISTANCIEL

PROGRAMME DU MODULE

Introduction à l'automatisation

- Le projet d'automatisation
- Périmètre d'automatisation
- Les bonnes pratiques
- L'automatisation basée sur les mots-clefs : intérêt

Présentation de Robot Framework

- Architecture et concept
- Les librairies embarquées
- Panorama des librairies externes
- Éditeur Ride

Écrire des tests avec les mots-clefs

- Structure d'un test, suite de tests et résultats de tests
- Les variables
- SetUp, TearDown et Tag
- Data Driven Test avec Robot Framework
- Behavior Driven Test avec Robot Framework

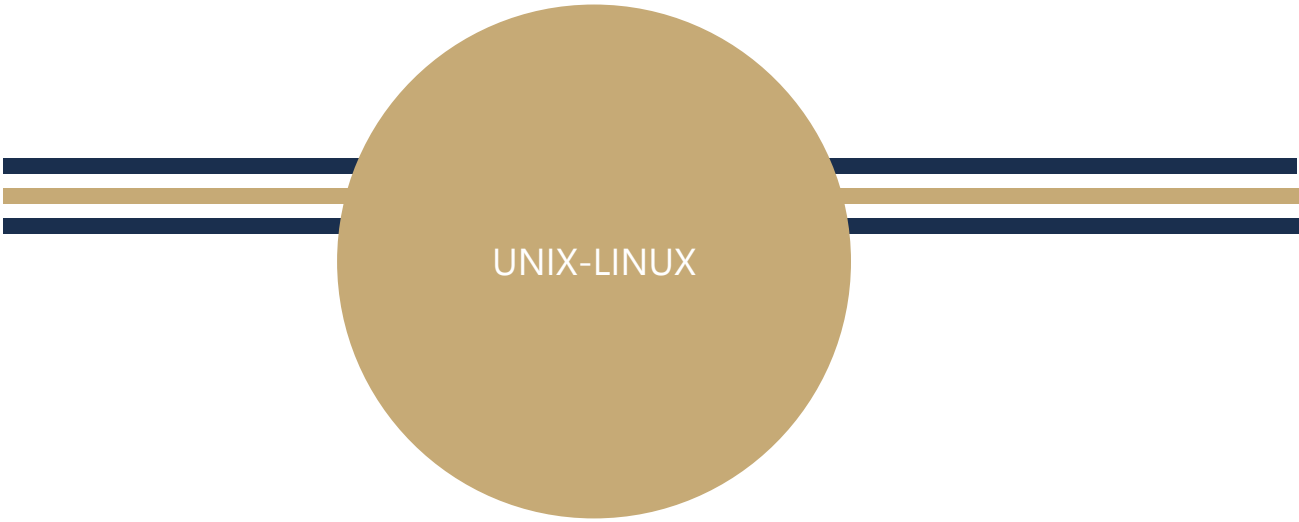
Les librairies standard de

- Panorama des librairies standard
- Built-in (gestion des tests)
- ScreenShot
- Dialogs

OBJECTIFS

- Maîtriser les bases du développement de test automatique avec Robot Framework
- Écrire, structurer et analyser un test par mot-clé
- Créer une librairie
- Comprendre l'intérêt de l'intégration continue et l'utilisation de Robot Framework avec Jenkins

PROGRAMME DETAILLE


UNIX-LINUX

DEV C++ LINUX EMBARQUE

PRESENTATION LINUX

1 jour,
7 heures

PRESENTIEL
et/ou
DISTANCIEL

PROGRAMME DU MODULE

L'historique

- Les fonctionnalités d'UNIX
- L'organisation du système d'exploitation

Connexion

- La connexion en mode texte (telnet, ssh)
- La connexion en mode graphique (XDMCP) La déconnexion

Environnement

- Les notions de UID et GID L'arborescence type sous UNIX Les principaux répertoires
- Le répertoire de connexion
- Le prompt

Arborescence et répertoires L'arborescence type sous UNIX Les principaux répertoires

- La structure de l'arborescence UNIX
- Le déplacement dans l'arborescence (pwd, cd) La manipulation de répertoires (mkdir, rmdir)

Fichiers

- Les différents types de fichiers
- La substitution de caractères
- La manipulation de fichiers (cat, more, pg, cp, mv, rm)

Droits

- Les principes
- Les droits par défaut
- La manipulation de droits (chmod)

Bases de l'éditeur de texte vi

Les différents modes de vi L'utilisation de vi

- Le mode vi sur la ligne de commandes

Personnalisation de l'environnement /etc /profile et $HOME/.profile

- Les alias
- L'historique des commandes
- La modification du mot de passe

OBJECTIFS

- Comprendre les logiciels libres et l'Open Source
- Connaître les principes fondamentaux du système d'exploitation
- Utiliser interactivement le Shell et connaître les commandes essentielles
- Gérer les fichiers et les dossiers
- Utiliser nano, vi et vim

PROGRAMME DETAILLE

DEV C++ LINUX EMBARQUE

COMMANDES LINUX

3 jours,
21 heures

PRESENTIEL
et/ou
DISTANCIEL

PROGRAMME DU MODULE

Introduction

- Présentation des shells
- Premières commandes
- Les différents Shell
- Comparaison de sh, bash, ksh et dash

Aide

- L'aide locale
- À savoir
- Freenode
- Usage IRC

Commandes de base

- Accéder au contenu des fichiers
- Commandes de compression, d'impression et de gestion du temps
- Gestion administration
- Commandes composites pipes et redirections

Variables

- Présentation
- Variables utilisateur
- Prompt
- Substitutions

Commandes internes

- Set
- cd, pushd, popd, umask, type, enable
- Historique et Alias
- Commande sur les processus : kill, jobs, wait, ulimit

Utilitaires

- Commandes cut, tr, uniq, sort, wc, find, grep
- Commandes de transformation : iconv, od, nl, basename, diff
- Commande utilitaires : xargs, tee, cmp, comm, paste, sed
- Expressions régulières

OBJECTIFS

- Les commandes de base, l'utilisation poussée du bash, les grand utilitaires Unix dont les expressions régulières, sed et awk

PROGRAMME DETAILLE

DEV C++ LINUX EMBARQUE

SHELL SCRIPT

3 jours,
21 heures

PRESENTIEL
et/ou
DISTANCIEL

PROGRAMME DU MODULE

Introduction

- Rôle d'un Shell
- Présentation des différents shell sous Unix/Linux
- Types et syntaxes

Aide

- Les man
- Help
- IRC freenode

Paramétrage de l'environnement

- Options du Shell
- Variables et fichiers d'environnement
- Historique des commandes

Utilisation du Shell en mode interactif

- Énumérer les commandes essentiels par thème
- Substitution de nom de fichiers
- Protection des caractères spéciaux
- Redirections et Tubes de communication
- Regroupement des commandes

Base de la programmation

- Structure d'un script
- Commentaires
- Exécution d'un script
- Débogage d'un script et Code de retour

Variables et constantes

- Variables et Constantes
- Tableaux
- E/S de données
- Commandes de substitution
- Pushd et popd

Structure de contrôle

- Instructions conditionnelles
- Choix multiples
- Boucles et Sauts inconditionnels

Alias et fonctions

- Alias
- Sous-programme sous forme de script
- Sous-programme sous forme de fonction

Arithmétiques

- Syntaxe
- Commande expr

Expressions régulières

- Meta-caractères des expressions régulières
- Utilisation des expressions régulières avecGrep

Chaine de caractères

- Manipulation de chaînes de caractères
- Expressions de variables
- Commandes basename et dirname

Filtre Sed

- Principe de fonctionnement
- Commandes de sed
- Utilisation des expressions régulières
- Présentation des sous-expressions

Processeur de texte AWK

- Principes de fonctionnement
- Structure d'un programme awk
- Critères
- Variables et les expressions
- Tableaux
- Instructions et Fonctions prédéfinies

OBJECTIFS

- Acquérir les compétences pour écrire des scripts en Shell et exploiter les possibilités des filtres Unix/Linux

PROGRAMME DETAILLE

DEV C++ LINUX EMBARQUE

ADMINISTRATION LINUX

1 jour,
7 heures

PRESENTIEL
et/ou
DISTANCIEL

PROGRAMME DU MODULE

Installation

- Philosophie Debian
- Préparation de l'installation
- L'installeur Debian
- Enchaînement des écrans d'installation

Utilisateurs

- Commandes de base
- Fichiers utilisateurs
- Gestion des groupes
- Réglage du shell

Mise à jour

- Mise à jour via les dépôts
- Mise à jour via des paquets
- Installation d'un paquet
- Mise à jour via les sources

File system

- Organisation des fichiers
- Système de fichiers
- Commandes
- RAIDS logiciels
- LVM

Démarrage

- Description du processus de démarrage
- Init et upstart
- Processus
- Gestion des processus
- Gestion des signaux
- Tâches avant et arrière plans
- Planification de tâches

Sauvegarde

- Présentation des principales commandes d'archivage
- Création d'une stratégie de sauvegarde
- Automatisation des tâches de sauvegarde
- Outils de sauvegarde

Journaux

- Lectures des journaux textes et binaires
- Rotation des journaux
- Effectuer une synthèse avec Logwatch

Réseau

- Gestion des pilotes d'interfaces
- Démarrage et arrêt du firewall (iptables),
- TCPIP
- Outils de diagnostic

À distance

- Configuration du serveur et du client OpenSSH

OBJECTIFS

- Les commandes de base de l'administration système en ligne de commande et via les programmes graphiques

PROGRAMME DETAILLE

DEV C++ LINUX EMBARQUE

ARCHITECTURE LINUX

1 jour,
7 heures

PRESENTIEL
et/ou
DISTANCIEL

PROGRAMME DU MODULE

Architecture

- Architecture Linux
- Les différents éléments d'un système Linux
- Le BIOS et le boot
- Le boot
- Introduction au Noyau
- La librairie LibC

Le noyau

- Présentation
- Versions
- Les sources
- Configuration du noyau
- Module / Kernel / None
- Avantage / Désavantage des modules
- Commandes sur les modules

Init et démons

- Inittab et init
- Niveau d'exécution
- Exemple fichier `/etc/inittab`
- Les différents types de démon

Init et shell

- Les commandes de base et programme
- Le Shell
- Les gestionnaires de fenêtre
- Qt

OBJECTIFS

- L'architecture Linux, le noyau, init et demons, init et shell

PROGRAMME DETAILLE

AJC FORMATION

DEV C++ LINUX EMBARQUE

SYSTÈME DE FICHIERS

1 jour,
7 heures

PRESENTIEL
et/ou
DISTANCIEL

PROGRAMME DU MODULE

Présentation des disques

- Histoire du disque
- Faits et chiffres
- Géométrie des disques
- IDE, ATA, PATA, SATA, SCSI,USB, DMA,ULTRA DMA
- Alimentation
- MOLEX SATA

Informations et customisation des disques

- Connaître les disques présents sur le système et non montés
- Connaître le hardware présent sur le système
- Les pseudos devices et leurs informations
- Les commandes de customisation
- Augmenter les taux de transfert disque
- Démon de vérification des erreurs disques SMART

Partitionnement des disques

- Partitionnement MBR
- Partitionnement GPT
- Prise en charge du firmware UEFI
- Partitionner en MBR
- Partitions primaires, étendues, secondaires
- Partitionner en GPT
- Conversion partitions MBR/GPT

Montage de systèmes de fichiers sans LVM

- Arborescence FHS
- Revue de différents systèmes de fichiers : description, caractéristiques, limitations, montage, option du kernel
- Les FS actuels : ext2, ext3, ext4, reiserFS, FAT, FAT32, NFTS, ISO9660, JFS, QNXFS, UDF, JFS, BTRFS, NILFS2
- Les FS anciens et récupération de données : ADFS, AFFS, EFS, HFSN HFSPLUS, HPFS, MINIX, SysV, ufs, umsdos
- Les FS réseaux : NFS, CIFS, NCPFS, AFS, Coda, GFS, CEPH GS
- Les FS mémoire : SQUASHFS, RAMFS, ROMFS, UBIFS, JFFS2, YAFFS2, LogFS
- Les pseudo File System : proc, sys, devpts, debugfs, autofs, fuse

La commande mount

- Syntaxe mount
- Les types de paramètres
- Option -t
- Option -o
- Option bind et move
- Device /dev/loop0
- Montage loop
- Démontage d'un périphérique
- UUID
- Le fichier fstab

LVM

Montages à chaud

Le cas des clés USB

Le RAID

OBJECTIFS

- Quels sont les disques présents sur votre système, même ceux qui ne sont pas montés ?
- Comment gérer le RAID ou le LVM sous Linux ?
- Comment installer les récentes partition GBT compatibles avec l'UEFI ?
- Comment fonctionnent les UUID ?
- Augmenter la vitesse de transfert d'un disque

PROGRAMME DETAILLE

DEVELOPPEMENT D'APPLICATIONS

AJC FORMATION

DEV C++ LINUX EMBARQUE

GIT

2 jours,
14 heures

PRESENTIEL
et/ou
DISTANCIEL

PROGRAMME DU MODULE

Les gestionnaires de version

- Histoire de GIT
- Gestionnaires de versions centralisés vs distribués
- Les différents gestionnaires de version
- Gestion shapshot vs gestion différentielle
- Les avantages de GIT
- **Git mise en oeuvre**
- Installation sous Linux
- Premières configurations
- Installation à partir des sources
- Installation sur MacOSX
- Installation sous Windows
- Commandes d'aide

Créer un dépôt Git

- Initialisation, clonage
- Status
- Les différents états d'un fichier
- Commandes add et commit
- Commandes rm et mv
- Visualisation de l'historique
- Outil graphique gitk
- Annuler des actions

Travailler avec un dépôt distant

- git remote
- git fetch
- git pull
- git push

Etiquettes et branches

- Étiquettes légères, annotées, signées
- Propagation des étiquettes
- Complétion et Alias
- Les branches
- Ajout d'une branche
- Checkout d'une branche

Fusion de branches

- Avance rapide
- Fusion triangulaire
- Résolution de conflit dans la branche
- Gestion des branches
- Travailler avec les branches
- Branches distantes

Rebaser les branches

- Principes et dangers

Git sur le serveur

- Les protocoles
- local, https, git
- Installation de git sur un serveur
- Dépôt nus
- Copie de dépôts
- Génération de clés publiques ssh
- Mise en œuvre du serveur
- Accès public

Extensions serveur

- GitWeb, Gitosis, Gitolite, Démon Git, Git hébergé

Git Distribué

- Développements distribués
- Contribution à un projet
- Maintenance d'un projet

Utilitaires Git

- Sélection des versions
- Indexation interactive
- Le remisage
- Réécrire l'historique
- Deboguer avec Git
- Sous-modules
- Fusion de sous-arborescences
- Migrer de SVN à Git

OBJECTIFS

- Comment utiliser et tirer partie du gestionnaire de versions distribué Git

PROGRAMME DETAILLE

AJC FORMATION

DEV C++ LINUX EMBARQUE

PROGRAMMATION C

5 jours,
35 heures

PRESENTIEL
et/ou
DISTANCIEL

PROGRAMME DU MODULE

Présentation du langage

- Historique, utilisation, organisation des fichiers, éditeur, compilation, environnement de développement, domaines d'utilisation, la norme ANSI

Le Langage

- Caractères autorisés, la fonction main, blocs et instructions, commentaires, initiation préprocesseur, types de données élémentaires

Les variables

- Déclaration, déclarations globales et locales, initialisation des variables
- Sorties formatées d'un programme
- Entrées/sorties formatées.
- Formatage numérique, formatage caractère, entrées formatées

Les opérateurs

- Opérateurs arithmétiques.
- Mécanismes d'évaluation des expressions.
- Post et pré-incrémentation de décrémentation.
- Précédence et associativité des opérateurs.

Expressions logiques

- Instruction d'affectation
- Mécanismes de fonctionnement des expressions logiques.

Opérateurs de comparaison

- if, switch, while, do, for, break, continue

Opérateurs de type

- Cast, sizeof, malloc, delete

Opérateurs travaillant au niveau du bit

- ET, OU, OU exclusif, complément à 1.Décalages.

Tableaux, pointeurs et chaînes de caractères

- Définition. Tableau à une dimension. Initialisation. Tableau multi-dimensionnel. Chaînes de caractères - Copie de chaînes de caractères. Opération sur les chaînes de caractères.

Les fonctions

- Présentation, définition, déclaration, paramètres de fonction, retour d'une fonction, appel, passage de l'adresse d'une fonction

Compléments sur les directives de compilation

OBJECTIFS

- Acquérir les bases du langage C, le langage utilisé pour sa rapidité
- Aborder les éléments du langage et les spécificités du compilateur gnc c

PROGRAMME DETAILLE

AJC FORMATION

DEV C++ LINUX EMBARQUE

C VERS C++

3 jours,
21 heures

PRESENTIEL
et/ou
DISTANCIEL

PROGRAMME DU MODULE

Programmation Orientée Objet

- Modélisation
- Abstraction
- Encapsulation
- Classe et Objets
- Instanciation d'objet
- Les membres, attributs, méthodes
- Héritage
- Polymorphisme

Du C au C++

- Préambule
- Définitions
- Types
- Paramètres de fonctions par référence
- Fonctions inline
- Appels de fonctions C
- Entrées et sorties simples

Les classes

- Déclaration dans le fichier cpp
- Méthodes
- Paramètres par défaut
- Masquage
- Surcharge

Les constructeurs

- Exemples d'utilisation
- Types de constructeurs
- Constructeur par copie
- Constructeurs par transtypage
- Usage du mot clé explicit
- Constructeurs à arguments multiples
- Liste d'initialisations
- Le destructeur

Qualifieurs

- Constance
- Variables statiques
- Méthodes statiques
- Autres types de classe de stockage
- Constance
- Fonctions amis
- Classes amies

Surcharge des opérateurs

- Liste des opérateurs qui peuvent être surchargés
- Surcharge d'un opérateur par une fonction membre
- Surcharge d'un opérateur par une fonction non membre
- Surcharge de l'opérateur d'affectation `operator=`
- Opérateur de conversion vers un autre type
- Symétrie

Les exceptions

- Exemple d'exception
- Remarques
- Attraper une exception
- La classe exception et sa dérivée stdexcept
- Réponses
- Laisser échapper une exception
- Déclaration de fonction avec exception non traitées
- Faire/défaire

OBJECTIFS

- Appréhender les concepts objets (Classe, Encapsulation, polymorphisme, héritage) qui ne sont pas présents en langage C
- Tester des Bibliothèques et API

PROGRAMME DETAILLE

AJC FORMATION

DEV C++ LINUX EMBARQUE

C VERS C++ (Suite 1)

3 jours,
21 heures

PRESENTIEL
et/ou
DISTANCIEL

PROGRAMME DU MODULE

Espaces de nommage

- Présentation
- Déclaration
- Définition
- Utilisation
- Exemples connus
- Exclusion de méthodes d'un namespace
- Espace de nommage anonyme

Préprocesseur

- Les directives
- Directives include et define simple
- Instructions conditionnelles
- ifndef define ifdef
- identificateur
- Les macros
- Macros prédéfinies

Dérivation - Héritage

- Classe fille
- Déclaration de la classe fille
- Encapsulation
- Particularité amusante
- Héritage multiple
- Portée d'héritage
- Redéfinition vs surcharge
- Problématiques des classes dérivées

Polymorphisme

- Méthodes virtuelles
- Contrainte sur les fonctions virtuelles
- Fonctions virtuelles pures et classes abstraites

Pointeurs

- Rappels
- Définition d'un pointeur
- Difficultés de notation
- Pointeurs et allocation mémoire
- Désallocation
- Organisation de la mémoire
- Pointeurs et classes
- Le pointeur *this*
- Arithmétique sur les pointeurs
- Cas d'utilisation des pointeurs : pas

RTTI

- Définition
- Exemple pratique
- Règles à respecter
- TypeId
- Utilisation

Templates

- Présentation
- Patrons de fonction
- Utilisation
- Remarques
- Piège sur les pointeurs
- Patron de fonction à l'intérieur d'une classe
- Instanciation implicite et explicite
- Spécialisation

Patrons de classe

- Exemple
- Fonctions exportées
- Template sur les opérateurs
- Foncteurs
- A quoi servent les foncteurs ?
- Cas d'utilisation
- Performance

OBJECTIFS

- Appréhender les concepts objets (Classe, Encapsulation, polymorphisme, héritage) qui ne sont pas présents en langage C
- Tester des Bibliothèques et API

PROGRAMME DETAILLE

DEV C++ LINUX EMBARQUE

C VERS C++ (Suite 2)

3 jours,
21 heures

PRESENTIEL
et/ou
DISTANCIEL

PROGRAMME DU MODULE

Pointeurs intelligents

- Problématique des pointeurs
- Présentation des pointeurs intelligents
- unique_ptr
- Exemple de unique_pointer
- shared_ptr
- Exemple de shared pointer
- weak_ptr
- Avantages et inconvénients des pointeurs intelligentsssage d'argument

C11

- Présentation
- Templates variadiques
- Templates externes
- Assertions statiques
- Expressions et fonctions lambda
- Sémantique RValue Reference/Move
- Enumérations typées
- Le type tuple
- Support des expressions régulières
- Tables de hachage
- Boucles d'intervalle

OBJECTIFS

- Appréhender les concepts objets (Classe, Encapsulation, polymorphisme, héritage) qui ne sont pas présents en langage C
- Tester des Bibliothèques et API

PROGRAMME DETAILLE

DEV C++ LINUX EMBARQUE

PROGRAMMATION PYTHON

3 jours,
21 heures

PRESENTIEL
et/ou
DISTANCIEL

PROGRAMME DU MODULE

Syntaxe du langage Python

- Les Identifiants et les références. Les Conventions de codage et les règles de nommage.
- Les blocs, les commentaires.
- Les types de données disponibles.
- Les variables, l'affichage formaté, la portée locale et globale.
- La manipulation des types numériques, la manipulation de chaînes de caractères.
- La manipulation des tableaux dynamiques (liste), des tableaux statiques (tuple) et des dictionnaires.
- L'utilisation des fichiers.
- La structure conditionnelle if / elif / else.
- Les opérateurs logiques et les opérateurs de comparaison.
- Les boucles d'itérations while et for. Interruption d'itérations break / continue.
- La fonction range.
- L'écriture et la documentation de fonctions.
- Les Lambda expression.
- Les générateurs.
- La structuration du code en modules.
- Travaux pratiques

Approche Orientée Objet

- Les principes du paradigme Objet.
- La définition d'un objet (état, comportement, identité).
- La notion de classe, d'attributs et de méthodes.
- L'encapsulation des données.
- La communication entre les objets.
- L'héritage, transmission des caractéristiques d'une classe.
- La notion de polymorphisme.
- Association entre classes.
- Les interfaces.
- Présentation d'UML.
- Les diagrammes de classes, de séquences, d'activités, ...
- Notion de modèle de conception (design pattern).
- Travaux pratiques Modélisation en UML d'un cas d'étude simple.

Programmation Objet en Python

- Les particularités du modèle objet de Python.
- L'écriture de classes et leur instanciation.
- Les constructeurs et les destructeurs.
- La protection d'accès des attributs et des méthodes.
- La nécessité du paramètre self.
- L'héritage simple, l'héritage multiple, le polymorphisme.
- Les notions de visibilités.
- Les méthodes spéciales.
- L'introspection.
- L'implémentation des interfaces.
- Les bonnes pratiques et les modèles de conception courants.
- L'utilisation du mécanisme d'exception pour la gestion des erreurs.
- Travaux pratiques

OBJECTIFS

- Initier les participants aux méthodes et réflexes de la programmation orientée objet et leur apporter la maîtrise opérationnelle du langage Python.

PROGRAMME DETAILLE

DEV C++ LINUX EMBARQUE

PROGRAMMATION PYTHON (Suite)

3 jours,
21 heures

PRESENTIEL
et/ou
DISTANCIEL

PROGRAMME DU MODULE

Utilisation StdLib

- Les arguments passés sur la ligne de commande.
- L'utilisation du moteur d'expressions régulières Python avec le module "re", les caractères spéciaux, les cardinalités.
- La manipulation du système de fichiers.
- Présentation de quelques modules importants de la bibliothèque standard : module "sys", "os", "os.path".
- Empaquetage et installation d'une bibliothèque Python.
- Les accès aux bases de données relationnelles, le fonctionnement de la DB API.
- Travaux pratiques

Outils QA

- Les outils d'analyse statique de code (pylint, pychecker).
- L'analyse des comptes rendus d'analyse (types de messages, avertissements, erreurs).
- Extraction automatique de documentation.
- Le débogueur de Python (exécution pas à pas et analyse post-mortem).
- Le développement piloté par les tests.
- Les modules de tests unitaires Python (Unittest, ...).
- L'automatisation des tests, l'agrégation de tests.
- Les tests de couverture de code, profiling.
- Travaux pratiques

Création IHM TkInter

- Les principes de programmation des interfaces graphiques.
- Présentation de la bibliothèque TkInter.
- Les principaux conteneurs.
- Présentation des widgets disponibles (Button, Radiobutton, Entry, Label, Listbox, Canvas, Menu, Scrollbar, Text, ...).
- Le gestionnaire de fenêtres.
- Le placement des composants, les différents layouts.
- La gestion des événements, l'objet event.
- Les applications multi-fenêtres.
- Travaux pratiques

Interfaçage Python/C

- Présentation du module ctypes.
- Le chargement d'une librairie C.
- Appel d'une fonction.
- La réécriture d'une fonction Python en C avec l'API Python/C.
- La création de modules C pour Python avec Pyrex.
- L'Interpréteur Python dans C.
- L'utilisation du profileur de code.
- Travaux pratiques.

Conclusion

- Analyse critique de Python.
- L'évolution du langage.
- Eléments de webographie et de bibliographie.
-

OBJECTIFS

- Initier les participants aux méthodes et réflexes de la programmation orientée objet et leur apporter la maîtrise opérationnelle du langage Python.

PROGRAMME DETAILLE

AJC FORMATION

DEV C++ LINUX EMBARQUE

SQLITE

1 jour,
7 heures

PRESENTIEL
et/ou
DISTANCIEL

PROGRAMME DU MODULE

Introduction

- Généralités
- Concepts et langage
- Avantages et différences avec MySQL et PostgreSQL

Installation/Configuration

- Pré-requis
- Installation
- Configuration à l'exécution
- Types de ressources

Les Outils

- Base de données
- Les objets
- Les tableaux
- Les lignes et colonnes
- Les différentes vues
- La bibliothèque

Débuter avec la bibliothèque

- Ouvrir une base
- Créer un objet : sqlite_factory
- Ouvre une connexion SQLite persistante et crée la base si elle n'existe pas : sqlite_popen
- Exécuter une requête : sqlite_array_query
- Exécuter une requête sans résultats sur une base de données : sqlite_exec
- Retourner un tableau
- Configurer un délai d'attente : sqlite_busy_timeout
- Retourner le nombre de lignes modifiées : sqlite_changes
- Lire la valeur d'une colonne : sqlite_column
- Lire une ligne de résultat dans un tableau : sqlite_current
- Place le pointeur de résultat au début : sqlite_rewind
- Déplace le pointeur de résultat vers une ligne : sqlite_seek
- Fermer la connexion à SQLite : sqlite_close

Enregistrements

- Enregistrer une UDF agrégeante pour les requêtes : sqlite_create_aggregate
- Enregistrer une fonction utilisateur classique : sqlite_create_function

Les tableaux

- Retourner toutes les lignes d'un jeu de résultats en tant que tableau de tableaux : sqlite_fetch_all
- Lire la prochaine ligne de résultats dans un tableau : sqlite_fetch_array
- Retourner dans un tableau des types de colonnes d'une table : sqlite_fetch_column_types

OBJECTIFS

- Connaître les fonctions essentielles de SQLite
- Installer et configurer la bibliothèque, utiliser l'ensemble de ses outils ou encore exécuter des requêtes

PROGRAMME DETAILLE

DEV C++ LINUX EMBARQUE

SQLITE (Suite)

1 jour,
7 heures

PRESENTIEL
et/ou
DISTANCIEL

PROGRAMME DU MODULE

Les lignes, colonnes et champs

- Retourner à la ligne suivante en tant qu'objet : sqlite_fetch_object
- Indique s'il reste des lignes à lire : sqlite_has_more
- Si oui ou non une ligne précédente est disponible : sqlite_has_prev
- Retourner à l'index de la ligne courante : sqlite_key
- Retourne le numéro de ligne de la dernière ligne insérée : sqlite_last_insert_rowid
- Déplace le pointeur SQLite vers la prochaine ligne : sqlite_next
- Exécute une requête et retourne soit un tableau pour une colonne unique, soit la valeur de la première ligne : sqlite_single_query
- Retourne le nom du champ : sqlite_field_name
- Retourne le nombre de champs dans un résultat : sqlite_num_fields

Codage

- Retourne le message d'erreur : sqlite_error_string
- Retourne le dernier code d'erreur : sqlite_last_error
- Retourne l'encodage utilisé par la bibliothèque : sqlite_libencoding
- Décode des données binaires, passées à une UDF SQLite : sqlite_udf_decode_binary
- Encode les données binaires d'une UDF SQLite avant de les retourner : sqlite_udf_encode_binary

Les contraintes

- Primary key
- Rowid et Autoincrement
- Foreign Key
- Pragmas

OBJECTIFS

- Connaître les fonctions essentielles de SQLite
- Installer et configurer la bibliothèque, utiliser l'ensemble de ses outils ou encore exécuter des requêtes

PROGRAMME DETAILLE

DEV C++ LINUX EMBARQUE

DEBUG

1 jour,
7 heures

PRESENTIEL
et/ou
DISTANCIEL

PROGRAMME DU MODULE

Problèmes rencontrés

- Mauvais adressage mémoire
- Problèmes de cadrage des informations
- Perte de mémoire

Profilage sans source

- strace et ltrace
- La mémoire
- lsof, netstat
- Sonde Nagios

Debug graphiques

- ddd
- Eclipse cdt

Cas avancés

- Debuguer un programme en exécution
- Debuguer plusieurs programmes simultanément
- Debuguer via un core dump
- Debuguer à distance

Outils code

- Valgrind
- Exemples
- Profile de code
- Electric Fence

OBJECTIFS

- Apprendre à repérer les pertes de mémoire, les problèmes de débordement mémoire et savoir se servir des outils de débugage
- Apprendre à utiliser un core dump, le débugage croisé, le debug d'un programme

PROGRAMME DETAILLE

EMBARQUE

AJC FORMATION

DEV C++ LINUX EMBARQUE

LINUX EMBARQUE

4 jours,
28 heures

PRESENTIEL
et/ou
DISTANCIEL

PROGRAMME DU MODULE

Uboot

- Présentation et utilisation de Uboot
- Les chargeurs de démarrage (Bootloaders)
- Licences
- Les distributions
  - Les distributions
  - Les outils de mise à jour
- Les licences
  - Objectif
  - Free vs Open Source
  - Avertissement
  - L'objectif
  - Les formes de licence
  - Les principales licences de logiciels libres

Applications et bibliothèques légères pour systèmes embarqués

Systèmes de fichiers pour stockage de type bloc

Le noyau

- Présentation
  - Présentation
  - Versions
  - Les sources
  - Configuration du noyau
  - Module / Kernel / None
  - Avantage / Désavantage des modules
  - Commandes sur les modules
- Configuration
  - make xconfig
  - make gconfig
  - make menuconfig
  - make nconfig
  - Autres possibilités
  - Les options de compilation du noyau
  - Options de compilation (fin)
- Options de compilation
  - La compilation
  - Exercice : compiler un noyau
  - Installation du noyau
  - Installation des modules
  - Disque minimal en RAM
  - Fichier config

Busybox

- Busybox : présentation
  - Busybox : configuration
  - Les commandes Busybox
  - Ajouter des commandes à Busybox
  - Finalisation d'ajout d'une commande

Raspberry

- Architecture ARM
  - Les processeurs ARM
  - Quelques termes à connaître
  - Les familles de processeurs ARM
  - Les processeurs ARM9 et ARM10
  - Les Cortex
- Présentation de la carte
  - Les cartes Raspberry
  - Carte Raspberry PI 2
  - Carte raspberry PI 2
  - Légendes
  - OS disponibles
- Installation de base
  - Principe d'installation
  - Formatage via fdisk
  - Copie de Raspbian sur la carte

OBJECTIFS

- Découvrir les outils de développement sous Linux
- Maîtriser les mécanismes d'ordonnancement temps partagé et réel souple.

PROGRAMME DETAILLE

DEV C++ LINUX EMBARQUE

LINUX EMBARQUE (Suite 1)

4 jours,
28 heures

PRESENTIEL
et/ou
DISTANCIEL

PROGRAMME DU MODULE

Compilation croisée

- Présentation
  - Compilation croisée
  - Machine Hôte
  - Les compilateurs C et C++
  - Le compilateur GCC (suite)
  - Contenu d'une chaîne croisée
  - Les binutils
  - Binutils (suite)
  - Les librairies de traitement mathématique
  - Les entêtes du noyau
- Compilation du compilateur
  - Chaîne de compilation croisée
  - Alternatives à la compilation croisée
  - Les chaînes précompilées
  - Les chaînes de fabrication de cross compilateurs
  - Les composants de la chaîne
  - Processus de fabrication de la chaîne soi-même
- Appels compilateurs/compilateurs croisés
  - Édition de lien
  - Les étapes de compilation
  - Exemples
- Mise en pratique dans l'EDI ECLIPSE ou autre EDI

Buildroot

- Présentation
  - Utilisation de BuildRoot
  - Buildroot : principe
  - Paquetages nécessaires à Buildroot
  - Compilation de Buildroot
  - Les paramètres qu'on peut ajouter
- Utilisation
  - Exécution
  - Les fichiers générés
  - Les fichiers .conig
  - Utiliser la toolchain générée
  - Utiliser une chaîne de compilation externe
- Installation de la distribution
  - Préparation de la carte Micro SD
  - Agrandir la partition à toute la mémoire SD
  - Divers réglages
  - Customiser la cible
- Paramétrage Linux temps réel Linux RT, XENOMAI

Divers

- Paramètres spécifiques
  - Configuration de l'ordonnanceur
  - Exemples lignes cron
  - Clavier français
  - Notes à propos des modifications de configuration buildroot
- Services réseau supplémentaires
  - Connexion sécurisée avec SSH et transferts de fichiers par SCP
  - Serveur httpd de Busybox
  - Ajustement d'horloge système avec NTP
  - Autres astuces et documentations
- Ajout de programmes tiers
  - Ajout de programmes spécifiques
  - Les patches
  - Niveaux de patch

OBJECTIFS

- Découvrir les outils de développement sous Linux
- Maîtriser les mécanismes d'ordonnancement temps partagé et réel souple.

PROGRAMME DETAILLE

DEV C++ LINUX EMBARQUE

LINUX EMBARQUE (Suite 2)

4 jours,
28 heures

PRESENTIEL
et/ou
DISTANCIEL

PROGRAMME DU MODULE

gdb

- Suivre l'exécution
  - Breakpoint - watchpoints - catchpoints
  - gdb : point d'arrêt
  - Watchpoint
  - Gestion des points d'arrêt
  - Les commandes pas à pas
  - Signals
- Affichage des variables
  - Revenir en arrière
  - La pile d'appel
  - Modification du contexte
  - print variable
  - Autres affichages
  - Printf dynamique
  - Break avec liste de commandes

Cas avancés

- Debuguer à distance
  - Debug croisé
  - Installation sur l'équipement distant
  - Mise en route sur l'équipement host
  - Remarques sur le debug croisé

OBJECTIFS

- Découvrir les outils de développement sous Linux
- Maîtriser les mécanismes d'ordonnancement temps partagé et réel souple.

PROGRAMME DETAILLE

AJC FORMATION

DEV C++ LINUX EMBARQUE

RASPBERRY PI

3 jours,
21 heures

PRESENTIEL
et/ou
DISTANCIEL

PROGRAMME DU MODULE

Prise en main du Raspberry Pi

- Introduction : présentation du Raspberry Pi, des systems-on-chip BCM2835/2836/2837 et du processeur Arm 1176.
- Distribution Linux pour Raspberry Pi : téléchargement, installation, test.
- Découverte de la distribution : paramètres essentiels, outils standard.
- Utilisation de base : configuration système, utilisateurs, interface graphique.
- Utilisation de Linux sur Raspberry Pi : avantages et inconvénient d'un système sur carte SD.

Configuration et personnalisation

- Réseau : configuration du réseau (Ethernet+WiFi), Internet, connexion distante.
- Services réseau : démarrage des services, choix adapté à un système embarqué.
- Mise à jour : installation de paquets, mise à jour.
- Serveurs : configuration d'un serveur Web sur le Raspberry Pi.
- Contrôle à distance : déport d'affichage et prise de contrôle à distance.

Entrées-sorties et interfaces de communication

- Interface RS-232 : communication entre Raspberry Pi et PC. Traces de boot.
- Entrées-sorties GPIO : utilisation depuis le Shell, dans un programme C.
- Interruptions et GPIO : détection des changements d'état d'une GPIO.
- SPI : dialogue en SPI avec un microcontrôleur.
- I^2C : interrogation en $I^2$2 d'un capteur de température.
- Bluetooth : identification, connexion, communication.

Programmation applicative sur Raspberry Pi

- Programmation en C/C++ : compilation native ou compilation croisée, débogage, exemples.
- Scripts Shell : exemples de scripts pour initialisation du système ou lancement d'applications.

Personnalisation avancée

- Recompilation du noyau : intérêts de la recompilation, configuration ajustée.
- Drivers supplémentaires : compilation et d'installation de drivers externes.
- Configuration "from scratch" : mise en œuvre d'un système entièrement personnalisé avec Buildroot.

OBJECTIFS

- Prendre en main le Raspberry Pi pour installer une distribution standard
- Ajuster et configurer le système en fonction de ses besoins spécifiques
- Communiquer en utilisant les interfaces d'entrées-sorties comme RS-232, SPI, I^2C, GPIO…
- Développer des applications personnalisées pour Linux embarqué
- Recompiler le noyau du système et ajouter des drivers supplémentaires

PROGRAMME DETAILLE

FONDAMENTAUX ET NORMES

DEV C++ LINUX EMBARQUE

AGILE SCRUM

1 jour,
7 heures

PRESENTIEL
et/ou
DISTANCIEL

PROGRAMME DU MODULE

Méthodes agiles

- Présentation des familles de conduite de projet
- Méthode prédictives
- Méthodes adaptatives

Cycle des projets

- Présentation des fondamentaux de la conduite de projet
- Expression des besoins
- Analyse
- Conception
- Réalisation
- Vérification et validation

Présentation de Scrum

- Conduite de l'équipe projet
- Gestion de projet généraliste
- Spécification dynamique
- Adaptation aux projets logiciels

Rôles dans un projet Scrum

- Les acteurs
- Répartition des responsabilités
- Client
- Equipe
- Scrum master

Itérations

- Présentation des phases de SCRUM
- Objectifs
- Version
- Sprint
- Scrum

Suivi du projet SCRUM

La communication dans SCRUM : Meetings

- La communication dans SCRUM
- Réunion quotidienne
- Revue de sprint

Les indicateurs dans SCRUM : Planification

- La mise en place des objectifs et des indicateurs dans SCRUM
- Estimation de charge
- Organisation des tâches et préséance
- Gestion des risques et indicateurs de pilotage

Travail journalier

- L'organisation du travail quotidien
- Espace de collaboration
- Répartition des tâches par objectif

Relation avec le client

- Les engagements réciproques MOA/MOE
- Spécification des besoins
- Respécification
- Validation et vérification
- Implication

Outillage SCRUM

- Outils pour le suivi, l'analyse, les tests logiciels

Conclusion

- Adapter SCRUM, voir les limites
- Spécificité du développement logiciel
- SCRUM et XP

OBJECTIFS

- Acquérir les fondamentaux de la méthodologie

PROGRAMME DETAILLE

COMPORTEMENTAL

DEV C++ LINUX EMBARQUE

PRESENTER SES NOUVELLES COMPETENCES

1 jour,

7 heures

PRESENTIEL
et/ou
DISTANCIEL

PROGRAMME DU MODULE

Les bases de la communication

- Ecoute active
- Le questionnement
- Reformulation et feedback

La communication verbale et non verbale

- Importance de la communication non verbale
- Savoir se présenter à l'oral
- Postures – Attitudes – Discours

Les profils comportementaux

- Les 4 profils
- Auto évaluation
- Développer son adaptabilité relationnelle

Développer son Capital Talents

- Définition d'un talent
- Talent vs points forts
- 5 stratégies pour gérer ses points faibles

OBJECTIFS

- Se présenter en entretien tout en mettant en valeur ses nouvelles compétences en les considérant acquises

PROGRAMME DETAILLE

PROJET

DEV C++ LINUX EMBARQUE

PROJET FINAL

13 jours,
91 heures

PRESENTIEL
et/ou
DISTANCIEL

PROGRAMME DU MODULE

Déroulement du module

- Les stagiaires travaillent en toute autonomie, en binôme. Ils sont libres d'effectuer les choix adaptés, de développer les parties dont ils jugent avoir le plus besoin et d'apporter leurs propres solutions aux problèmes posés.
- Le formateur encadre les stagiaires par sa présence et répond aux questions. Il intervient pour épauler un binôme en difficulté ou pour faire le point à l'ensemble du groupe sur des notions non acquises. Il peut être amené à approfondir ou compléter certaines connaissances.

Enoncé

- Mise en œuvre d'un projet C sur carte Raspberry Pi
  - Élaboration d'un projet de pilotage et gestion des périphériques pour l'administration d'un robot mobile

OBJECTIFS

- Mettre en application les acquis de la formation en complétant les mini projets réalisés dans tout le cursus

NOUS CONTACTER

AJC FORMATION
01 81 51 64 85
formonsnous@ajc-formation.fr
6 rue ROUGEMONT
75009 PARIS

www.ajc-formation.fr
www.ajc-classroom.fr