

SOFTWARE DEVELOPER C++

Filière



PROGRAMME
DE LA FILIERE

Programme

OBJECTIFS

- Exploiter et administrer l'environnement Linux
- Acquérir les bases de la programmation structurée en langage C
- Apprendre à développer objet en C++
- Maitriser les concepts de Big Data
- Utiliser les outils indispensables au bon développement d'applications
- 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

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

METHODES ET MOYENS PEDAGOGIQUES

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

- Minimum BAC +4 informatique ou niveau équivalent scientifique (Maths ..)
- De la région IDF

POSTES VISES

- Développeur C, Développeur C++, Développeur Système embarqué, Ingénieur Développement C++, Développeur Logiciel Embarqué C-C++, Ingénieur Logiciel C++ QT ...

LIEU

- Distanciel

CERTIFICATION / ATTESTATION

- Attestation de formation
- Certification Expert en informatique et système d'information - RNCP36286- Niveau 7 (BAC +5)

Programme - Contenu pédagogique

COMPORTEMENTAL	RÔLE ET COMPORTEMENT DU CONSULTANT OBJECTIF « QUALITÉ » DE LA MISSION	2 jours
	TRAVAIL EN ÉQUIPE	1 jour
FONDAMENTAUX	ALGORITHMIQUE	3 jours
UNIX - LINUX	PRÉSENTATION LINUX	1 jour
	COMMANDES LINUX	2 jours
	SHELL SCRIPT	2 jours
	ADMINISTRATION LINUX	2 jours
	ARCHITECTURE LINUX	2 jours
	SYSTÈME DE FICHIERS	1 jour
FONDAMENTAUX	BIG DATA	5 jours
DEVELOPPEMENT D'APPLICATIONS	GIT	2 jours
	ECLIPSE	1 jour
	PROGRAMMATION C	5 jours
PROJET	PROJET C	3 jours
DEVELOPPEMENT D'APPLICATIONS	DEV. OPEN SOURCE	2 jours
	C VERS C++ 11	14 jours
	QT	5 jours
	SQLITE	1 jour
	DEBUG	2 jours
PROJET	PROJET C++	5 jours
EMBARQUE	LINUX EMBARQUÉ	5 jours
COMPORTEMENTAL	PRÉSENTER SES NOUVELLES COMPÉTENCES	1 jour
	CONDUITE DE RÉUNION	1 jour
	GESTION DU TEMPS ET DES PRIORITÉS	1 jour
PROJET	PROJET FINAL & SOUTENANCE - APPLICATION EMBARQUÉE	11 jours

80 JOURS



PROGRAMMES
DÉTAILLÉS



COMPORTEMENTAL

ROLE ET COMPORTEMENT DU CONSULTANT

2 jours,
14 heures



DISTANCIEL

PROGRAMME DU MODULE

Pourquoi s'intéresser aux comportements en tant que consultant ?

- Qu'est-ce qu'un comportement ? Qu'est-ce qu'un rôle ?
- En quoi les comportements peuvent faire la différence ?
- Pourquoi choisit-on d'adopter un comportement ? Le processus d'apprentissage d'un « savoir-être »

Adopter la meilleure stratégie de coopération pour mieux travailler en équipe

- Comment agir pour des développer des relations positives et durables ?
- La théorie CRP

Savoir communiquer et éviter les malentendus

- Pourquoi la communication passe-t-elle mal : les filtres, le cadre de référence ?
- Savoir utiliser l'écoute active : questionnement ouvert et reformulation
- Savoir convaincre : comment influencer positivement les échanges

Comment faire évoluer ses comportements

- Qu'est-ce qui conditionne nos comportements ?
- Sur quel levier agir pour ajouter des « cordes à son arc »

Comprendre sa personnalité et mieux cerner celle des autres

- Savoir se situer et comprendre en quoi notre personnalité se traduit à travers nos comportements
- Situer les autres et comprendre leur

mode de fonctionnement pour mieux coopérer

Développer son intelligence émotionnelle pour modifier ses comportements

- Qu'est-ce que l'intelligence émotionnelle ?
- En quoi notre QE est-il déterminant par rapport à nos comportements
- Apprendre à gérer son stress pour éviter les comportements inadaptés
 - Le stress : de quoi parle-t-on ?
 - Comment prévenir le stress et le gérer ?

Appréhender le rôle des croyances et de l'éducation dans nos comportements

- Qu'est-ce qu'une croyance ?
- Pourquoi conditionnent-elles nos comportements ?

L'assertivité et l'empathie pour mieux travailler en équipe

- Qu'est-ce que l'assertivité ? Qu'est-ce que l'empathie ?
- La notion de respects des besoins et de gagnant-gagnant
- Savoir recadrer un comportement qui ne nous convient pas et renouer avec des relations positives

OBJECTIFS

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

LE TRAVAIL EN EQUIPE

1 jour,
7 heures



DISTANCIEL

PROGRAMME DU MODULE

Le travail en équipe

- Définition
- La dynamique de groupe
- La structuration de l'équipe de travail
- La taille de l'équipe
- Les facteurs d'influence
- Les comportements
- Les styles de leadership
- Les points clés de réussite du travail en équipe.

La dynamique de groupe

- Les facteurs de cohésion et de dissociation
- La vie affective du groupe et son évolution dans le temps

La structuration de l'équipe

- Sa mission
- Ses objectifs
- Les ressources et les moyens
- L'information et le suivi d'activité

Les facteurs d'influence

- Les facteurs de démoralisation
- Les facteurs de cohésion

Les comportements

- Individuels et de groupe

Les points clés de réussite du travail en équipe

- Savoir écouter et s'exprimer
- Savoir accepter le consensus
- Savoir négocier.
- Respecter les autres.

- Savoir mettre en œuvre une méthode de travail qui vise à atteindre les objectifs fixés

OBJECTIFS

- Comprendre la dynamique d'une équipe
- Susciter la participation et l'engagement
- Utiliser les techniques et les outils appropriés pour agir en équipe
- S'organiser au sein d'une équipe
- Communiquer efficacement quel que soit son rôle



FONDAMENTAUX

ALGORITHMIE

PROGRAMME DU MODULE

Maîtriser les outils de l'algorithmique

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

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

Apprendre à raisonner sur un algorithme

Maîtriser les notions de fichiers

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

3 jours,
21 heures



DISTANCIEL

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



UNIX-LINUX



PRESENTATION LINUX

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



COMMANDES LINUX

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
- Commandes utilitaires : xargs, tee, cmp, comm, paste, sed
- Expressions régulières

OBJECTIFS

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

SHELL SCRIPT

2 jours,
14 heures



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 incondtionnels

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

Chaîne 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.

ADMINISTRATION LINUX

2 jours,
14 heures



DISTANCIEL

PROGRAMME DU MODULE

Installation

- Philosophie Debian
- Préparation de l'installation
- L'installateur 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.

ARCHITECTURE LINUX

2 jours,
14 heures



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.

SYSTÈME DE FICHIERS

1 jour,
7 heures



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, NFS, 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.



FONDAMENTAUX



BIG DATA

PROGRAMME DU MODULE

Introduction

- Introduction au Big Data : de quoi s'agit-il ?
- Exemples pratiques
- Les technologies concernées
- Les outils
- Les langages :
- Hortonworks
- MapR
- Cloudera
- IBM Watson
- Démystification du Big Data
- Les acteurs principaux
- Les différents métiers du Big Data (Administrateur, Architecte, Développeur, Data Analyst, Data Scientist, DPO, Data Owner, CDO...)

Enjeux et évolutions du Big Data

- La qualité des données
- Le traitement des données
- La protection des données
- L'image de la donnée
- L'humanité des données

Enjeux stratégiques et organisationnels

- Le challenge technique
- Investissement dans la capacité de stockage
- Investissement dans l'analyse
- Le Web sémantique
- Les nouvelles techniques de Data mining
- L'enjeu économique
- L'impact organisationnel
- La conduite du changement
- L'apparition de nouveaux métiers

Travaux pratiques : Définir un scénario avec un sujet Big Data à réaliser par un prestataire.

Les technologies utilisées dans des projets Big Data

- Les outils de stockage
- L'écosystème Hadoop (Apache, Hortonworks, Cloudera, MapR, IBM, Oracle)
- Focus sur les outils
 - Le paradigme MapReduce.
 - Le système de gestion des fichiers distribués HDFS
 - Echange de données via Sqoop, Flume, NIFI, Kafka
 - Analyse des données avec Pig et Hive
 - Les bases NoSQL
 - Ordonnanceur Hadoop : Oozie
 - Moteur de recherche : Solr, ElasticSearch
 - Spark : framework de calcul distribué in memory
- Les solutions de visualisation (Microstrategy, Tableau, QlikView ...)

Sécurité, éthique et enjeux juridiques

- Assurer la protection des données
- L'anonymisation d'une donnée
- Le contrôle d'intégrité
- Le chiffrement d'une donnée
- Qu'est-ce que la blockchain
- Cas d'usage : Bitcoin

Travaux pratiques : Réaliser une architecture Big Data avec L'écosystème Hadoop du scénario réalisé précédemment .

OBJECTIFS

- Appréhender l'ensemble des enjeux et facteurs à prendre en compte pour réussir l'intégration du Big Data dans la vision large du SI

A decorative graphic consisting of a central gold circle. Three horizontal lines pass through the circle: a dark blue line on top, a gold line in the middle, and a dark blue line on the bottom. The text "DEVELOPPEMENT D'APPLICATIONS" is centered within the gold circle.

DEVELOPPEMENT
D'APPLICATIONS

GIT

2 jours,
14 heures



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, Gitis, 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

ECLIPSE

1 jour,
7 heures



DISTANCIEL

PROGRAMME DU MODULE

Introduction

- Les objectifs et les principes d'Eclipse.
- Les concepts de base : vue, éditeur, perspective, espace de travail, projet, répertoire, fichier, nature.
- Le démarrage d'Eclipse. Fonctions d'aide. Gestion des vues et des perspectives. Gestion des ressources.
- Fonctions de navigation, de recherche. Gestion des projets.

Le debug

- La perspective Debugger dans Eclipse.
- Les différents Breakpoints et Watchpoints.
- L'inspection des variables ou expressions, la navigation dans la pile d'exécution.
- Le contrôle de l'exécution.
- Le debug réparti.

Gestion des versions avec GIT

La configuration d'Eclipse

- La gestion des préférences.
- La gestion des propriétés.
- Les références entre projets.
- L'ajout de Plug-ins et Features.
- Le paramétrage d'Eclipse.

OBJECTIFS

- Prendre en main l'interface de développement Eclipse
- Mettre en oeuvre les différents outils Eclipse pour développer des applications : édition, compilation, debug
- Configurer Eclipse pour qu'il corresponde à vos besoins de développement
- Gérer les versions d'une application



PROGRAMMATION C

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.



PROJET



SOFTWARE DEVELOPER C++

PROJET C

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.

3 jours,
21 heures



DISTANCIEL

OBJECTIFS

- Mettre en application les acquis de la formation sur le langage C



DEVELOPPEMENT
D'APPLICATIONS

DEV OPEN SOURCE

2 jours,
14 heures



DISTANCIEL

PROGRAMME DU MODULE

Connexion à distance

- Connexion via une liaison série
- Connexion non sécurisée via IP
- Connexion sécurisée via IP
- ssh permanent
- Connexion ssh et déport de l'affichage X

Administration élémentaire de l'OS

- le super utilisateur
- les commandes de base
- Sauvegarde
- cron et tâches automatisées
- KDE, Gnome, XFCE
- Installations de la distributions, d'un paquetage, synaptics, logithèque, en ligne de commande, ajouter des dépôts
- Installation via les sources
- Dépendances

Les Makefile

- Syntaxe de base
- mode enrichi
- Autotools, automake, autoscan, alocal
- qmake

Les gestionnaires de versions

- Principe
- SVN, utilisation gestion de conflits
- Git
- Sourceforge
- Créer un serveur SVN

Documentation avec Doxygen

- Principe
- Installation

- Configuration
- Syntaxe

Les environnements de développement

- Eclipse
- QtCreator
- Clion
- Emacs
- Code::Block

Techniques de débogage

- gdb
- débogueurs graphiques ddd, débogueurs intégrés
- Autres outils : electric fence, strace, ltrace, nm
- Débogage par fichier core
- Tuning des fichiers core
- Débogage à distance sur machine cible

Déploiement des logiciels

- Philosophie des paquets Debian
- Création de paquets Debian
- Debhelper dpkg-buildpackage
- Contrôle qualité des paquets
- Création d'un dépôt de paquets

OBJECTIFS

- Expliquer aux développeurs comment utiliser tous les outils du monde Opensource indispensables au développement d'applications



C VERS C++

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



C VERS C++ (Suite 1)

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
- Typeld
- 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



C VERS C++ (Suite 2)

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 intelligents

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

DESIGN PATTERNS

Introduction

- Problème de conception en programmation Objet
- Présentation des Design Pattern
- Design Pattern du
- GoF
- Principe de conception en programmation Objet

Les Design Patterns de comportement

- Itérateur,
- Stratégie,
- Template Method,
- Etat,
- Observateur,
- Médiateur,
- Visiteur
- Mise en œuvre pratique

Les Design Patterns de création

- Singleton
- Fabrique Abstraite
- Builder
- Mise en œuvre pratique

Les Design Patterns de structure

- Composite,
- Pont,
- Adaptateur,
- Décorateur,
- Façade,
- Proxy,
- Value Object
- Mise en œuvre pratique

Conception avec les Design Patterns

- Design Pattern et bibliothèque standard du C++
- Mise en œuvre systématique
- Refactoring

OBJECTIFS

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

QT

5 jours,
35 heures



DISTANCIEL

PROGRAMME DU MODULE

Présentations Qt

- Historique, intérêt, domaine d'application, les mots clés
- Licences et implication pour le développement industriel
- L'environnement de développement
- Rappels C++

Présentation de QtCreator

- Caractéristiques
- Projets/Modèles
- Types de projets : console, application, librairies
- Utilisation
- Les différentes aides
- La complétion
- Astuces éditeurs
- Compilation et débogage

qmake et fichiers pro

- Compilation
- Les différents fichiers générés
- Directives de compilation
- Débogage

Programmation Qt5

- L'objet QObject
- Les grandes familles d'objets
- Structure d'un programme
- QCoreApplication et QApplication

QtDesigner

- Philosophie
- Les différents objets graphiques
- Détail de QLineEdit
- QLayout et QSpacers
- Les menus et les QAction

Signaux et Slots

- Déclaration d'un Signal et d'un Slot

- Connexion d'un Signal et d'un Slot
- Utilisation de moc et les fichiers générés
- Quelques classes Qt5

QString

- QList, QStringList
- QVariant
- QDate
- QCompleter
- QMap, QHash
- Les fenêtres
- QDialog
- Fenêtre ui : QMessageBox, QFileDialog
- Ouverture directe de QWidget
- Classes évoluées
- Qdebug, QResources
- QSettings
- QRegExp
- Modèle MVC

Les Views : QListView, QTableView, QColumnView, QTreeView

- Les différents modèles associés
- Les équivalents sans modèles, QListWidget, QTableWidget, QColumnWidget, QTreeViewWidget
- Internationalisation

Traduction et internationalisation

- tr
- commande TRANSLATION
- Utilisation de QLinguist
- Traduction dynamique

WebKit, Phonon

- Utilisation d'un module
- Utilisation de Webkit
- Utilisation de Phonon

OBJECTIFS

- Apprendre à utiliser des bibliothèques de classe pour développer des applications à interfaces graphiques

SQLITE

1 jour,
7 heures



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`
- Ouvrir une connexion SQLite persistante et créer 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`
- Placer le pointeur de résultat au début : `sqlite_rewind`
- Déplacer le pointeur de résultat vers une ligne : `sqlite_seek`
- Fermer la connexion à SQLite : `sqlite_close`

Enregistrements

- Enregistrer une UDF agrégante 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.

SQLITE (Suite)

1 jour,
7 heures



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
- Pragma

OBJECTIFS

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

DEBUG

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

2 jours,
14 heures



DISTANCIEL

OBJECTIFS

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



PROJET



SOFTWARE DEVELOPER C++

PROJET C++

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.

5 jours,
35 heures



DISTANCIEL

OBJECTIFS

- Mettre en application les acquis de la formation sur le langage C++



EMBARQUE

LINUX EMBARQUE

5 jours,
35 heures



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.

LINUX EMBARQUE (Suite 1)

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 bibliothèques 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 .config

- 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

gdb

- Suivre l'exécution
 - Breakpoint - watchpoints - catchpoints
 - gdb : point d'arrêt
 - Watchpoint
 - Gestion des points d'arrêt
 - Les commandes pas à pas
 - Signals

5 jours,
35 heures



DISTANCIEL

OBJECTIFS

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

LINUX EMBARQUE (Suite 2)

PROGRAMME DU MODULE

- 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

5 jours,
35 heures



DISTANCIEL

OBJECTIFS

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



COMPORTEMENTAL

PRESENTER SES NOUVELLES COMPETENCES

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

1 jour,
7 heures



DISTANCIEL

OBJECTIFS

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



CONDUITE DE REUNION

PROGRAMME DU MODULE

Faire le point sur ses pratiques actuelles

- Faire le bilan des réunions existantes : points forts, points faibles
- Augmenter la pertinence dans la sélection des participants
- Lutter contre les réunions stériles et réduire le temps passé en réunion (sans perdre en efficacité)

Organiser une réunion et en définir l'objectif

- La préparation et l'organisation matérielle
- Le cadrage de la réunion : objectif, durée et règles du jeu
- Les conditions nécessaires à l'implication des participants

Structurer ses réunions pour les rendre productives

- Utiliser les techniques adaptées à chaque réunion : réunion de service, réunion d'information ascendante et descendante, réunion de négociation, réunion de résolution de problèmes avec consensus ou avec concertation
- Formaliser pendant et après la réunion : conclure, valider et formaliser les points clés de la réunion, rédiger un compte-rendu (pertinence des informations et rapidité de diffusion)

Exercer les fonctions clés de l'animateur pour faire fonctionner efficacement le groupe de travail

- Développer ses capacités d'écoute
- Répartir les rôles pour être plus efficace
- Faciliter les échanges et la production d'idées

- Connaître et repérer les phénomènes de groupe pour mieux les utiliser
- Favoriser la créativité en utilisant des techniques appropriées
- Gérer les participants difficiles

Gérer les comportements des participants

- Réaliser votre « casting »
- Fixer le rôle des participants
- Reconnaître les comportements types des participants pour mieux comprendre leurs réactions
- Réguler les échanges et distribuer la parole
- Gérer les désaccords
- Aboutir à un plan d'action partagé

OBJECTIFS

- Savoir organiser une réunion productive : l'avant et l'après
- Gérer les comportements des participants
- Acquérir des techniques d'animation pour rendre les réunions participatives

GESTION DU TEMPS ET DES PRIORITES

PROGRAMME DU MODULE

Le temps : un allié de la croissance professionnelle

Connaître les différentes manières de structurer son temps

- Types de personnalités et structuration du temps
- Bilan de ses pratiques actuelles et de l'influence de son environnement
- Prise de conscience individuelle, premier diagnostic et niveaux de motivation de chacun

Savoir faire des choix

- Clarifier sa mission et les tâches qui en découlent
- Fixer et fractionner des objectifs
- Hiérarchiser ses priorités
- Savoir filtrer, sélectionner les véritables urgences

Maîtriser son temps sans subir

- Déterminer et agir sur les "voleurs de temps"
- Mieux renoncer pour mieux choisir

Gérer son temps avec les autres

Savoir dire "non"

- Gérer les interruptions
- Savoir déléguer

Utiliser ses forces positives

- Mieux connaître son capital énergie, ses rythmes de travail
- Contacter ses ressources positives, s'en servir comme multiplicateur d'énergie
- Savoir se concentrer, se motiver,

s'arrêter, se relaxer

Intégrer le stress

- Rôle du stress, personnalités sensibles
- Se servir du "bon" stress, se protéger du "mauvais" stress
- Gestion des situations de stress les plus fréquentes ou cas particuliers

Qu'acceptez-vous de changer ?

- Déterminer les points réalistes de son contrat de changement
- Visualiser les résultats, modéliser ceux qui savent gérer leur temps

1 jour,
7 heures



DISTANCIEL

OBJECTIFS

- Acquérir des outils et des méthodes de gestion du temps afin de mettre en place des comportements nouveaux
- Prendre conscience de son comportement
- Reprendre le contrôle de son temps



PROJET

PROJET FINAL & SOUTENANCE APPLICATION EMBARQUEE

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.

11 jours,
77 heures



DISTANCIEL

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

