

Projet de Programmation en C et/ou Shell

TELECOM Nancy 2015/2016

Modalités:

- Le projet se fait par binôme.
- Quatre sujets sont proposés au choix. Ce qui est demandé est le minimum à rendre, toute extension éventuelle est la bienvenue. Une interface graphique peut être fournie en fonction des sujets mais elle n'est pas obligatoire.
- Le projet est à rendre le 30 mai au plus tard.
- Une soutenance est prévue le 06 juin 2016

Ce qu'il faut rendre le 30 mai 2016

Une archive du code source de votre projet nommé `<noProjet>-<Nom1>-<Nom2>-G<nogroupe>.zip` est à déposer sur Arche et doit contenir en plus de vos sources :

- un compte rendu (cf plus bas) ;
- un fichier texte nommé INSTALL qui explique comment compiler et exécuter le projet ;
- un fichier texte nommé README qui contient vos noms et une présentation succincte du projet ;
- un Makefile.

Important. le code source doit être suffisamment commenté mais pas trop afin de ne pas nuire à la lisibilité de votre code source. Avoir un commentaire par prototype de fonction et devant les structures des données les plus importantes, par exemple, semble judicieux.

le compte rendu

Le compte rendu au format PDF (10 pages maximum) doit être organisé comme suit :

1. une page de garde contenant au moins vos noms, groupe et titre du projet ;
2. en guise d'introduction, un résumé court de ce que fait votre logiciel ;
3. sorte d'un bref cahier de charge contenant la liste des services rendus ;
4. les choix de conception retenus : exposer brièvement la structure globale (un schéma éventuellement commenté pourrait convenir), les structures de données utilisées, les algorithmes choisis et les difficultés rencontrées. Il est intéressant d'identifier le problème de conception jugé le plus important dans la réalisation du projet, l'expliquer et donner la solution retenue à comparer éventuellement avec d'autres solutions ...

Important. Il ne s'agit pas d'expliquer ou de joindre en annexe le code source, ni d'expliquer chaque fonction en détail mais d'en présenter une synthèse du projet.

5. une présentation visuelle de la réalisation à l'aide d'une ou deux captures d'écran de votre logiciel,
6. un paragraphe expliquant comment vous avez géré votre code source (gestion de versions, hébergement),
7. un décompte du nombre d'heures engagées sur chaque élément du projet et par quel membre du binôme (conception, codage, tests, rédaction du rapport).
8. Une liste des sites et autres sources d'informations que vous avez utilisés.

La soutenance orale de 10 minutes (5 minutes de présentation et 5 minutes de questions) par binôme:

- chaque membre d'un groupe exposera le travail qu'il a effectué
- Vous vous attacherez à expliquer brièvement ce que fait votre programme: nous savons lire du code, il n'est pas nécessaire de détailler l'action de chaque ligne. En revanche expliciter les structures de données choisies et les modules de votre programme semble judicieux.

Remarques

- Les points suivants devront faire l'objet d'une attention particulière lors de l'écriture du programme : sa correction, sa lisibilité et sa clarté, sa modularité (développement de petites fonctions, répartition du source en plusieurs fichiers regroupant des fonctions de façon logique), sa portabilité, ainsi que le choix des structures de données et si possible l'efficacité des algorithmes. le développement.
- La tricherie sera **sévèrement punie**.
- L'évaluation prendra en compte les points suivants :
 - les sujets ne sont évidemment pas tous de la même difficulté ;
 - le respect de l'énoncé et l'originalité de la réalisation ainsi que les extensions éventuelles (algorithmes employés, services rendus ; ...)
 - la qualité technique du projet : organisation, modularité, découpage en fonctions, choix des instructions, algorithmes, efficacité, gestion des erreurs ...
 - le travail en équipe : gestion et organisation ;
 - le compte rendu.

Sujet 1: Gestion d'un calendrier

Dans ce projet, on souhaite réaliser un programme qui permet la gestion d'un agenda **en ligne de commande** avec une application au choix, exemple, gestion de l'emploi du temps d'une école comme Télécom Nancy. Pour l'affichage du calendrier, on utilisera une application Web comme *google calendar* par exemple. La version la plus simplifiée du programme doit permettre :

- la création d'un évènement et l'affectation d'un créneau ;
- la sauvegarde de l'agenda sur un support permanent (fichiers).

Toute amélioration et/ou extension de cette version simplifiée en fonction de l'application finale retenue est la bienvenue.

Sujet 2: Gestion de budget

Le but de ce sujet est de réaliser une application qui permet à un utilisateur, à partir de son relevé de compte bancaire (format à définir), de :

- classer ses dépenses et revenus. Par exemple pour les dépenses, on peut imaginer des catégories comme : charges immobilières, habillement, ... Chaque catégorie peut être aussi découpée en sous catégories ;
- fournir des statistiques sur les dépenses/revenus mensuelles et éventuellement des graphiques pour voir leur évolution (gnuplot <http://www.gnuplot.info/> peut être utilisé) ;
- de définir un budget maximal par poste de dépense et à être alerté dès que l'on s'approche des seuils ;
- de sauvegarder tous les données en utilisant des fichiers.

Sujet 3: Vérification d'orthographe

Le but est d'écrire une application qui vérifie l'orthographe d'un document (fichier texte) en se basant sur un dictionnaire (un autre fichier avec un format à définir). Ce programme doit être en mesure de détecter les fautes d'orthographe et de proposer des corrections possibles. Une fois le document corrigé, le programme permet de sauvegarder le texte corrigé.

Sujet 4: Visualisation d'un réseau de transport

Le but de ce sujet est de simuler un réseau de transport de marchandises. Pour simplifier, on considère une seule marchandise. On dispose d'un certain nombre de dépôts reliés entre eux par des liaisons qui permettent le transport de la marchandise en question. Chaque paire de dépôts ne dispose pas forcément d'une liaison directe. Le programme à réaliser doit permettre à un utilisateur de :

- définir son réseau (le nombre de dépôts, leurs localisation, les liaisons existantes, ...). On doit être en mesure de définir la topologie manuellement ou de la générer aléatoirement : on spécifie le nombre de dépôts et on génère aléatoirement leurs positions, le dépôt source, le (les) dépôt(s) destinataire(s), le coût éventuel d'une liaison (cf plus bas),
- visualiser le réseau. Il est imposé d'utiliser *graphviz* <http://www.graphviz.org/>, un logiciel libre de visualisation de graphes.
- trouver un chemin qui permet le transport de la marchandise d'un dépôt à un autre avec visualisation graphique de ce chemin.