# Licence 2 - Info 42

**Semestre 4 - Année 2015**

*Projet Systèmes d'exploitation* • à rendre 2 semaines avant les examens

## Cadre général du projet

Le projet à pour but d'appliquer les concepts des systèmes d'exploitation afin de réaliser l'une des applications décrites dans la suite du document : 1) une application de synchronisation, sauvegarde de répertoires, 2) un jeu des années 80 en mode multi-joueurs en réseau.

## Contraintes pour la réalisation

Le projet est à réaliser par binôme. La programmation se fera en utilisant la plate-forme Java. Il n'est pas obligatoire de développer une interface graphique. Votre programme devra être conçu de façon modulaire (packages, interfaces, classes respectant une encapsulation stricte, classes utilitaires, constantes, fichiers de configuration, etc.) et il devra refléter la conception en couches d'un système d'exploitation (les couches et leurs fonctionnalités sont à détailler dans le rapport). Les structures de données que vous utiliserez ne sont pas nécessairement dynamiques (structure à base de références). L'emploi des membres de classe déclarés *static* devra être justifié.

Vous devez faire une présentation de votre projet deux semaines avant la semaine des examens. Ce jour là vous remettrez un rapport papier dont les consignes strictes de présentation sont : au moins 15 pages , police 12 maximum, marges 2cm maximum. Le rapport doit comporter au moins les éléments suivants :

- une analyse fonctionnelle du sujet, précisant, entre autre, toutes les règles de fonctionnement de votre application ;
- une description des structures de données envisagées et retenues ;
- la spécification des classes principales et des méthodes essentielles ;
- l'architecture logicielle de votre application, c'est-à-dire, dans le cadre du module Info42, une conception en couches fonctionnelles à l'image de ce qui est réalisé dans l'architecture d'un système d'exploitation. Pour chaque couche vous spécifierez les classes qui implémentent les services de la couche fonctionnelle.
- une description des algorithmes principaux ;
- un jeu de tests montrant que votre programme fonctionne.

Il est conseillé de rédiger le rapport en utilisant LaTeX car il permet d'inclure facilement des extraits de code source avec une mise en évidence des éléments syntaxiques. Il est recommandé d'utiliser des schémas pour illustrer vos propos. Des diagrammes UML comme le diagramme de classes ou de séquences peuvent être ajoutés à votre rapport.

Une archive [1] (tgz ou jar) de l'ensemble de code source et de l'exécutable de votre programme devra être produite à la date du jour de la démonstration, puis envoyée par mail, ou déposée dans vos pages personnelles (`public_html` de votre répertoire).

L'évaluation se fait sur la base du rapport papier, de la qualité technique de la réalisation, du déroulement de la démonstration. Lors de la démonstration vous devrez avoir préparé un scénario et des jeux de tests afin de présenter les fonctionnalités de votre application (vous disposerez de 10 minutes).

## Sujet 1 : FileSynchro

L'objectif est de développer une application comparable à la commande rsync, à l'application TimeMachine, ou encore à l'application en ligne OwnCloud (équivalent libre de DropBox), permettant de sauvegarder et synchroniser des répertoires et sous-répertoires entre plusieurs machines au travers du réseau. Les utilisateurs de l'application devront être authentifiés et devront pouvoir consulter les différentes versions d'un même fichier et éventuellement les restaurer.

Le principe est le suivant :

- l'utilisateur sélectionne un répertoire source et un répertoire cible qui contiennent une arborescence de fichiers. Le répertoire source est forcement sur la machine sur laquelle l'utilisateur est connecté. Le répertoire cible peut être sur la même machine, par exemple sur un autre système de fichiers (autre disque) ou sur une machine distante (dont l'utilisateur aura spécifié l'adresse) dans ce cas il est nécessaire d'avoir une authentification.
- tous les fichiers et sous-répertoires qui ne sont pas présents dans la cible sont transférés (recopiés), pour ceux qui existent déjà dans la cible on compare la date pour déterminer le plus récent :

1. Les autres formats d'archive ne sont pas autorisés

  - si le plus récent est sur la source, celui de la cible est archivé dans un sous-répertoire spécifique dit répertoire d'archive (par exemple `Archive/2015-02-15` ou la date représente la date de la synchronisation). La structure (répertoires et sous-répertoires) de l'archive doit être identique à la structure de la cible.
  - si le fichier de la cible est plus récent que celui de la source : soit il est déplacé dans archive soit une action est demandée à l'utilisateur.
- si des fichiers ou répertoires ne sont plus présent sur la source mais existent encore sur la cible, ils sont déplacés dans archive.

Il doit être possible de spécifier les actions par défaut au lancement de l'application (définition de l'emplacement du répertoire archive, comportement en cas d'ambiguïté). Dans la partie client, le sens de la synchronisation doit pouvoir être spécifié par l'utilisateur (de la machine A vers la machine B ou l'inverse)

L'application comporte deux parties c'est-à-dire deux programmes exécutables :
- une partie client utilisée par les utilisateurs ;
- une partie serveur.

Il n'est pas nécessaire de réaliser une interface graphique, ce n'est d'ailleurs pas recommandé pour la partie serveur. Afin d'éviter de parcourir un arborescence pour trouver le date, taille de chaque fichier, il est utile de stocker, à chaque connexion, sur le client et sur le serveur la liste des fichiers existant et leur taille (éventuellement une clé de hachage MD5). Un thread pourra prendre en charge cette opération. Java permet de lancer une commande du shell depuis un programme. Les commandes `find ou ls -alR` vous permettront d'établir facilement la liste des fichiers d'une arborescence.

La partie réseau doit être développée après le que votre programme fonctionne en local c'est-à-dire sur une seule machine et entre deux répertoires.

# Sujet 2 : LodeRunner

Lode Runner est un jeu vidéo de plates-formes développé par Douglas Smith et publié par Brøderbund en 1983. Le joueur évolue dans un décor (tableau) constitué d'échelles, de murs de briques et de passerelles. Le but du joueur est de ramasser des éléments (paquets, lingots, etc) qui ont été disséminés dans le tableau. Une fois que tous les éléments sont collectés le joueur doit sortir du tableau par l'échelle la plus haute. Des adversaires (automatiques ou humains) essayent d'attraper le joueur, celui ci peut creuser les briques autour de lui pour ensevelir ses adversaires. Il peut également creuser les briques pour accéder à une portion du tableau. Les adversaires ensevelis réapparaissent en haut du tableau. Si un joueur est coincé ou touché par un adversaire, il perd une vie (sur les cinq prévues en début de partie) et réapparait en haut du tableau.

Vous avez libre choix d'adapter ou d'ajouter des règles en respectant l'esprit du jeu. Vous devez implanter un mode collaboratif (avec plusieurs joueurs qui collectent les éléments) et un mode combat (pour lequel les adversaires peuvent être contrôlés par d'autres humains sur d'autres machines).

Vous devez impérativement réaliser les fonctionnalités suivantes :
- les adversaires doivent pourvoir être contrôlés soit par des joueurs (mode combat) ou soit par le programme (IA) ;
- il doit être possible de jouer en réseau à plusieurs (mode coopératif) avec plusieurs machines dont l'une est serveur ;
- les scores doivent être enregistrés sur un serveur et la liste des meilleurs joueurs doit être maintenue à jour, il doit être aussi possible de définir des équipes en mode collaboratif ;
- la description de tableaux de jeu (niveaux) au moyen de fichier texte.

Avant d'implanter le réseau vous devez avoir réalisé et testé le fonctionnement du jeu en mode machine simple.

**Références :**
- `http://fr.wikipedia.org/wiki/Lode_Runner`
- `http://en.wikipedia.org/wiki/Lode_Runner` donne plus de détails
- `https://archive.org/details/msdos_Lode_Runner_1983_1983` jouable en ligne