

TP 2 : Simulation d'une course

Sujet du TP

On souhaite simuler le déroulement d'une course (quelle qu'en soit l'activité). Dans un premier temps, on suppose que le nombre d'équipes et de coureurs par équipe sont fixés. On suppose de plus que, si les coureurs partent dans un ordre quelconque, il n'y a pas d'abandon (tous les coureurs partent et arrivent).

Le temps est mesuré par un chronomètre, démarré au début du programme et incrémenté avant chaque nouvel événement.

Bases de départ

Vous trouverez dans le répertoire `/users/prof/scheuer/AP/TP2` un fichier d'archive compressé nommé `course.tgz`, ainsi qu'un exécutable `verif_course`. L'archive contient, répartis dans plusieurs répertoires, les fichiers en-têtes et sources du programme `course`, un fichier `Makefile` permettant de compiler, de faire le ménage et d'archiver les fichiers importants. Certains des fichiers nécessitent d'être complétés, afin d'obtenir un exécutable `course` aussi proche que possible de l'exécutable `verif_course`.

Travail à effectuer

Des rappels concernant les commandes *Unix* et les raccourcis clavier sous *Emacs* sont disponibles à la fin de cet énoncé.

1. Copiez l'archive compressée `course.tgz` sur votre compte, par exemple dans un répertoire `AP`, puis décompressez la et extrayez son contenu (avec `guntar zxvf`); renommez le répertoire `Top` généré avec un nom plus explicite (`Course`, par exemple); vous pouvez ensuite détruire l'archive décompressée (commande `rm`).
2. Copiez l'exécutable `verif_course` dans le même répertoire que le fichier `Makefile`, puis compilez les fichiers fournis, soit en tapant `make` dans un répertoire, soit (RECOMMANDÉ) en tapant `M-x compile` sous *Emacs*, une fois un fichier chargé; comparez le comportement de l'exécutable `course` obtenu avec celui de `verif_course`.
3. Étudiez les différentes classes (définies dans les fichiers en-têtes, disponibles dans le répertoire `inc`), puis l'implémentation de leur méthodes les plus longues (détaillée dans les fichiers sources, disponibles dans le répertoire `src`).
4. Ensuite seulement, essayez de compléter les différents fichiers afin d'obtenir un exécutable `course` similaire à `verif_course`.

Résultat attendu

Lorsque vous avez obtenu un exécutable `course` semblable à `verif_course`, ou à la fin du TP, envoyez à l'adresse `Alexis.Scheuer@loria.fr` l'ensemble des modifications que vous avez effectuées sur les classes existantes (ET NON LA TOTALITÉ DES FICHIERS). Si vous avez conservé les fichiers originaux dans un répertoire, vous pouvez utiliser la commande `diff` pour gagner un peu de temps.

Compléments FACULTATIFS

SI VOUS AVEZ TERMINÉ AVANT LA FIN DU TP, je vous invite à étudier le fichier `Makefile`, afin de comprendre comment les différentes actions (compilation, nettoyage, archivage) ont été spécifiées.

Ensuite, vous pouvez aussi réfléchir à une implantation des classes `equipe` et `course` afin d'autoriser un nombre indéfini (variable avec le temps) d'équipes et de coureurs par équipe. Pour cela, il faut utiliser la classe générique `list` définie dans le fichier `/usr/local/dp/gnu/gcc/include/g++-3/stl_list.h`.

Rappel divers

Commandes *Unix* :

- Copie d'un fichier (copy): `cp <origine> <destination>;`
- Suppression d'un fichier (remove): `rm <fichier>;`
- Décompression d'un fichier (g-unzip): `gunzip <fichier>;`
- Extraction de fichiers d'une archive (eXtract Verbose File): `tar xvf <archive>;`
- etc...

Raccourcis clavier *Emacs* :

- Sauvegarder (save): `C-x-s` ;
- Sauvegarder sous un autre nom (write): `C-x-w` ;
- Charger un fichier (file): `C-x-f` ;
- Recharger un fichier: `C-x-v` ;
- etc...

Pour ces raccourcis, `C-` indique qu'il faut appuyer **simultanément** sur la touche contrôle (`Ctrl`) et sur les touches nommées ensuite, `M-` indique qu'il faut appuyer **simultanément** sur la touche méta (`Escape`) et sur les touches nommées ensuite.