Informatique I (SMA/SPH) : SÉRIE NOTÉE

(sujet A : 28 novembre 2013, 10:15–11:00)

INSTRUCTIONS (à lire attentivement)

IMPORTANT ! Veuillez suivre les instructions suivantes à la lettre sous peine de voir votre série annulée dans le cas contraire.

1. Cette série doit être réalisée **individuellement !** L'échange d'information avec un tiers est bien entendu **strictement interdit**. Le code rendu doit être le résultat de **votre propre production**.
2. Vous avez droit à toute documentation papier et accès à votre compte comme d'habitude. Par contre, vous devrez **impérativement** faire cette série sur les ordinateurs des salles CO et sur votre compte. Vous ne pouvez pas utiliser d'ordinateur personnel, ni de téléphone portable, ni aucun autre matériel électronique.
3. Vous n'avez **pas** le droit d'imprimer, ni d'envoyer d'email, ni de vous connecter sur une autre machine ou un site Internet externe pendant les deux fois 45 minutes des deux séries.
4. Pour démarrer la série, cliquez sur le lien (rouge) « Série Notée » du calendrier du cours pour accéder à la page de résumé `http://icwww.epfl.ch/~chappeli/examens/`.
   Cette page reprend les instructions ci-dessous :
   (a) La *première* chose à faire est de vous **INSCRIRE** sur
   `http://icwww.epfl.ch/~chappeli/examens/SNA/`
   (b) Une fois l'inscription validée, déconnectez vous de la page d'inscription en suivant le lien qui y est indiqué, puis fermez la page/l'onglet correspondant.
   (c) Téléchargez (en suivant le lien) le fichier `ordi.cc` indiqué (vous devez **impérativement** travailler sur ce fichier, en aucun cas partir d'un fichier vide) et sauvegardez le dans votre répertoire `myfiles/cpp`.
   Si vous ne savez pas le faire depuis le navigateur, vous pouvez toujours le faire depuis le terminal à l'aide de la commande suivante :
   `wget http://icwww.epfl.ch/~chappeli/infomaph/admin/ordi.cc`
   (d) Travaillez comme d'habitude sur ce fichier (vous pouvez utiliser l'éditeur de votre choix). Pensez en particulier à le *sauvegarder régulièrement*.
5. Pour rendre la série notée :
   (a) Reconnectez vous sur la page d'inscription.
   (b) Fournissez le bon fichier `ordi.cc` et **validez** !
   (c) Attendez la confirmation orale des assistants avant de vous déloguer de la machine et de quitter la salle.
   Ces opérations ne sont **à exécuter qu'une seule fois**, lorsque vous avez fini.
   Vous avez **jusqu'à 11:00** pour soumettre votre rendu.
6. Le fichier `ordi.cc` rendu ne devra plus être modifié après le rendu, et cela jusqu'à la publication des notes (en cas de litige/problème).
7. Lisez attentivement et *complètement* les questions de façon à ne faire que ce qui vous est demandé. Si l'énoncé ne vous paraît pas clair, ou si vous avez un doute, demandez des précisions à l'un des assistants.

Exercice : Répartition de tâches [sur 40 points]

Avant de commencer, assurez vous d'avoir bien pris connaissance de toutes les instructionssur la page précédente.

On s'intéresse ici à écrire un programme permettant de répartir des tâches à exécuter sur des ordinateurs.

On vous demande, en répondant aux questions ci-dessous, de réaliser ce programme dans le fichier `ordi.cc` ***fourni***, en utilisant le langage C++ et une approche « procédurale » avec le moins de duplication de code possible.

Question 1 : Tâches et ressources [sur 8.5 points]

Les tâches a répartir sont représentées par un nom et leurs besoins en ressources. Une ressource est représentée par une taille mémoire (un entier) et un besoin en calculs, noté `mips` (pour « Million Instructions per Second »), de type `double`.

Dans le fichier `ordi.cc` fourni :

1. **[3 points]** Définissez une **structure** `Ressource` correspondant à la description ci-dessus.
2. **[3 points]** Définissez ensuite une **structure** `Tache` correspondant à la description précédente. Une seule ressource suffit par tâche.
3. **[2.5 points]** Définissez enfin une **fonction** `affiche` qui prend en paramètres une tâche et l'affiche en respectant *strictement* le format suivant :

   nom (m=*mémoire*, op=*mips*),

   Par exemple, pour la première tâche déclarée dans le `main()` fourni, la fonction `affiche` donne :
   `tache1 (m=20, op=10),`

Question 2 : Modélisation et affichage des ordinateurs [sur 8 points]

Un ordinateur est représenté par son nom, sa capacité mémoire (nombre entier), une liste de tâches qui lui sont assignées et une liste de processeurs, représentés ici simplement par leur seule capacité de calcul, qui sera un entier.

4. **[3 points]** Définissez une **structure** `Ordinateur` correspondant à la description précédente.
5. **[5 points]** Complétez la **fonction** `affiche` fournie qui prend en paramètres un ordinateur et affiche ses propriétés comme indiqué ci-dessous.

   Par exemple, si la tâche 1 a été affectée à l'ordinateur `ordi1` du `main()` fourni, l'affichage produira :

```
Ordinateur: ordi1
  Mem   : 100
  Procs : 100, 200, 150,
  Taches: tache1 (m=20, op=10),
```

suite au dos ☞

Question 3 : Capacité de calcul moyenne d'un ordinateur [sur 8.5 points]

On vous demande maintenant de calculer la capacité de calcul moyenne d'un ordinateur. Il s'agit *simplement* de la moyenne (arithmétique) des capacités de calcul de ses processeurs.

6. **[6 points]** Définissez une **fonction** `calculer_opmoyen` qui prend en paramètres un ordinateur et retourne sa capacité de calcul moyenne.

Question 4 : Allocation des tâches aux ordinateurs [sur 15 points]

On vous demande enfin d'écrire une fonction `allocation` qui prend en paramètres une liste d'ordinateurs et une liste de tâches. Cette fonction va essayer d'allouer chaque tâche de la liste à un ordinateur de la liste. L'algorithme d'allocation (simple) est le suivant :

7. **[5 points]** On considère les ressources disponibles offertes *au départ* par chaque ordinateur comme étant sa mémoire et sa capacité de calcul moyenne, calculée au moyen de la fonction `calculer_opmoyen` précédente (celle-ci sera calculée *une fois pour toute* et non pas recalculée à chaque fois).
8. **[10 points]**
   Pour chaque tâche, on décide de l'affecter au premier ordinateur trouvé dans la liste tel que sa mémoire et sa capacité de calcul restantes soient supérieures ou égales aux besoins de la tâche .
   Si tel est le cas, la tâche est affectée à l'ordinateur en question et ses ressources disponibles sont remises à jour simplement en soustrayant les ressources de la tâche à celles encore restantes de l'ordinateur.

Si à ce stade vous compilez et exécutez le code fourni puis complété, vous devriez obtenir le résultat suivant :

```
Ordinateur: ordi1
  Mem   : 100
  Procs : 100, 200, 150,
Ordinateur: ordi2
  Mem   : 80
  Procs : 300, 200, 150,
Ordinateur: ordi3
  Mem   : 200
  Procs : 200, 300, 200,
=== Allocation ===
Ordinateur: ordi1
  Mem   : 100
  Procs : 100, 200, 150,
  Taches: tache1 (m=20, op=10), tache2 (m=50, op=100), tache4 (m=30,
                                                              op=40),
Ordinateur: ordi2
  Mem   : 80
  Procs : 300, 200, 150,
  Taches: tache5 (m=60, op=110), tache8 (m=10, op=105),
Ordinateur: ordi3
  Mem   : 200
  Procs : 200, 300, 200,
  Taches: tache3 (m=10, op=220), tache7 (m=180, op=10),
```