

IFT1169 – TRAVAIL PRATIQUE #2 – 30 septembre 2015**« Une commande Linux à la sauce C++ ... »
Mohamed Lokbani**

Équipes : le travail est à faire en monôme (une seule personne).

Remise : une seule remise est à effectuer par voie électronique **le vendredi 30 octobre 2015, 23h59 au plus tard**, sans possibilités de prorogation.

Conseils : n'attendez pas le dernier jour avant la remise pour engager votre travail. Vous n'aurez pas le temps nécessaire pour le réaliser.

But : ce TP a pour but de vous faire pratiquer le traitement des entrées et sorties et la gestion dynamique des objets.

Énoncé : le but de ce travail est de réaliser quelques options de la commande « find », disponible sous Linux.

La commande « find » permet de chercher un élément donné. On peut associer à cette recherche une série d'options permettant de formater le résultat. Voici quelques exemples d'utilisation de la commande « find ».

find . -ls	Affiche le contenu du répertoire courant
find . -type f -ls	Affiche uniquement les fichiers
find . -mtime -3 -type f -ls	Affiche les fichiers qui n'ont pas plus de 3 jours d'âge
find . -mtime +3 -type f -ls	Affiche les fichiers qui ont plus de 3 jours d'âge
find . -mtime -3 -type f -size +200c -ls	Affiche les fichiers qui ont plus de 3 jours d'âge et dont la taille dépasse 200 octets.
find . -perm +755 -mtime -3 -type f -size +200c -ls	Affiche les fichiers qui ont plus de 3 jours d'âge, dont la taille dépasse 200 caractères et qui ont au plus la permission 755 (rwxr-xr-x).

Les fichiers de données fournis ont été obtenus en utilisant aussi l'option « -maxdepth 1 ». Elle n'est pas nécessaire pour ce travail, car les données fournies sont limitées à un seul niveau.

L'option « maxdepth » permet d'éliminer de la recherche tous les sous-répertoires à partir de la profondeur « n ». La valeur de « n » est positive non nulle.

Vous ne pouvez pas utiliser ni les classes (ou fonctions) génériques ni les « STL ».

Étape -1- : on s'intéresse dans cette étape à la commande suivante :

tp2a15 -perm +755 -mtime -3 -type f -size +200c < fichier_entree.txt

- Réalisez un programme qui permet de traiter la précédente commande. À ce stade, vous ne devez pas implémenter le fonctionnement des 4 options. Vous devez juste valider leur présence.

- Votre programme doit afficher sur la sortie standard que vous allez rediriger dans le fichier « fichier_sortie_1.txt », le contenu du fichier lu en entrée.

- Exécutez la commande « diff fichier_entree.txt fichier_sortie_1.txt ». Comme les fichiers sont identiques, vous ne devez obtenir aucun affichage en sortie. Dans le cas contraire, vous avez un bogue.

Étape -2- :

- Complétez votre programme en stockant le contenu du fichier « fichier_entree.txt » dans une structure de données appropriées. Pour ce travail, votre structure doit-être à base de tableaux. Vous ne pouvez pas utiliser les conteneurs de la STL. La taille du tableau représente le nombre d'enregistrements dans le fichier. Pour ce travail pratique, nous allons fournir cette valeur. Elle est indiquée dans la première ligne du fichier « fichier_entree.txt ».

- Votre programme doit afficher par la suite en sortie le contenu du tableau.

- Vous allez rediriger cette sortie dans le fichier « fichier_sortie_2.txt ».

- Exécutez la commande « diff fichier_entree.txt fichier_sortie_2.txt ». Comme les fichiers sont identiques, vous ne devez obtenir aucun affichage en sortie. Dans le cas contraire, vous avez un bogue.

Étape -3- :

- Implémentez l'option «perm +/-n ». Cette option va permettre d'afficher les éléments ayant une certaine permission. Le signe (-) affiche les éléments ayant au minimum la permission n, le signe (+) au maximum la permission n, et aucun signe, la permission demandée (n).

- Vous allez rediriger cette sortie dans le fichier « fichier_sortie_3.txt ».

- Exécutez la commande « diff tst_3.txt fichier_sortie_3.txt ». Comme les fichiers sont identiques, vous ne devez obtenir aucun affichage en sortie. Dans le cas contraire, vous avez un bogue.

Étape -4- :

- Implémentez l'option « mtime n ». Cette option va permettre d'afficher les éléments qui remontent à moins de n jours (entre aujourd'hui et il y a n jours) si la valeur de n est négative. Elle affichera les éléments dont les dernières modifications remontent à plus de n jours, si la valeur de n est positive. Sans signe, la commande affiche les fichiers ayant la date n.

- Vous allez rediriger cette sortie dans le fichier « fichier_sortie_4.txt ».

- Exécutez la commande « diff tst_4.txt fichier_sortie_4.txt ». Comme les fichiers sont identiques, vous ne devez obtenir aucun affichage en sortie. Dans le cas contraire, vous avez un bogue.

Étape -5- :

- Implémentez l'option « type f ». Cette option ne va permettre d'afficher que les fichiers. Les répertoires seront éliminés de l'affichage.

- Vous allez rediriger cette sortie dans le fichier « fichier_sortie_5.txt ».

- Exécutez la commande « diff tst_5.txt fichier_sortie_5.txt ». Comme les fichiers sont identiques, vous ne devez obtenir aucun affichage en sortie. Dans le cas contraire, vous avez un bogue.

Étape -6- :

- Implémentez l'option « size n ». Cette option va permettre d'afficher les éléments dont la taille est supérieure à n, si n est un entier positif ou inférieure à n s'il est un entier négatif. La valeur de n est accompagnée d'une lettre.

Ces lettres sont comme suit : c (octet), k (kilooctet), M (mégaoctet) et G (gigaoctets).

- Vous allez rediriger cette sortie dans le fichier « fichier_sortie_6.txt ».

- Exécutez la commande « diff tst_6.txt fichier_sortie_6.txt ». Comme les fichiers sont identiques, vous ne devez obtenir aucun affichage en sortie. Dans le cas contraire, vous avez un bogue.

Makefile

L'ensemble des commandes de compilation doit figurer dans le fichier « makefile ». Vous devez inclure aussi dans ce fichier les différentes commandes « diff » décrites dans ce travail et nécessaires pour valider votre travail.

Fichiers à remettre : vous devez remettre le fichier « tp2A15.zip ». Par ailleurs, vous devez inclure dans votre remise un rapport décrivant le travail effectué.

Remise : vérifiez que le code que vous avez produit (devant normalement fonctionner correctement chez vous) fonctionne aussi bien sur les ordinateurs de la DESI. La procédure a déjà été décrite dans le TP#0.

Le répertoire de remise sur le serveur est tp02.

Barème : ce TP2 est noté sur 10 **points**. Les points sont répartis comme suit :

Étapes 1, 3, 4, 5, et 6 : 1 point chacune, pour un total de **5 points**.

Étapes 2 : **3 points**.

Rapport : **2 points**.

Makefile : **-2 points s'il est absent**.

En plus du précédent barème, vous risquez de perdre des points dans les cas suivants

- La non-remise électronique (volontaire ou par erreur) est sanctionnée par la note 0.
- Un programme qui ne compile pas : 0.
- Un programme qui compile, mais ne réalise pas les choses prévues dans la spécification : 0.
- Les avertissements (warnings) non corrigés : cela dépend de la quantité! À partir de -0.25 et plus.
- Aberration dans le codage : même si tous les chemins mènent à Rome, faites l'effort nécessaire pour éviter de prendre le plus long!

Des questions à propos de ce TP? Une seule adresse : dift1169@iro.umontreal.ca

Pour faciliter le traitement de votre requête, inclure dans le sujet de votre courriel, au moins la chaîne [IFT1169] et une référence au tp02.

Mise à jour

30-09-2015 diffusion de l'énoncé.