Collège Lionel-Groulx Département d'informatique
Hiver 2017

Système exploitation 2

Cours: 420-KHC-LG

Travail Pratique intégration final

Compétences à développer

Ce laboratoire vous permettra de développer les compétences liées aux aspects suivants :

- Utiliser les fonctions disponibles des appels systèmes pour programmer une nouvelle commande Linux en langage C/C++,
- Utiliser les notions relatives aux scripts pour développer un script fonctionnel,
- Porter un programme qui fonctionne sous Windows vers le système Linux,
- Installer un serveur Web et peupler le site Web.

Travail à réaliser (peut-être réaliser en équipe de 2 personnes MAXIMUM)

Ce travail est composé de quatre (4) parties.

Partie 1 : Réalisation d'un script pour créer des comptes usagers automatiquement.

On vous demande de réaliser un script qui permettra de créer un certain nombre d'usager sur le système. Votre script permettra de créer un certain nombre de compte usager en lisant un fichier texte contenant les différentes informations sur les comptes.

La compagnie comporte plusieurs départements. En fait, il existe au total 10 départements pour lesquels on doit créer un certain nombre de compte usager. La compagnie vous donne le format du fichier qui contiendra les informations des comptes à créer. Ce fichier s'appelle « userfile.txt » et a le format suivant :

NomCompte:MotPasse:Groupe:Shell

À vous de créer ce fichier pour faire vos tests. Par contre, j'utiliserai exactement le même format de fichier mais avec plusieurs compte (au delà d'une centaine) pour tester votre script.

On vous mentionne que le 4e champ (le shell) peut-être vide. Auquel cas, le shell « bash » sera alors le shell par défaut, c'est-à-dire, le « /bin/bash ».

Rappelez-vous que pour tester si une variable est vide, on peut utiliser l'opérateur « -z » par l'intermédiaire d'un if. Ex. : if [-z $Var]; then ... Si la condition est vraie, alors la variable $Var est vide.

Ex. :	stef:stef:Developpement :	Ce compte utilisera donc le « bash ».
	jack:jack:Web:bash	Celui-ci utilisera le « bash ».

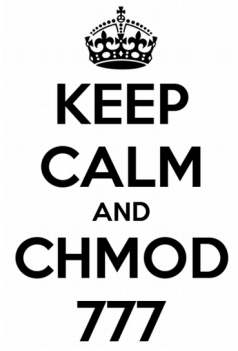
Dans l'exemple ci-dessus, un compte nommé « jack » sera créé avec le mot de passe « jack ». Cet usager sera dans le groupe d'usager nommé « Web » et aura le « /bin/bash » comme interpréteur de commande. (shell)

Les répertoires « maison » (home directories) des usagers prendront la forme suivante :

« /home/nom_groupe/nom_compte.

Ainsi l'usager « jack » ci-dessus aura le répertoire maison : « /home/Web/jack » et son interpréteur de commande sera le « /bin/bash ».

Partie 2: Réalisation d'une commande Linux à l'aide du langage C/C++



Cette portion du travail consiste à réaliser une nouvelle commande Linux à l'aide du langage C++.

Au lieu de se rappeler les différents caractères à utiliser dans la commande chmod ou encore les valeurs octales, nous aimerions plutôt utiliser directement les caractères r w x dans la commande tel qu'on aimerait les voir afficher.

De façon plus spécifique:

- Il s'agit de permettre à l'usager de pouvoir changer les droits d'un fichier ou d'un dossier en utilisant la syntaxe rwx r-- — directement à la ligne de commande.

- Votre commande se nommera « monchmod » et effectuera une opération similaire à la commande « chmod » que vous connaissez.

Donc au lieu d'écrire chmod 750 lefichier.txt
votre commande permettra d'écrire :

./monchmod rwxr-x--- lefichier.txt

Exemple d'utilisation :

./monchmod r-xr----- Dossier
Change les droits du dossier pour r-xr-----

./monchmod r-------- ex1.cpp
Change les droits du fichier ex1.cpp pour lecture à l'usage seulement.

Gestion des erreurs

Si le fichier ou le dossier n'existe pas, un message d'erreur approprié est affiché

Si il n'y a pas 9 caractères pour les droits, une erreur est affichée.
Ex. : ./monchmod r-- r-- ex2.cpp → erreur minimum de 9 caractères non respectés
Si le caractère n'existe pas, un message d'erreur est également affiché.
Ex. : ./monchmod rpx –t r-- → Erreur caractère non reconnu.

Si le caractère n'est pas au bon endroit, un message d'erreur est affiché.
Ex. : ./monchmod wrxr--r-- → Mauvais caractère à la position 1 w

Partie 3: Les vaisseaux (Exercice 6.2) version Linux!

Cette partie vous permettra de transformer un programme destiné à la plate-forme Windows vers une plate-forme Linux.

Vous devrez porter votre exercice 6.2 (Les vaisseaux) pour le faire fonctionner de la même façon sous Linux.

Partie 4 : Site Web et Linux – Intégration avec conception de site Web.

Cette partie s'intègre avec le cours de conception de site Web. Vous devrez aller porter sur le serveur Web les pages Web de votre site pour le rendre fonctionnel dans un environnement réseau restreint.

- **Installation d'un serveur Web (Apache)**
- **Distribuer les pages Web de votre site sur le serveur.**

Pour le contenu du site Web, vous prendrez les pages Web que vous avez développé dans le cadre de votre cours de Conception de site Web.

Pour l'installation de apache, vous devrez l'installer tel que vu en classe.

Vous devrez ensuite montré que votre site Web fonctionne avec vos pages et sur le serveur local que vous aurez installé.

Barême et pointage

- Portion 1 : Fonctionnement 30%
- Portion 2 : Fonctionnement 40%
- Portion 3 : fonctionnement : 10%
- Portion 4 : fonctionnement : 10%
- Commentaires et logique du programme et code source : 10%

Fonctionnement :

- Montrer le fonctionnement de chaque partie lors du :
  - Groupe 1 : Laboratoire du jeudi 11 mai 2017.
  - Groupe 2 : Laboratoire du vendredi 12 mai 2017.
  - Groupe 3 : Laboratoire du mercredi 10 mai 2017.

Remise du code source ET de l'exécutable

Pour chaque partie, vous devrez remettre le code source ET l'exécutable dans le boite de remise Col.Net le MERCREDI 17 MAI 2017 avant 23:59.