IUT du Limousin
BUT 1ère Année

Année 2024/2025

Debian 11

SAé 1.03 « Installation d'un poste pour le développement »

Compétence évaluée: Administrer des systèmes communicants complexes – Niveau 1

Apprentissages critiques:

- Identifier les différents composants (matériels et logiciels) d'un système numérique
- Utiliser les fonctionnalités de base d'un système multitâches / multiutilisateurs
- Installer et configurer un système d'exploitation et des outils de développement

0 – Contexte

La réalisation de ce travail se fera en binôme. Vous travaillerez toujours dans les salles 108/109 sur une machine qui vous sera attribuée pour toute la durée de la SAé.

I – Informations générales

L'entreprise de formation « ButIné » désire installer sur ses postes informatiques (ensemble de PCs 64 bits) un environnement de développement sous Linux. Elle dispose de postes de travail sur lesquels le logiciel VirtualBox est installé. Le PDG de la société demande à son administrateur système d'installer sur chacune des machines une distribution DEBIAN ainsi que l'éditeur de code VisualStudioCode.

II – Première étape : Installation de Debian

Vous devez installer sur la machine qui **vous est attribuée** une distribution Debian sous VirtualBox. Vous allez paramétrer de façon rigoureuse la Machine virtuelle en choisissant notamment :

- La taille de Mémoire vive allouée
- La taille du disque dur (>16 Goctets),
- La taille de la partition Swap,
- Le nombre de processeurs alloués
- L'environnement de bureau : Gnome, KDE, Xfce..... On souhaite un environnement peu consommateur en ressources même s'il est un peu moins agréable et fonctionnel.

Vous donnerez à « votre Debian » le nom : DebSAé1.03_XXX_YYY ou XXX et YYY sont les trois premières lettres de vos noms.

Notice d'installation

III – Deuxième étape - Outils de développement/paramétrage

« ButIné » organise des formations en HTML et python. Vous installerez pour cela sous Débian l'éditeur de code VisualStudioCode. Vous vérifierez (en paramétrant si nécessaire) que VisualStudioCode permet le développement en HTML et python.

Les formations dispensées s'adressent à des étudiants de première, deuxième et troisième années. Trois groupes d'utilisateurs (un pour chaque année) sont donc à créer sur vos machines. Un compte « test » sera également crée pour chacun des groupes avec des droits judicieusement choisis.

IV – Gestion de la bibliothèque

Le PDG de la société demande à son administrateur système d'écrire plusieurs scripts permettant de gérer la bibliothèque de Butiné. Il devra écrire des scripts permettant de gérer les emprunts de livres par les étudiants adhérents à bibliothèque.

Pour cela, 4 fichiers de données seront utilisés (membres, livres, exemplaires et emprunts), contenant des champs séparés par un ' ; ' . Chaque ligne de ces fichiers représente un élément de la base de données :

- *membres* contient la liste des membres (adhérents) de la bibliothèque. Il est constitué de lignes de la forme :

  num_membre;nom _membre;prenom_membre;adresse_membre

où les champs contiennent respectivement le numéro, le nom, le prénom et l'adresse de l'adhérent (la ville). Le fichier est trié par ordre croissant de numéro de membre. Ce numéro est attribué automatiquement lors de l'inscription du membre.

- *livres* contient la liste des livres de la bibliothèque. Il est constitué de lignes de la forme :

  num_livre;titre_livre;auteur

où les champs contiennent respectivement le numéro, le titre et l'auteur du livre. Le fichier est trié par ordre croissant de numéro de livre. Ce numéro est attribué automatiquement lors de l'inscription du livre ;

- *exemplaires* contient la liste des exemplaires des livres de la bibliothèque et indique pour chacun s'il est disponible ou non. Il est constitué de lignes de la forme :

  num_livre;num_exemplaire;disponibilité

où disponibilité vaut **oui** ou **non** selon que l'exemplaire num_exemplaire du livre num_livre est disponible (non emprunté) ou non.

- *emprunts* contient la liste des emprunts en cours. Il est constitué de lignes de la forme :

  num_membre;num_ livre;num_ exemplaire;date

indiquant que l'adhérent num_membre a emprunté (et non encore rendu) à la date *date* l'exemplaire num_exemplaire du livre num_livre. Le format de date est :

jour mois année heure:minutes:secondes

où num_jour est le numéro du jour dans le mois (de 1 à 31) et num_mois est le numéro du mois dans l'année (de 1 à 12). La date courante au bon format est obtenue par la commande :

date '+%-d %-m %Y %X'

Exemples de fichiers

$ cat membres
1;dupont;albert;marseille
2;martin;marcelle;nimes
3;mart;leon;aix

$ cat livres
1;quatre_mousquetaires;dumas
2;robinson_crusoe;defoe
3;de_la_terre_a_la_lune;vernes
4;lequipe;anonyme

$ cat exemplaires
1;1;non *Ce qui signifie qu'il existe les exemplaires 1, 2 et 3*
1;2;non *du livre n°1 (quatre_mousquetaires)*
1;3;oui *et que seul le 3 est disponible*
2;1;oui
3;1;oui
4;2;oui
4;3;oui
4;4;oui

4;5;oui
4;1;non

$ cat emprunts
2;1;1;1 2 2007 14:20:10
2;4;1;1 2 2007 14:22:22
1;1;2;2 4 2007 17:37:41

L'ordre des lignes n'est pas important pour les fichiers exemplaires et emprunts.

1) Écrire un script **inscrire.bash** qui avec l'option :
   - **-a** ajoute un nouvel adhérent à la fin du fichier membres. Les 3 arguments suivants sont le nom, le prénom et l'adresse du membre. Le numéro de membre doit être attribué automatiquement par le script (incrémenter le numéro du membre figurant en dernière ligne). On suppose pour simplifier qu'il n'existe pas d'homonyme, et que les informations ne contiennent pas d'espace.
   - **-l**, le script ajoute un livre à la fin du fichier livres et ajoute ses exemplaires à la fin du fichier exemplaires. Les 3 arguments qui suivent sont le nombre d'exemplaires, le titre et l'auteur du livre. Le numéro de livre doit être attribué automatiquement (incrémenter le numéro du livre figurant en dernière ligne). Pour simplifier, on suppose que le livre n'existe pas déjà (il ne s'agit donc pas d'ajout d'exemplaire), et que les informations ne contiennent pas d'espace.

2) Écrire un script **demande.bash** qui prend 2 arguments : le nom de l'adhérent et le titre du livre. Le script vérifiera la validité de ceux-ci (présence dans leur fichier respectif) puis contrôlera si un exemplaire du livre est disponible. Si tout est correct, le script insèrera un nouvel emprunt à la fin du fichier emprunts et modifiera le fichier exemplaires (pensez à utiliser la commande sed) pour marquer l'exemplaire emprunté comme indisponible. S'il y a erreur, le script affichera le message adéquat et le code de retour sera mis à jour suivant l'erreur.

3) Écrire un script **rend.bash** qui accepte 2 arguments : le nom de l'adhérent et le titre du livre. Le script doit vérifier la validité de ceux-ci (présence dans leur fichier respectif) puis contrôler si l'emprunt est bien enregistré dans le fichier emprunts. Si oui le script le supprime du fichier et modifie le fichier exemplaires pour remettre l'exemplaire disponible (il n'est pas nécessaire de vérifier qu'il ne l'était pas).

4) Écrire un script **comptemps.bash** qui compare la date courante à une date donnée en arguments dans le même format que celui obtenu par **date '+%-d %-m %Y %X'** et renvoie :
   - 0 si la date en argument est postérieure ou égale à la date courante ;
   - 1 si la date en argument est antérieure à la date courante de plus d'un an ;
   - 2 si la date en argument est antérieure à la date courante de plus d'un mois et moins d'un an ;
   - 3 si la date en argument est antérieure à la date courante de moins d'un mois.

   Le script reçoit 4 arguments représentant la date : le premier est le numéro de jour dans le mois, le second le numéro du mois dans l'année, puis l'année et la date (*jour mois année heures:minutes:secondes*).
   La comparaison des dates est une opération fastidieuse si l'on veut prendre en compte les particularités (nombre de jours différents selon les mois et les années, etc.).
   On fera, comme lors de votre TD d'algorithmique, une comparaison sur l'année et le mois :
   Exemple pour le mois : 16/02/23 -> 16/03/23 cela fait un mois
   31/10/23 -> 30/11/23 cela ne fait pas un mois
   31/10/23 -> 01/12/23 cela fait au moins un mois.
   On procèdera de la même façon pour l'année.

5) Écrire un script **rappel.bash** qui vérifie les emprunts datant de plus d'un mois et affiche les lignes correspondantes. Pour cela, traiter le fichier emprunts ligne à ligne pour en extraire la date d'emprunt, la comparer à la date courante en utilisant **comptemps.bash**, et afficher la ligne si un rappel doit être fait.

Robuste : - contrôle de données (test)
- commentaire (grand bloc)

V – Documents à rendre

Une fois le travail terminé vous devrez remettre à votre enseignant encadrant :

- Un dossier dans lequel vous justifierez les choix faits lors de l'installation de la Débian et des différents « outils ». Ce dossier pourra également comporter les parties importantes des différents scripts demandés. Il comportera un guide d'installation de la VM. Il est à remettre au plus tard **le 10 Janvier 2025 à 12h**. Il fera au maximum **12 pages** (hors page de garde).
- Tout autre document que vous jugerez nécessaire.

Pour la présentation orale vous utiliserez un **document power point**. (Présentation installation, parties essentielles des scripts, démonstration, questions).
A titre indicatif : 25 minutes de présentation + 15 minutes de questions et test des scripts.

VI – Quelques sources

- https://fr.joecomp.com/how-install-visual-studio-code-debian-9#menu-2
- https://blog.desdelinux.net/fr/instalar-visual-studio-code-linux/
- https://grafikart.fr/tutoriels/linux-installation-vscode-1154
- https://cdiese.fr/installation-de-debian-sur-une-machine-virtuelle-virtualbox/
- https://www.youtube.com/watch?v=Mok2J__Ci3Y
- https://www.youtube.com/watch?v=Du722KSObas

Et n'oubliez surtout pas la commande **man** de Linux !!!

Bon courage !