TPS INTERNET - LINUX

1. Orientations et Consignes générales pour toutes les séances de TP

(1) Il n'y a pas de compte-rendu à rendre, et il n'y pas de séance spécifique d'examen de TP sur machine. MAIS, des questions sur ces TP peuvent être posées lors de l'examen final de l'UE.

(2) Pendre des notes personnelles, notamment dans les zones associées à la plupart des questions.

(3) Avant de répondre aux questions, bien lire intégralement, à chaque fois, chaque section (ou sous-section), pour bien identifier les questions posées (pour éviter de faire du travail inutile).

(4) Le cahier de TP donne des quelques informations complémentaires (par rapport aux Cours/TD). De plus, l'exécution de certaines commandes produira davantage d'informations que nécessaire pour répondre aux questions, et toutes ne sont pas compréhensibles (car au-delà des connaissances enseignées). Il faudra savoir extraire / déduire ce qui sera nécessaire.

(5) Si vous ne faites que « juste taper » les commandes, cela est sans intérêt/apport. Il faut chercher à prévoir/interpréter/comprendre/extrapoler les résultats obtenus, et la progression dans l'enchaînement des questions.

(6) Dans les premières parties, les commandes sont à saisir « en ligne », c'est-à-dire dans un terminal, donc en mode texte et non dans une interface graphique.

(7) Le terme « **machine** » désigne un ordinateur (ou poste de travail), une imprimante, etc.

2. Objectifs des TP

Il s'agit de commencer avec une machine qui n'a pas encore de « configuration réseau », puis d'enrichir étape par étape cette configuration jusqu'à permettre à cette machine des communications et des services « internet » avec « n'importe quelle autre machine ». Les grandes étapes sont :

(1) Faire les premières configurations minimales pour permettre à une machine d'accéder à d'autres machines situées dans environnement très « local » : configuration de la carte réseau et de l'adresse IP.

(2) Compléter la configuration pour accéder « facilement » à n'importe quelle machine située dans un environnement « local » ou « distant » : routage et serveur DNS.

(3) Enrichir cette configuration pour permettre à la machine de proposer / utiliser les services usuels accessibles via l'internet : NFS, NIS, imprimante réseau, travail à distance, FTP, serveur WEB, firewall, etc.

TOUT sera à faire sous linux. Et dans les premières étapes, ce sera **« en ligne de commande »**, ce qui nécessite de savoir utiliser/interpréter certaines notions et commandes de base dans linux (en rappelant que linux stocke les « données » dans **une arborescence de répertoires et de fichiers**).

1) **A un instant donné, dans UN terminal ou UNE application** donnée :
   - Le système d'exploitation pointe sur UNE seule position dans l'arborescence : c'est le répertoire courant.
   - Vous interagissez avec le système d'exploitation en étant « connecté en tant qu'utilisateur X » : selon le profil d'utilisateur, les droits d'interactions ne sont pas les mêmes.
2) **A un instant donné, PLUSIEURS terminaux/applications** peuvent être simultanément ouverts, sans être nécessairement dans le même répertoire courant ou avec le même utilisateur.
3) TOUTE fonction/commande/application/… a nécessairement **un comportement par défaut** … qu'il faut connaître.

3. Linux : commandes de base en ligne (dans un terminal) : rappels

Réalisez cette partie en étant connecté à votre compte habituel en salle I1/I3. Vous pouvez également réaliser le travail dans n'importe quel environnement linux, mais dans ce cas, toutes les questions ne sont pas nécessairement réalisables et les résultats peuvent selon l'environnement et le compte utilisé.

Ouvrir un terminal, et pour toutes les questions proposées, répondre et/ou tester les commandes et interpréter les résultats : tester avec différentes options et dans différents cas de figures lorsque c'est possible.

1) Quel est le symbole de la racine de l'arborescence ?

2) A quoi correspond un « chemin d'accès » à un élément dans l'arborescence de fichiers et répertoires ?
   a. chemin « absolu » :

   b. chemin « relatif » :

3) Obtenir de l'aide sur une commande à l'aide de la commande « **man** » : **man** *nom_commande*

4) Renommage de commandes : la commande « **alias** » sert à voir les « alias » (ou commandes renommées) définis par défaut à l'installation du système. Cela permet de comprendre pourquoi, éventuellement, certaines commandes ne donnent pas (ou même pas du tout) les résultats auxquels vous êtes habitués. Est-ce que des alias sont définis ? Si oui, certains vous surprennent-ils ?

5) Voir le « contenu » des éléments (fichiers, répertoires) de l'arborescence :
   a. voir le contenu d'un répertoire : **ls** … sans ou avec option et/ou arguments (chemin d'accès à un répertoire)

   b. voir le contenu d'un fichier : **cat** ou **more** suivi d'un nom de fichier

6) Se déplacer dans l'arborescence : **cd**… sans ou avec option et/ou arguments (chemin d'accès à un répertoire)

7) « Où suis-je » : **pwd** … pour identifier le répertoire courant pointé dans le terminal utilisé.

8) « Qui suis-je » : **whoami** … pour savoir quel « compte/profil » est en activité dans le terminal utilisé

9) Changer de « rôle / profil /d'utilisateur » : **su** *nom_utilisateur*

10) Localiser l'exécutable d'une commande donnée (si elle existe) : **whereis** … ou **which** **…** sans ou avec option et/ou arguments (nom ou chemin d'accès à une commande). Ex. **whereis** *su*

11) Que fait et comment s'utilise la commande **ping** *@IP*. Essayer la commande **ping** *140.93.0.15*

12) Rappel : complétion de nom à l'aide de la touche TABULATION : cette fonctionnalité est proposée par certains shells, mais pas tous. Elle permet ne taper qu'une partie d'un nom de commande ou de fichier ou de répertoire.

PARTIE CONFIGURATIONS RÉSEAUX SOUS KNOPPIX

4. Informations/données générales pour la salle de TP

Avertissement : tous les PC de la salle ne sont pas configurés exactement de la même manière (nombre de carte réseaux, prise en compte du lecteur de CD, etc.).

- **Nom du réseau (ou « domaine ») et Adresse IP du réseau local** :
  - Nom : **ups-tlse.fr**
  - Adresse IP : **130.120.12.x**
    - 130.120.12 est la référence commune à toutes les machines de la salle,
    - x est la référence qui caractérise une machine donnée

- **Nom des machines** / **Adresse IP** (tableau complet plus loin). Exemple :
  - Nom de machine : **alina**
  - Nom complet : **alina.ups-tlse.fr**
  - Adresse IP : **130.120.12.8**

- **Masque de sous réseau (netmask)**
  - Adresse IP : **255.255.252.0**
  - C'est le même pour toutes les machines de la salle de TP
  - Il a été attribué par l'université pour un de ses sous-réseaux

- **Adresse de broadcast** : elle est à calculer
  - (adresse IP d'une machine) OU BIT à BIT (complément du masque du sous réseau)

- **Passerelle (ou routeur)** : **130.120.12.1**

- **Serveur de nom (DNS) :**
  - Primaire – adresse IP : **130.120.124.102**
  - Secondaires : 195.220.43.67 ou 195.220.59.6 … (à VÉRIFIER)

- **Serveur NIS** :
  - Adresse IP : **130.120.12.121**
  - Nom du domaine NIS : **masterEEA**

- **Serveur de fichier NFS**
  - Adresse IP : **130.120.12.121**
  - Répertoires partagés avec NFS : **/home** et **/usr/local**

- **Imprimante réseau** :
  - Adresse IP : **130.120.12.43**
  - Type : LEXMARK M5811dn (à VÉRIFIER)
  - Type de Serveur : Network Printer / TCP (à VÉRIFIER)

Tableau complet des noms et adresses IP des machines :

alina	**130.120.12.8**	**felicia**	**130.120.12.18**	**julia**	**130.120.12.26**
bea	**130.120.12.10**	**gina**	**130.120.12.20**	**katia**	**130.120.12.28**
claudia	**130.120.12.12**	**helena**	**130.120.12.22**	**lea**	**130.120.12.30**
dalva	**130.120.12.14**	**isadora**	**130.120.12.24**	**maeva**	**130.120.12.32**
elisa	**130.120.12.16**				
		imp3	**130.120.12.46**	**imp4**	**130.120.12.43**

5. Configuration réseau d'une machine linux : phase 1 - configurations élémentaires

Knoppix est une distribution GNU/Linux basée sur la distribution Debian. *Knoppix* **démarre et s'exécute seulement à partir d'un CD** (CD « exécutable »)
Knoppix inclut de nombreux logiciels tels que *LibreOffice.org, the Gimp, Iceweasel, Apache, Wireshark* et des centaines d'autres logiciels. Comme tout système Linux, il dispose de toutes les interfaces nécessaires pour une connexion au réseau Internet.

5.1 Prise en main de Knoppix : démarrage, interface, éditeur de texte

a) Démarrage (boot, lancement)

Attention, par défaut Knoppix démarre en anglais avec un clavier configuré en qwerty. Pour travailler en français et avec un clavier configuré en azerty, il faudra le spécifier au bon moment. C'est-à-dire dès l'apparition du prompt[1] du premier écran de boot, il faudra saisir au clavier (qui sera encore en qwerty) :

linux lang=fr

Attention : être attentif et très rapide, car la durée d'affichage de cet écran est très courte. Si vous le manquez, il faudra redémarrer la machine.

1) Peut-on insérer le CD dans le lecteur ? Si c'est non, comment faut-il faire ? Puis, dans tous les cas, redémarrer la machine.

2) Est-ce que le redémarrage de la machine provoque la lecture et l'exécution automatique du CD ?
Si c'est oui, appeler l'enseignant. Si c'est non, expliquer la cause, puis ce qu'il faut faire pour provoquer un démarrage avec exécution du CD. Faites-le !!!
Attention : c'est à cette étape qu'il faut configurer la langue et le clavier en français.

3) **Utilisateurs** : le boot crée automatiquement 2 utilisateurs, et sans mot de passe. Ils sont nommés :
   - ***root*** : utilisateur de niveau administrateur de la machine (il a tous les droits)
   - ***knoppix***, : utilisateur de base (ses droits sont restreint)

   Important : dans la suite de ce TP, beaucoup de manipulations à réaliser nécessitent d'être administrateur (root) de la machine. Pour devenir/se loguer en tant que root, la commande est :

   su : sans paramètre

b) Interface : identification des fonctionnalités de base sur l'écran : barre d'icônes en bas

1) Dans le menu accessible dans l'icône la plus gauche, identifier comment :
   - quitter la session et Knoppix
   - ouvrir un terminal
   - ouvrir un navigateur internet

2) Est-ce que certaines commandes linux de base sont renommées par défaut par Knoppix ? Ouvrir un terminal et vérifier avec la commande **alias**. Est-ce que la commande **su** est renommée ?

[1] *prompt : désigne « l'invite de commande », c'est-à-dire le caractère (ou la chaine de caractères) qui s'affiche/signale dans une fenêtre de commande (un terminal ici), l'endroit où du texte (typiquement des commandes) peut être saisie au clavier pour être interprété par le système d'exploitation de l'ordinateur.*

3) Toujours dans la barre de menu en bas de l'écran, trouver l'icône de l'interface de gestion de la configuration réseau.
   - Comment peut-on voir l'état par défaut de l'option « activer le réseau » ?
   - Cette option est-elle activée ou non (attention à la symbolique choisie dans cette interface pour représenter cette information) ?
   - Quel sont les effets de l'état par défaut et est-ce compatible avec les manipulations à faire dans les questions suivantes (pour répondre, il faut avoir lu les 2 à 3 questions suivantes ?
   - Si c'est non, dans quel état faut-il mettre cette option ?

c) Éditeur de texte (pour modifier le contenu d'un fichire) de Knoppix : leafpad - préalable

ATTENTION : **pour un bon fonctionnement de leafpad (et éviter de « planter » le TP), avant de le lancer à partir d'un terminal, il faudra d'abord exécuter la commande suivante dans ce terminal**

xhost +

RAPPEL général : avec tout éditeur de texte, toujours s'assurer de bien faire un retour chariot à la fin de la dernière ligne du fichier édité, sinon, dans certains cas, la dernière ligne pourrait ne pas être prise en compte … ce qui est problématique surtout si cette ligne comporte des commandes.

d) Leafpad (prise en main) : paramétrage du nom de la machine (dans Knoppix)

Beaucoup de fonctionnalités d'un système d'exploitation ont besoin de faire référence à la machine sur laquelle le système d'exploitation est en cours d'exécution. C'est pourquoi il faut donner un nom à la machine.
La référence à la machine se fait via une variable appelée **hostname** (nom de l'hôte du système d'exploitation). Cette variable prend sa valeur dans un fichier, qui est (dans Knoppix) : ***/etc/hostname*.**

- Est-ce que le fichier /etc/hostname existe déjà ?
- Contient-il déjà quelque chose ?
- A votre avis, est-ce que n'importe quel utilisateur devrait avoir l'autorisation de modifier le contenu de ce fichier, ou seulement est-ce que c'est seulement l'administrateur système ? Pourquoi ?
- Quel est l'utilisateur en cours ? A-t-il le droit de modifier le fichier ? Si c'est non, changer pour devenir un utilisateur qui a ce droit de modification.
- Ajouter dans ce fichier le nom de votre machine (ce fichier ne doit contenir QUE le nom de la machine)
- Vérifier que l'enregistrement a bien marché.

5.2 Configuration de base la carte réseau

Attention, les questions/manipulations ne commencent qu'après ce premier paragraphe qui n'est que de l'information.

La première étape de configuration d'une carte réseau, **serait** de charger le module (driver) dédié au pilotage de la carte réseau. Heureusement, depuis des années la majorité des cartes réseaux sont *plug & play* et sont détectées au *boot*. Le module est donc normalement chargé automatiquement lors du démarrage. Toutefois si la carte n'était pas détectée, **il faudrait (mais ce n'est pas à faire dans ce TP)** :

- *Vérifier la présence de cartes réseaux sur le bus PCI avec la commande* **`lspci`** *(qui, notamment, affiche les informations nécessaires pour trouver le N° de la carte)*
- *Vérifier avec la commande* **`lsmod`** *la liste des modules chargés*
- *Puis, si besoin, charger, avec la commande* **`modprobe`** *le module réseau correspondant (ex. pour une carte 3com de la série 590, il* ***faudrait*** *taper* **`modprobe 3c59x`***). Tous ces drivers sont situés dans /lib/modules/3.6.11/kernel/drivers/net.*

Dans ce TP, tous les PC sont déjà équipés d'au moins 1 ou 2 carte réseau ETHERNET.

1) Taper la commande **lspci**, et relever ce qui est indiqué pour la ligne « Ethernet »

2) En pratique, pour vérifier si la carte réseau est détectée, utiliser la commande : **ifconfig *–a*** qui affiche l'ensemble des interfaces réseau disponibles. Relever :
   - Le nombre d'interfaces réseau, et parmi elle, le nombre qui ont de type ETHERNET :
   - Le nom très abrégé attribué à chaque interface Ethernet :
   - Leur adresse MAC (HWaddr) :
   - Leur adresse IP :
   - Les valeurs pour TX et RX
   - S'il y a 2 cartes ETHERNET, choisir celle nommé ETH0 (si c'est celle qui connectée au réseau, mais ça … ce sera identifié dans les questions suivantes … si ça ne marche pas avec ETH0).

3) Relancer plusieurs fois la commande **ifconfig *–a***. Les valeurs de TX et RX ont-elles changé ? Comment interpréter leurs changements ou leur non-changement ?

4) Tester si machine peut déjà communiquer avec d'autres machines ?
   - Quelles commandes utiliser pour faire ces tests ? Noter le ou les messages qui s'affiche.

   - Essayer d'interpréter les informations données par ces messages, c'est-à-dire où se situe la cause de non communication parmi les éléments matériels ou logiciels du réseau entre la machine locale et la machine cible ?

5.3 Configuration d'une adresse IP (il faut être en root)

Maintenant que l'on s'est assuré de la présence d'une carte réseau, il faut configurer « manuellement » (avec des commandes en ligne) l'interface correspondante afin de lui associer une adresse IP et les informations nécessaires à sa connexion au réseau local (masque de sous réseau, adresse de broadcast, …).

Les commandes pour réaliser ceci sont : **ifconfig** ou **`ip addr`**. MAIS, on ne détaillera ci-dessous que **ifconfig**, dans laquelle il faudra remplacer le texte en italique par des données qui indiquées dans la section 4 du sujet, ou qui sont à calculer à partir de ces données :

ifconfig *interface @IP* **netmask** *masque_sous_réseau* **broadcast** *@broadcast*

1) Avant d'utiliser la commande ci-dessus, identifier à quoi correspondent les paramètres suivant et préciser leur valeur dans ce TP :
   - interface :
   - @IP :
   - masque_sous_réseau :
   - @broadcast (à calculer) :

2) Taper la commande complète ci-dessus, puis taper à nouveau **ifconfig *–a***.
   - Quels sont les changements par rapport à la section 5.2, question 2 ?
   - S'il n'y a pas de changement, qu'est-ce que cela signifie ? Qu'est-ce qui aurait dû changer ?

3) Testez, avec la commande **ping** *@IP* la connectivité de la machine au réseau local.

   Quels messages s'affichent ? Qu'apportent-ils comme indication sur la localisation de l'éventuel problème, c'est-à-dire où se situe la cause de non communication parmi les éléments matériels ou logiciels du réseau entre la machine locale et la machine cible ?
   - tester vers une machine éteinte :
   - tester vers une machine de la salle :
   - tester vers la passerelle :
   - tester vers le serveur DNS :

4) Est-il possible de communiquer avec une machine hors du réseau (local) de la salle ?

   Testez avec les machines suivantes qui se trouvent soit dans un autre réseau de l'université, soit hors de l'université ? Quels messages s'affichent ? Qu'apportent-ils comme indication sur la localisation de l'éventuel problème, c'est-à-dire où se situe la cause de non communication parmi les éléments matériels ou logiciels du réseau entre la machine locale et la machine cible ?
   - 140.93.0.15 (laas.laas.fr)
   - 216.58.205.99 (google.fr)
   - 141.115.28.2 (www.irit.fr)
   - 80.247.237.75 (www.toulouse.fr)
   - 194.167.156.113 (www.univ-pau.fr --> frontend-ametys.univ-pau.fr)

5.4 Configuration du routage

Pour permettre à une machine de communiquer en dehors de son réseau local, il faut envoyer le trafic de cette machine vers un serveur (sur une machine) nommée ***routeur*** ou ***passerelle***, qui a pour rôle de diriger les paquets (d'un réseau local) à destination de réseaux distants vers la bonne direction (d'autres routeurs).

Pour cela, il faut indiquer (paramétrer) à la machine utilisée au moins un *routeur par défaut* qui recevra tout le trafic sortant de cette machine et qui n'est pas à destination du réseau local. A noter qu'il est possible d'indiquer plusieurs routes, c'est le cas par exemple des routeurs qui peuvent être des systèmes Linux.

La commande pour faire ce paramétrage est **route** et ses options et arguments pour ajouter une route sont :

route *add default gw* `passerelle`

De plus, cette commande **route**, quand elle est utilisée sans paramètre/argument, permet aussi de vérifier la *table de routage*.
Une autre commande à noter : **traceroute** *@IP* qui offre une possibilité supplémentaire, en indiquant tous les routeurs traversés pour arriver à destination donnée.

1) Taper la commande **route** et relever ce qui est affiché (sans chercher à tout comprendre, juste pour pouvoir comparer ensuite les changements observés après la définition du routeur).

2) Définir comme routeur celui indiqué en section 4

3) Taper à nouveau la commande **route**, et noter ce que qui a changé.

4) Comment tester la connectivité avec un réseau extérieur ?

5) Comment supprimer cette route par défaut (trouver le paramètre de la commande **`route`** qui permet cela … comment trouver ce paramètre ?) ? Supprimer là et vérifier la suppression par deux moyens complémentaires.

6) **Définir à nouveau cette route après suppression, car cela est nécessaire pour la suite du TP**

5.5 Configuration de la « résolution de nom » : configuration très locale

Il n'est pas bien pratique d'utiliser les adresses IPv4 (dans leur format numérique) pour communiquer avec une machine du réseau. Il est plus pratique et simple d'utiliser un service de *résolution de nom* qui associe une adresse IP ***xxx.xxx.xxx.xxx*** (c'est un format « numérique ») à un nom « lisible » sous la forme ***machine.reseau.pays*** (c'est un format « alphabétique).

a) Résolution de nom par configuration « très locale »

Pour cela, il faut paramétrer/définir/enregistrer sur la machine utilisée le **nom** de cette machine, puis la liste des associations *@IP <–> nom de machine* des autres machines.

1) Paramétrage du nom de la machine : cela a déjà été fait au 5.1) d), lors de la prise en main de l'éditeur de texte *leafpad*. Pour rappel, le fichier dédié[(2)] pour contenir le nom de la machine est ***/etc/hostname***. .
Vérifier que c'est bien le cas. Sinon, recommencer la manipulation du 5.1) d).

[(2)] *« Fichier dédié » veut dire un fichier dans lequel le système d'exploitation va rechercher des types d'informations spécifiques dont il a besoin (dans le cas présent, il s'agit du nom de la machine).*

2) Paramétrage des associations *@IP <–> nom de machine* des autres machines. Cette liste locale (à la machine utilisée) est nécessaire pour la résolution de nom. Le fichier dédié pour contenir cette liste est ***/etc/hosts***.

   - Existe-t-il déjà ?
   - Contient-il déjà quelque chose ?
   - Y ajouter le nom de quelques machines de la salle de TP (y compris celui de votre machine) en ajoutant des lignes de type *@IP nom*
     exemples :
     - *130.120.12.8 alina*
     - *130.120.12.10 bea*

3) Tester si la résolution de nom est maintenant opérationnelle, en utilisant la commande **ping** sur des noms de machines tels que définis dans le fichier ***/etc/hosts.*** Quels résultats ?

Cette méthode simplifie déjà certaines choses. Mais elle reste encore très/trop limitée car on ne peut pas procéder ainsi pour configurer tous les noms de toutes les machines connectées à l'Internet.

b) Résolution de nom par appel à un service/serveur de nom (DNS) … voir partie section suivante

6. Configuration réseau d'une machine linux : phase 2 – services de base l'Internet

Maintenant que la configuration réseau minimale de la machine est opérationnelle, on va améliorer la configuration de travail en permettant à la machine d'accéder à différents services proposés par « l'internet », c'est-à-dire des services universels, mis en place au niveau mondial, « normalisés », et qui ne sont pas la charge de l'administrateur de chaque machine.

6.1 Configuration de la « résolution de nom » : service DNS

Pour la résolution de noms, face aux limites de la méthode précédente, « l'internet » a mis en place une méthode totalement dédiée à ce travail de conversion *@IP nom*) : le **DNS** qui veut dire ***Domain Name System***, aussi appelé ***Domain Name Server***.

Pour qu'une machine puisse accéder à ce service, il suffit seulement d'enregistrer sur la machine le nom d'un ou plusieurs serveurs à solliciter. Le fichier dédié pour ce paramétrage est :
/etc/resolv.conf

1) Pourquoi vaut-il mieux paramétrer plusieurs (et non pas un seul) DNS ?

2) Ouvrir ce fichier ***/etc/resolv.conf***. Existe-t-il déjà ? Contient-il déjà quelque chose ? Y ajouter une (ou des) ligne(s) de type **nameserver** *@IP du serveur_de_nom*

3) Tester en utilisant des noms de machines
   - Locales, définis plus haut dans le fichier ***/etc/hosts***
   - Locales dans la salle, mais non définie dans ce fichier
   - Distantes : prendre les 2 premières éléments de la liste donnée au 5.3), question 4

4) Généralisation : ce mécanisme de résolution de nom va encore plus loin, puisque qu'il s'applique même aux « noms » de site web (puisqu'un site Web est toujours hébergé sur une machine … qui a un nom). Tester sur les 3 derniers éléments de la liste donnée au 5.3), question 4

6.2 Montage d'un répertoire distant : service NFS

Le système de fichiers de Linux permet l'accès à divers supports, localisés en local ou à distance (disque dur, cdrom, ou fichiers sur un serveur séparé) de manière transparente et intégrée à l'arborescence globale.
Cela permet d'améliorer l'environnement de travail des utilisateurs qui peuvent ainsi : disposer de davantage d'espace de travail, et réduire les risques de perte/altération de leurs fichiers.

Pour cela, il faut utiliser le principe de « **montage** » de fichiers, qui peut être réalisé l'aide de la commande **mount** dont la syntaxe à utiliser dans ce TP est :

mount ***–t*** *type_de_support* *adresse_physique_du_repertoire_distant_à_monter* *repertoire_local_de_montage*

Le 1er argument de la commande définit le service ou le type utilisé pour le montage. Dans le cas d'un **partage** de fichiers (localisés sur une machine distante) on va utiliser le service ***nfs*** (*Network File System*).

Le 2ème argument doit être l'adresse physique du répertoire distant que l'on souhaite « monter » sur la machine. Sa syntaxe est :
- serveur:/chemin_acces_au_répertoire
- où serveur est l'adresse IP (ou le nom complet) de la machine qui héberge le serveur de partage, c'est à dire le serveur nfs dans ce TP.

1) A l'aide de la commande **man**, chercher à quoi correspond le 3ème argument, que l'on a nommé ici `repertoire_local_de_montage` (attention, ce n'est pas le nom utilisé dans l'aide en ligne obtenu avec la commande man).

2) Créer, sur la machine locale, un répertoire ***/local*** (il faut impérativement respecter exactement le chemin et le nom donné). Quel est le contenu de ce nouveau répertoire ?

3) Maintenant, à l'aide de la commande **mount**, associer à ***/local***, le répertoire ***/home/m1eea_xxx*** (avec xxx, la valeur de votre compte habituel en salle de TP I1/I3) situé sur le serveur de fichier *electre* (nom de la machine qui héberge le serveur NFS).
   - Attention : cette opération peut durer plusieurs minutes
   - Quel message s'affiche lorsque l'opération est terminée ?

4) Comment vérifier que le montage a réussi ? Quelles sont avantages / dangers de ce montage ?

5) Pour effectuer un « démontage » (rompre l'accès au répertoire distant), la commande est **unmout** ou **umount** (selon la distribution linux utilisée).

Note : des montages automatiques au démarrage du système sont réalisables en plaçant des commandes de montage dans le fichier dédié ***/etc/fstab*** (dédie aux montages à faire au démarrage du système).

6.3 Information sur les Programmes « démons » pour activer certaines fonctionnalités

Cette section ne demande de manipulation, mais donne des informations pour les sections suivantes.

Le système Linux utilise un type particulier de programmes au rôle très spécifique, totalement dédié au fonctionnement du système. On les appelle usuellement des ***démon (daemon)***.

Chacun de ses programmes ont un nom (comme tout programme), et les démons de base sont usuellement regroupés dans un répertoire dédié : ***/etc/init.d***.

Les démons sont entièrement gérés/contrôlés par le système. Un utilisateur, peut seulement les lancer (ou activer), les arrêter ou les relancer en leur donnant comme paramètres : *start, stop ou restart*

Exemple, la commande pour activer un démon nommé ***nis*** sera (à utiliser plus loin) : **/etc/init.d/nis *start***

6.4 Authentification distante d'utilisateurs : service NIS – utilisation de « démons »

La notion de *compte utilisateur* que vous utilisez est en fait l'association :
- d'une *authentification* faîte par un serveur d'authentification
- et du *montage automatique* du répertoire ***/home*** contenant les fichiers (des utilisateurs) localisés sur le serveur de fichiers.

L'authentification est la vérification du nom de l'utilisateur *(login)* et de son mot de passe *(password)*. Elle est réalisée par le service NIS *(Network Information System)*.

a) Authentification NIS : configuration

Pour configurer *NIS* il faut éditer plusieurs fichiers, dont un pour indiquer quel est le serveur *NIS* et un autre pour indiquer quels sont les « services » que doit rendre *NIS*.
En fait *NIS* ne fait pas directement d'authentification, c'est le système Linux qui va s'en charger, en remplaçant le fichier contenant les *login* et *password* (c'est le fichier dédié local ***/etc/passwd***) par celui contenu par le serveur, un peu comme le ferait *NFS*.

Pour chacun des 3 fichiers ci-après, AVANT de les modifier, il faudra déterminer s'il existe déjà, et s'il contient déjà quelque chose ?

1) Éditer ***/etc/yp.conf*** et y ajouter la ligne ci-après (les valeurs des argumentes sont donnés en section 4)

domain *nom_du_domain_NIS* **server** *adresse_serveur_NIS*

2) Éditer ***/etc/defaultdomain*** et y ajouter le nom du domaine NIS (ajouter, ou remplacer l'existant).

3) Éditer ***/etc/nsswitch.conf*** et ajouter une « entrée » ***nis***, à la fin de chacune des lignes *passwd, group, shadow* (sans effacer ce qui est déjà écrit sur ces lignes).

b) Authentification NIS : activation

La configuration des fichiers seule ne suffit pas, il faut encore activer le service NIS. Cette activation passe par l'utilisation d'un *démon.* Ici le démon s'appelle ***nis***.

1) Lancer le démon ***nis***. Qu'est-ce qui s'affiche ? Est-ce que le programme est actif ?

La commande **ypwhich** doit donner le nom complet de la machine qui héberge le serveur *NIS* et la commande **ypcat** *passwd* permet d'accéder à la carte des comptes utilisateur localisée sur le serveur (ainsi qu'aux mots de passes … cryptés).

2) Tester ces commandes pour vérifier la bonne configuration de NIS

3) Se loguer sur son compte habituel m1eea_xxx avec la commande **su** *nom_utilisateur*.
   - Quelles informations permettent de savoir si l'utilisateur a bien été authentifié ?
   - Est-ce qu'il y a d'autres commande pour tester la bonne authentification ?

4) Est-ce que la bonne authentification permet à l'utilisateur authentifié d'accéder à son arborescence de répertoires et fichiers, située dans ***/home/votre_login*** ? Si c'est non :
   - Où (à quel endroit) sont localisés/situés les fichiers de l'utilisateur authentifié ?
   - À quel endroit se situe actuellement l'utilisateur authentifié ?

c) Montage du répertoire associé au compte

L'étape précédente s'est limité à permettre l'authentification d'un utilisateur, sans lui permettre de naviguer dans son arborescence de répertoires et fichiers, située dans ***/home/votre_login.***

Pour rendre cela possible, il faut encore monter par *NFS* de ce répertoire ***/home/votre_login*** (localisé sur *electre)* sur le point de montage local sur votre machine : ***/home/votre_login***. Il faudra réaliser ce montage en appliquant la même méthode que celle utilisée au 6.2 pour le répertoire ***/usr/local***.

- Qu'elle est la commande ?
- Si la commande échoue, quel est le message ?
- En fait, sous couvert de quel utilisateur la commande a-t-elle été lancée ? Est-ce normal que le montage a échoué ?
- Quel utilisateur est autorisé à lancer une commande de montage ?
- Comme changer d'utilisateur ? Faire le changement et relancer la commande de montage, et vérifier à nouveau l'accès aux fichiers de l'utilisateur connecté.

6.5 Utilisation d'interface graphique (pour réaliser les configurations précédentes)

Les commandes utilisées jusqu'ici sont généralement celles utilisées par les administrateurs systèmes habitués des systèmes Linux. Afin de vulgariser l'utilisation de Linux, un gros effort a été fait sur la configuration du système, y compris pour sa partie réseau. Pour cela des interfaces graphiques sont disponibles.

1) Rebooter le système pour le réinitialiser puis utiliser l'interface graphique de configuration réseau accessible par le menu identifié à la section 5.1) question 3 (ou, dans le menu en bas à gauche, puis *préférences*, puis *connexion réseau*).

2) Est-ce l'option « activer le réseau » doit être active ou désactive pour effectuer le paramétrage ? Pourquoi ?

3) Quels sont les paramètres à configurer pour obtenir la même connectivité qu'avec les commandes « manuelles » ? Bien les noter, car cette phase de configuration réseau sera sûrement à répéter encore plusieurs fois dans la suite du TP.

   Précisions : la connexion est filaire, en IPV4, il ne faut pas utiliser un serveur DHCP (qui sert à une attribution automatique des adresses IP, sans choix possible de la part de l'utilisateur).

   Attention : s'il y a plusieurs cartes réseaux, choisir celle qui est réellement utilisée (voir section 5.2, question 2), en la sélectionnant dans l'onglet « adresse mac périphérique » et dans la liste déroulante proposée.

4) Que faut-il encore faire pour rendre opérationnel le paramétrage réalisé ? Attention : le temps nécessaire pour que paramétrage réalisé devienne opérationnel est un peu long.

6.6 Configuration de l'imprimant réseau

Il existe aussi un programme permettant de configurer l'accès à des imprimantes en réseau. Pour configurer l'imprimante *imp4*, choisissez dans la barre de menu en bas, à gauche, le menu *Knoppix*, puis *Printer configuration*. Dans la fenêtre qui s'ouvre, ajouter une nouvelle imprimante. Sélectionner un serveur du type *Network Printer (TCP)*, puis positionner l'adresse IP de l'imprimante. Sélectionner les drivers en fonction du type de l'imprimante, puis donnez-lui un nom.

7. Configuration réseau d'une machine linux : phase 3 – Services avancés et sécurité

Cette deuxième phase de travaux pratiques a pour objectif d'étudier les différents services, protocoles et applications pour le réseau disponible sur un système Unix ainsi que leur implication sur la sécurité. Nous verrons aussi les principes d'encapsulation de protocoles au travers de captures de trafic.

Une machine particulière sera disponible, avec certains services déjà opérationnels pour vous permettre de faire certains tests à partir de votre propre machine.

7.1 Changement des mots de passe et création de comptes utilisateur

La première des règles de sécurité consiste à protéger les comptes crées par défaut (*root*, *knoppix*) par de nouveaux mots de passe. Pour cela vous utiliserez la commande **passwd** depuis une console en tant qu'utilisateur *root (rappel : utiliser la commande* **su***, en revérifiant les alias)* : **passwd** *utilisateur*

1) Changer le mot de passe de chacun de ces comptes … après avoir réfléchi à la « politique » de définition des mots de passe à utiliser dans CE TP !

Maintenant que ces comptes sont protégés, créer 2 nouveaux comptes utilisateurs sur la machine. Pour cela, utiliser la commande **adduser** dans une console *root.*
Exemple : **adduser** *luc*

2) Créer 2 utilisateurs puis changer leur mot de passe … après avoir réfléchi à la « politique » de définition des noms d'utilisateurs et des mots de passe à utiliser dans CE TP ! Hormis ces noms et mots de passe, les autres informations demandées sont sans importance.

   Remarque : dans le texte de TP, on utilisera de manière générique les dénominations *user1* et *user2*.

3) Comment vérifier et tester la création des deux comptes utilisateur ?
   - Se loguer en tant que *user1.* Est-ce que le mot de passe est demandé ? Est-ce normal ? Peut-on attendre le répertoire ***/home/user1 ?*** Et répertoire ***/home/user2 ?***
   - Essayer changer d'utilisateur (soit vers root, soit vers user2) .

4) Est-ce que cela a fonctionné comme prévu ? A partir de ce que vous avez constaté, d'où vient le problème ? Comment contourner le problème ?

5) Contourner le problème et recommencer l'opération et vérifier à nouveau la bonne création des utilisateurs.

7.2 Configurer votre machine en réseau

Commencez à configurer votre machine en réseau, soit en utilisant l'interface graphique, soit en utilisant les commandes en lignes étudiées à la première séance (alors, ne pas configurer les services NFS/NIS, etc.)

1) Comment avez-vous testé la connectivité ?

7.3 Connexion/travail à distance (rappel : être en root) : service SSH

Internet permet de réaliser un grand nombre d'opérations à distance, comme le transfert de fichiers ou l'administration de serveurs. Le protocole *SSH* (Secure Shell), mis au point en 1995, remplace les protocoles *telnet*, *rlogin*, *rsh* qui ne sont pas sécurisés. *SSH* est un protocole qui permet à un client (un utilisateur ou bien même une machine « locale ») d'ouvrir une session interactive sur une machine distante (serveur) pour y envoyer des commandes ou y manipuler des fichiers, le tout, de manière sécurisée.

Le protocole *ssh* fonctionne en mode client/serveur, c'est à dire qu'un **programme** *serveur* doit s'exécuter sur une machine « distante » (la machine est, elle aussi, appelée « serveur » par abus de langage), pour exécuter les requêtes émises par le client (à partir d'une machine « locale »).

Le principe est de **lancer un tel serveur** par l'intermédiaire du démon approprié, **`ssh`** (disponible dans le répertoire de base pour les démons, ***/etc/init.d***), puis, d'y **accéder à partir d'une autre machine** du réseau local en utilisant le compte utilisateur qui a été créé à cet effet, et la commande à utiliser est :

ssh *@IP* **–l** *utilisateur* (selon le cas, penser à faire précéder de *./*).

1) A quoi sert l'option ***–l*** de la commande **ssh** ?

2) Sur le serveur : sur quelle machine faut-il être pour lancer le serveur ? Sous quel compte faut-il être pour le lancer ? Quelle est la commande exacte pour lancer le serveur ssh (et de manière plus générale, pour lancer ou arrêter un démon) ?

3) Sur quelle machine doit être le compte utilisateur via lequel on veut se connecter au serveur ? Quelles sont les différentes commandes possibles qui permettent de déduire qu'on est sur la machine distante ?

4) Quel est l'avantage d'un tel contrôle à distance ? Le service (protocole) *ssh* permet-il aussi de travailler sur la machine locale, ou seulement sur la machine distante ?

7.4 Échanges de fichiers : service FTP

FTP pour *File Transfer Protocol* (protocole de transfert de fichiers) est un protocole dédié/spécifique à « l'échange » de fichiers sur un réseau TCP/IP : charger/télécharger (envoyer/récupérer) des fichiers d'un ordinateur vers/depuis un autre ordinateur, mais aussi administrer un site web ou encore supprimer ou modifier des fichiers. Bien que les bases de ce protocole furent jetées en 1971 dans la RFC 141, il reste toujours d'actualité grâce à ses évolutions régulières.

FTP obéit lui aussi à un modèle client-serveur. En pratique, le « serveur FTP » est un logiciel qui rend public une arborescence de fichiers similaire à un système de fichiers Unix. L'accès à un serveur FTP, se fait via un logiciel « client FTP », utilisable soit directement dans un terminal à l'aide de commandes en ligne, soit via une interface graphique.

a) Configuration du serveur FTP

Le serveur FTP fonctionne par l'intermédiaire d'un démon (`ftpd`). Mais cette fois (comme pour certains autres services, tels que *telnet, finger, etc)* pour des raisons de sécurité, il faut en plus disposer d'un contrôle des accès à ces services. C'est pourquoi le démon approprié n'est pas directement accessible depuis */etc/init.d*. Plus précisément, pour gérer ce contrôle d'accès, un outil (*TCP/WRAPPER)* a été mis en place afin de ne pas avoir à changer les sources des démons, tout en permettant de restreindre et de tracer les accès de certains services en fournissant l'origine de la requête (adresse de la machine), et donc de réduire ainsi l'accès à des services qui permettraient les tentatives de piratage.
Le principe est simple : au lieu de lancer directement le démon souhaité (par exemple *in.telnetd* pour le service telnet), TCP/WRAPPER active un programme (`tcpd`) qui se charge de lancer le serveur voulu, après avoir vérifié que le client est bien autorisé à se connecter au serveur.

Pour cela, le fichier ***/etc/inetd.conf*** doit contenir « l'entrée » *tcpd* (voir ci-après) pour permettre l'activation du service *ftp* :

ftp stream tcp nowait root /usr/sbin/tcpd /usr/sbin/in.ftpd[3]

1) Consulter le contenu de ce fichier ? Conclusion ?

Ensuite, *tcpd* se base sur 2 fichiers ***/etc/hosts.allow*** et ***/etc/hosts.deny*** qui autorisent ou interdisent l'accès à certains démons. Les « entrées » dans ces fichiers doivent avoir la forme ci-après (généralement, il faut écrire la ligne en entier, et avant la dernière ligne du fichier) :

in.ftpd : ALL : ALLOW ou, symboliquement **`liste_démons:hôtes_clients [:commande`**[4]**`]`**

2) Configurer (si c'est nécessaire) avec précaution ces 2 fichiers.

3) Lancer le démon approprié pour activer le service *ftp* : dans ce TP, le démon est ***inetutils-inetd***. Qu'est-ce qui s'affiche ?

4) Créer un fichier (le contenu n'a pas d'importance) dans le compte knoppix du serveur FTP.

b) Utilisation d'un client FTP : client en ligne de commande ou client graphique

L'utilisation d'un client *ftp* (pour accéder au serveur *ftp*) peut se faire en ligne de commande : **`ftp`** *@IP*. Il faut alors encore donner le compte sur lequel on veut aller. Et des commandes en lignes simples permettent alors de charger (respectivement télécharger) un fichier : ***put nom fichier*** (respectivement ***get***).
De plus, certaines commandes classiques d'Unix sont utilisables pour se déplacer dans l'arborescence, créer des répertoires, etc. : **`ls`**, **`cd`**, **`mkdir`**, etc. Et surtout, il est possible d'exécuter ces commandes :

- soit sur la machine distante (serveur),
- soit sur la machine locale (client), mais dans ce dernier cas, il faut faire précéder la commande par le symbole '!'. Par exemple, **`!ls`** provoque l'affichage des fichiers disponibles sur la machine locale.
- Et dans tous les cas, par défaut, c'est au répertoire courant que cela s'applique.

[3] *C'est ici qu'on peut choisir un autre logiciel serveur FTP, car il en existe plusieurs, plus ou moins puissants.*
[4] man hosts.allow pour plus d'informations sur les commandes.

Mais on peut aussi utiliser un client FTP graphique en utilisant une *URL* (*Uniform Ressource Locator)* du type ***ftp://utilisateur@serveur*** (le serveur peut être décrit par son adresse IP ou son nom réseau).

1) Sur quelle machine doit être le compte à utiliser : client ou serveur ?

2) À l'aide de commandes en lignes, se connecter sur un serveur FTP voisin (créé sur une des machines du réseau local) et récupérez un fichier préalablement crée cette « machine serveur ». Qu'elle est la séquence de commandes à utiliser pour télécharger ce fichier.

3) Recommencer ces manipulations en utilisant cette fois une interface graphique.

4) A quoi servent les commandes **`open`**, **`close`**, **`bye`**, **`quit`**, proposées par le service FTP ?

5) Le service *ftp* permet-il des manipulations à fois sur la machine locale et sur la machine distante ?

6) Penser à quitter les sessions ftp à la fin de cet exercice. Pourquoi ?

7.5 Créer et configurer un serveur WEB (rappel : être en root)

Un serveur « http » est un logiciel qui répond (sert) des requêtes respectant le protocole de communication client/serveur *HyperText Transfer Protocol* (*HTTP*), qui a été développé pour le World Wide Web.

Un ordinateur sur lequel tourne un serveur HTTP est appelé serveur Web. Le terme « serveur Web » peut aussi désigner le « serveur http » (le logiciel) lui-même. Les deux termes sont utilisés pour le logiciel car le protocole HTTP a été développé pour le Web, et les pages Web sont en pratique toujours servies/accessibles avec ce protocole. D'autres ressources du Web comme les fichiers à télécharger ou les flux audio ou vidéo sont en revanche fréquemment servis avec d'autres protocoles (*ftp, rtp, etc.*).

a) Création de pages web

Avant de configurer un « serveur Web », il faut d'abord commencer à en rédiger le contenu. Par défaut, le serveur *http* est configuré pour stocker des pages *html* (pages écrites en langage *html*) dans le répertoire ***/var/www***.

Pour cela, utiliser un éditeur de texte ou un logiciel *wysiwyg* (ex. : *LibreOffice*) pour créer une ou plusieurs pages web à placer dans ce répertoire. Attention :

- les pages (fichiers) doivent être installées dans le répertoire en question par l'utilisateur *root*
- les fichiers créés doivent porter l'extension *.html*,
- au moins une page doit être placée dans fichier nommé *index.html*, qui est la page ouverte par défaut par le serveur.
- Les fichiers doivent avoir des droits en lectures … pour être accessibles en lecture …

1) Créer 3 pages : une uniquement avec du texte, une avec aussi des images, et une avec aussi des liens internet, en respectant toutes les contraintes décrites ci-dessus.

b) Configuration du serveur Web

Le serveur *http* disponible sur le système *knoppix* est un démon nommé `thttpd`.

1) Vérifier le contenu des fichiers de configurations (de ce démon) situés dans ***/etc/thttpd/*** et lancer le démon. Quelle est la ligne commande utilisée ? Qu'est-ce qui s'affiche ?

c) Utilisation du serveur Web

1) Utiliser navigateur web pour vérifier l'accès aux serveurs web distants ainsi configurés. Quelle URL a été utilisée pour effectuer ce test ?

d) Traces d'accès

Traditionnellement, tous les systèmes Unix maintiennent un ensemble de traces révélant l'utilisation récente d'un système. Ceci peut être utile pour obtenir des messages d'erreurs détaillant par exemple le mauvais fonctionnement d'un démon, ou bien permettant de vérifier quels ont été les accès au système. Sous *linux*, ces fichiers se trouvent dans le répertoire ***/var/log***. Le fichier de trace spécialement dédié au serveur *http* est ***/var/log/thttpd.log***

1) Qui a accédé à votre serveur Web ?

7.6 Configuration du Firewall

Le pare-feu (*Firewall*) est aujourd'hui considéré comme la pierre angulaire de la sécurité d'un réseau informatique. Il permet d'appliquer une politique d'accès aux ressources réseau (serveurs). Il regarde chaque paquet indépendamment des autres et le compare à une *liste de règles* préconfigurées pour déterminer si le paquet doit être accepté sur la machine qu'il protège. La configuration de ces dispositifs est souvent complexe, mais grâce à l'arrivée de l'ADSL qui a popularisé l'utilisation de ces logiciels, de nombreuses offres simples d'accès sont apparues.

Knoppix offre un *firewall* simple qui offre plusieurs niveaux de configuration : *easy*, *medium*, *expert*. Le paramétrage du firewall peut se faire :

- soit directement dans un terminal à l'aide de commandes en ligne,
- soit via une interface graphique, mais dont la manipulation n'est pas toujours très facile à appréhender, en particulier en ce qui concerne la validation effective du paramétrage choisi.

1) Lancer le firewall (en ligne de commande, simplement avec la commande `firewall`), soit en trouvant dans les menus comment le lancer.

2) Activer le firewall en mode débutant (*easy*) et tester si votre serveur web est toujours accessible ? Pourquoi n'est-il pas accessible ?

3) Activer maintenant le firewall en mode *medium* et configurez-le afin qu'il autorise les connexions *http* à passer au travers. Pour cela, il faut « ouvrir des ports », ce qui signifie autoriser les connexions de niveau transport à utiliser ces ports (adresses de niveau transport). « Redémarrer » le firewall. Tester si le serveur web est à nouveau accessible.

4) Une fois que cela fonctionne, configurez le firewall afin qu'il autorise le *ping*, puis les services que vous avez mis en place dans les questions précédentes (*ssh*, *ftp*, *etc*.). Quels ports avez-vous du ouvrir ?

Comme vous le voyez, les règles que vous appliquez sont très rudimentaires, du type tout ou rien, et ne permettent que d'autoriser ou interdire totalement un service. En mode *expert* il est possible de raffiner ces règles par l'ajout de règles ***iptables*** dans le fichier de configuration du *firewall*. Ces règles sont assez complexes[(5)] , mais en première approximation voici le type de règle qui permet de filtrer tous les paquets provenant d'une machine donnée :

ex. pour tout bloquer *: iptables -I INPUT 1 -s @IP - j DROP*

ex. pour autoriser tcp *: iptables -I INPUT -p tcp --dport 80 -j ACCEPT*

1) Comment filtrer explicitement les requêtes WEB effectuées sur votre serveur et provenant de votre voisin de droite, considéré comme malveillant, sans altérer le service pour votre voisin de gauche ?

7.7 Utilisation d'un sniffer de paquets : Wireshark

Wireshark est un logiciel libre d'analyse de protocole, ou « packet sniffer », utilisé dans le dépannage et l'analyse, de réseaux informatiques, le développement de protocoles, et l'éducation. Il crée une copie (« *capture* ») de chaque paquet circulant sur le réseau associé à une interface réseau donnée et affiche sous forme lisible, c'est à dire en traduisant les encapsulations protocolaires diverses, le contenu de ces paquets.

Utiliser la commande `whireshark` en console *root* pour démarrer le logiciel. Pour démarrer une analyse, utiliser la menue *capture*, sélectionnez *l'interface* sur laquelle vous souhaitez réaliser la capture, puis ***start***. Ainsi, tous les paquets « vus » par l'interface apparaîtront dans l'afficheur graphique. Pour une meilleure

(5) `man iptables` *fournira des informations complémentaires*

lecture, il est possible de filtrer ces paquets, pour ne garder par exemple que ceux concernant votre machine. Pour cela, dans la zone de saisie *filter* introduisez : *« ip.host = = "@IP" »*.

Il est de plus possible de suivre un échange relatif à un protocole, comme par exemple, une session *FTP*, en sélectionnant un des paquets de la session et à l'aide du bouton droit de la souris choisir « *follow stream* ». Il existe bien sûr de nombreuses possibilités de configuration des filtres.

Déclenchez une capture, puis immédiatement après une session *FTP* avec une machine distante. Filtrez afin que seule la session *FTP* soit visible sur l'écran.

1) Quels sont les ports mis en jeu dans une session FTP ? Combien de connexions sont actives ? Que déduisez-vous de cette analyse ? Conclure.
2) Mêmes questions pour http.

PARTIE INTERFACE DE PROGRAMMATION TCP/IP

Vous n'utiliserez plus Knoppix, mais votre compte habituel dans l'environnement Linux habituel.
Cette partie a pour objectif de découvrir les interfaces de programmation réseau disponibles sous Unix et Windows en langage C. En vous basant sur des exemples, vous réaliserez différents programmes utilisant le protocole TCP/IP.

8. Téléchargement des exemples

Télécharger les exemples disponibles sur : http://www.laas.fr/~berthou/UPS/IUP_SI/Exemples/

1) Comment utiliser ces programmes ? Que font-ils ?

9. Utilisation du service DNS dans un programme C

Dans le programme précédent, on remarque que le *client* se connecte au serveur grâce à son adresse IP. Comme étudié précédemment, il serait plus simple d'utiliser le service de résolution de nom afin d'utiliser une adresse au format alphabétique. Ceci est possible grâce à la fonction *gethostbyname* disponible dans les librairies C. Sa syntaxe est la suivante :

```
struct hostent *gesthostbyname (char *hostname)
```

Cette fonction prend en paramètre une adresse IP sous forme de chaîne de caractères et retourne un pointeur sur une structure *hostent* :

```
struct hostent {
  char *h_name;          /*NOM DE LA MACHINE*/
  char **h_aliases;      /*liste d'ALIAS*/
  int h_addrtype;        /*PF_INET*/
  int h_lenght;          /*4 octets*/
  char **h_addr_list;    /*liste d'adresses Internet*/
};
```

2) Où ajouter les lignes suivantes ?

```
struct hostent *he ;
if (( he = gethostbyname ("salsa.laas.fr"))==NULL)
{  perror("gethostbyname");  exit(2); }
memcpy(&sock_host.sin_addr,he->h_addr_list[0],sizeof(sock_host.sin_addr))
;
```

Le serveur est capable de déterminer l'adresse des clients connectés sur son port grâce au paramètre *serveur_adr* passé à la primitive *accept*. Toutefois, avant d'afficher cette adresse, vous devez la convertir du format illisible, dit *« réseau »*, en une chaîne de caractères dit format *« ascii »* que vous pourrez afficher avec un simple *printf*. Pour cela utilisez la fonction :

```
char *inet_ntoa(struct in_addr in) ;
```

La structure suivante possède un champ de type *in_addr*.

```
struct sockaddr_in {
sa_family_t     sin_family;   /* address family: AF_INET */
 in_port_t      sin_port;     /* port in network byte order */
struct in_addr sin_addr;     /* internet address */
};
```

3) Qu'elle ligne de code faut-il ajouter à votre programme pour afficher l'adresse du client ?

10. Consultation de Services Internet

a) Daytime

Daytime est un petit service activable sur une machine Unix qui fournit l'heure de la machine à un client distant. Ce programme écoute sur un port spécifié dans */etc/services,* et lors d'une demande de connexion *TCP* ou *UDP,* il retourne automatiquement l'heure sous forme alphanumérique. Bien sûr, ce programme est géré par *inetd*.

1) Quel est le port associé au service *daytime* ?
2) Quel fichier faut-il modifier pour que le service *daytime* soit actif sur votre machine ? Quel démon faut-il activer ?

Un premier test de ce programme peut se faire depuis un terminal par l'intermédiaire du programme *telnet* dont la syntaxe est la suivante : **`telnet @IP port`**

3) Quelle commande faut-il taper pour interroger le service ?

En se basant sur le programme client précédent, proposer une modification qui permette d'obtenir l'heure d'une machine distante sans utiliser le programme *telnet*.

4) Comment faut-il modifier l'algorithme de votre programme pour que cela fonctionne ?

b) Mini Client Web

Le protocole HTTP (*HyperText Transfer Protocol*) est le protocole le plus utilisé sur Internet depuis 1990. Le but du protocole HTTP est de permettre un transfert de fichiers (essentiellement au format HTML) localisés grâce à une chaîne de caractères appelée *URL* entre un *navigateur* (le *client*) et un *serveur* Web. La communication entre le navigateur et le serveur se fait en deux temps:

- Le navigateur effectue une **requête HTTP** sur le port **80**
- Le serveur traite la requête puis envoie une **réponse http**

Voici donc un exemple de requête HTTP [6] :

```
GET /index.html HTTP/1.1
Host: www.laas.fr
```

Une réponse HTTP est un ensemble de lignes envoyées au navigateur par le serveur.

1) Comment faut-il modifier le programme client pour qu'il puisse interroger un serveur web ? Qu'elle est la réponse reçue ?

c) Application d'interrogation de serveurs de données

Il est possible d'utiliser ce type de programme pour collecter automatiquement et régulièrement (par exemple tous les matins) des informations mises à jour sur des sites Web publiques. On peut par exemple, afficher la hauteur de neige de votre station de ski favorite grâce à « *www.skiinfo.fr* ». Indice : utiliser la fonction *strstr* qui recherche dans une chaîne de caractère une occurrence donnée. La syntaxe de cette fonction est (et `man strstr` pour plus d'informations) :

```
char *strstr (const char *meule_de_foin, const char *aiguille)[7]
```

1) Quels serveurs autres que *skiinfo* renvoient des résultats facilement exploitables ?

d) Robot d'indexation pour le Web

Les robots d'indexation sont les programmes utilisés par les éditeurs de moteurs de recherche afin d'explorer le Web, en suivant les liens qui relient les pages entre elles et en localisant les pages nouvellement créées. En cela, ils constituent une classe d'agents intelligents. En indexant les mots de tout ou partie de chaque page rencontrée, ils permettent aux recherches effectués par les internautes d'être extrêmement rapides.

2) Comment programmer un tel robot d'exploration (sans se préoccuper des problèmes d'indexation) ?

---

[6] *Attention, la requête doit être terminée par 2 caractères de contrôle '\n'*
[7] *man strstr pour davantage d'informations*