***NOLLE***

***Damien***

***L3 – Informatique***

**SR – Devoir 2 :**

|  |  |
| --- | --- |
| ***Note :*** | ***Observation :*** |
| ***/20*** |  |

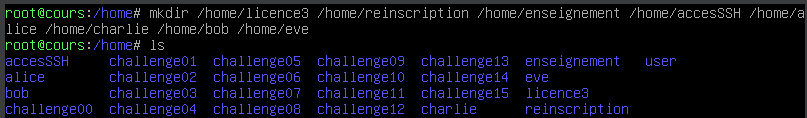
Exercice 1)

Q1 :

Etape 0 : Accès au compte « root » avec la commande « su » et le mot de passe « EadSr2017 » :

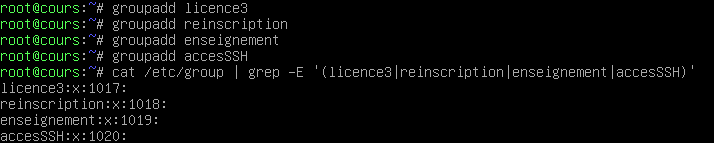


Etape 1 : Création des répertoires « licence3 », « reinscription », « enseignement », « accesSSH » (correspondant aux groupes), « alice », « charlie », « bob » et « eve » (correspondant au utilisateur avec la commande « mkdir » :

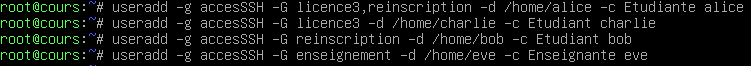


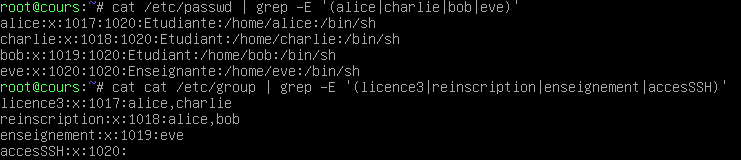
*Pour l’instant, tout les répertoires précédemment créés ont pour propriétaire « root » et pour groupe, le groupe principal du super-utilisateur nommé « root ».*

Etape 2 : Création des différents groupes (licence3, reinscription, enseignement et accesSSH) avec la commande « groupadd » :

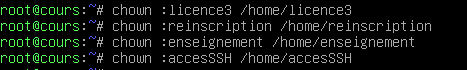


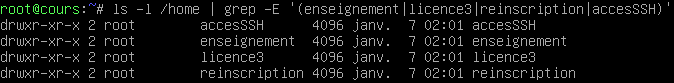
Etape 3 : Création des différents utilisateurs (alice, charlie, bob et eve) avec la commande « useradd », en sachant que tous les utilisateurs auront comme groupe principal, le groupe accesSSH et que alice appartient au groupe licence3 et reinscription, charlie au groupe licence3, bob au groupe reinscription et eve au groupe enseignement :





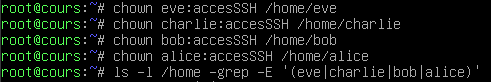
Etape 4 : Changement des groupes sur les répertoires « licence3 », « reinscription », « enseignement » et « accesSSH » avec les groupes correspondants via la commande « chown » :

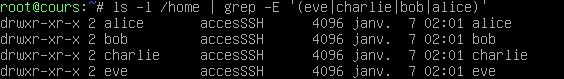




*Seulement le groupe a été changé, le propriétaire restera donc « root ».*

Etape 5 : Changement des groupes et des utilisateurs propriétaire pour chaque répertoire personnel en fonction de l’utilisateur correspondant avec la commande « chown » :





*On attribue le groupe principal de tous les utilisateurs (à savoir « accesSSH ») de manière à attribuer des droits pour tous les utilisateurs du serveur de fichier, afin que seul l’utilisateur propriétaire n’y ai accès. En réalité, le groupe n’a pas d’importance ici car, quel que soit le groupe d’appartenance, seul le propriétaire y aura accès. Donc les droits pour le groupe seront les mêmes que pour les autres.*

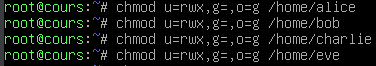
Etape 6 : Changement des droits des répertoires personnels pour que seul l’utilisateur propriétaire ai un accès :

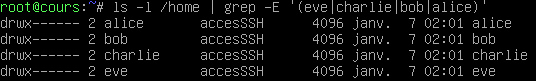
Pour cela, il faut que les propriétaires puissent afficher (r), modifier (w) et explorer le contenu du répertoire.

Il faut alors attribuer les droits :

* Le propriétaire (u) : « rwx » ou 22 \* 1 + 21 \*1 + 20 \* 1 = 4 + 2 + 1 = 7 en octal.
* Le groupe (g) : « » ou 22 \* 0 + 21 \*0 + 20 \* 0 = 0 + 0 + 0 = 0 en octal.
* Les autres (o) : « » ou 22 \* 0 + 21 \*0 + 20 \* 0 = 0 + 0 + 0 = 0 en octal.

On attribut ensuite les droits avec la commande « chmod », soit en écriture octal, soit en écriture rwx :





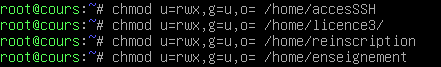
Etape 7 : Changement des droits des répertoires de chaque groupe pour que seul le groupe propriétaire et l’utilisateur propriétaire (à savoir « root ») ai un accès :

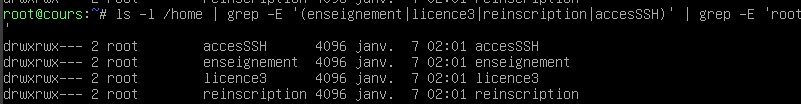
Pour cela, il faut que le propriétaire et les membres du groupe puissent afficher (r), modifier (w) et explorer le contenu du répertoire.

Il faut alors attribuer les droits :

* Le propriétaire (u) : « rwx » ou 22 \* 1 + 21 \*1 + 20 \* 1 = 4 + 2 + 1 = 7 en octal.
* Le groupe (g) : « rwx » ou 22 \* 1 + 21 \*1 + 20 \* 1 = 4 + 2 + 1 = 7 en octal.
* Les autres (o) : « » ou 22 \* 0 + 21 \*0 + 20 \* 0 = 0 + 0 + 0 = 0 en octal.

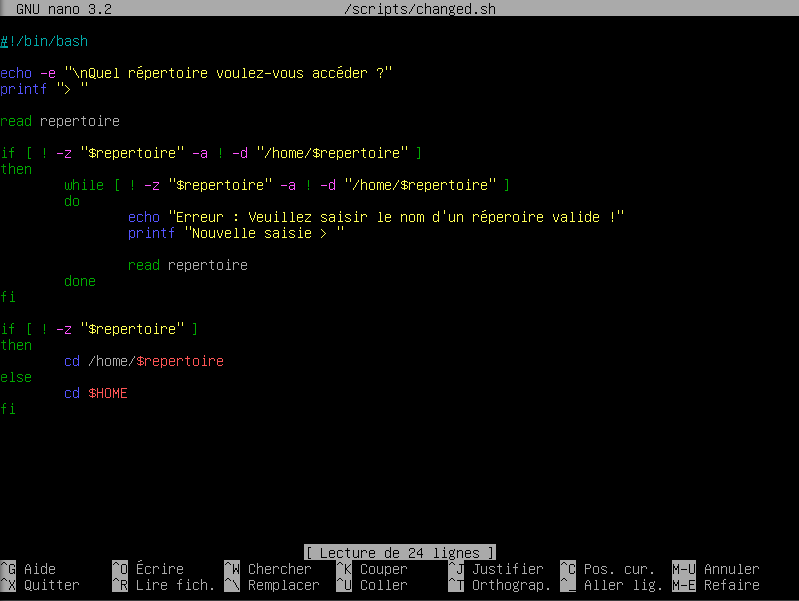
On attribut ensuite les droits avec la commande « chmod », soit en écriture octal, soit en écriture rwx :





Q2 :

Script permettant de demander à l’utilisateur un répertoire dans « home » vers lequel se déplacer, si rien n’est saisi, il reste dans son répertoire personnel :



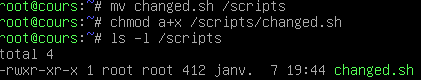
Pour que ce script soit lancé à la connexion d’un utilisateur, il faut soit écrire le script dans le fichier « /etc/profile », soit permettre l’exécution du script via un fichier de commande depuis le fichier « /etc/profile ».

Prenons ce dernier cas, on va venir mettre le script bash dans un répertoire ayant les permissions pour que les autres utilisateurs puissent lire et explorer le contenu de celui-ci :

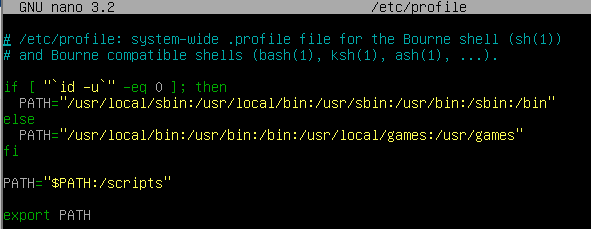


*On remarque que les bonnes permissions sont déjà accordées à la création du répertoire, on n’a donc pas besoin de les modifier.*

On déplace le script dans le répertoire précédemment créé et on ajoute la permission d’exécution pour tout le monde :



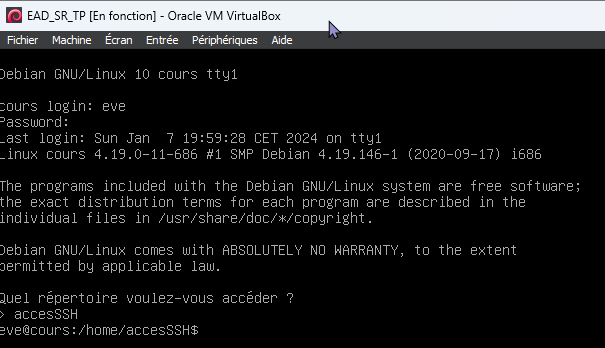
On vient modifier le fichier « /etc/profile » de manière à modifier la variable d’environnement PATH pour ajouter le répertoire où se trouve le script :



Puis dans ce même fichier, on vient ajouter la commande pour exécuter le script (à savoir le nom du fichier, soit : « changed.sh ») avec la commande « . » afin d’exécuter le script dans le processus shell courant pour permettre le déplacement avec la commande « cd » :



On redémarre la machine avec la commande « reboot » et on se connecte avec un utilisateur, par exemple, « eve » :



Q3 :

Les unmask permettent de changer les permissions par défaut appliquées lors de la création d’un nouveau fichier ou répertoire.

Pour déterminer les permissions appliquées, il faut réaliser les calculs suivants en fonction de si c’est :

* Un fichier :

1. Complément à 1.
2. On retire la permissions d’exécution à 1 en le mettant à 0.

* Un répertoire :

1. Complément à 1.

Prenons les 3 différents umask que l’on peut positionner pour les utilisateurs :

* 044 :

044 en octal donne : 000 100 100.

1. Pour un fichier :

Complément à 1 : 111 011 011.

On retire les permission d’exécution : 110 010 010.

1. Pour un répertoire :

Complément à 1 : 111 011 011.

Lorsqu’un utilisateur créera un nouveau fichier, il pourra le lire et le modifier. Alors que son groupe et les autres utilisateurs pourront uniquement le modifier. Lorsqu’il créera un nouveau répertoire, il pourra afficher, modifier et explorer son contenu. Alors que son groupe et les autres utilisateurs pourront modifier et explorer son contenu.

* 066 :

066 en octal donne : 000 110 110

1. Pour un fichier :

Complément à 1 : 111 001 001.

On retire les permission d’exécution : 110 000 000.

1. Pour un répertoire :

Complément à 1 : 111 001 001.

Lorsqu’un utilisateur créera un nouveau fichier, il pourra le lire et le modifier. Alors que son groupe et les autres utilisateurs n’auront aucune permission dessus et donc n’auront aucun accès. Lorsqu’il créera un nouveau répertoire, il pourra afficher, modifier et explorer son contenu. Alors que son groupe et les autres utilisateurs pourront uniquement explorer son contenu.

* 077 :

077 en octal donne : 000 111 111

1. Pour un fichier :

Complément à 1 : 111 000 000.

On retire les permission d’exécution : 110 000 000.

1. Pour un répertoire :

Complément à 1 : 111 000 000.

Lorsqu’un utilisateur créera un nouveau fichier, il pourra le lire et le modifier. Alors que son groupe et les autres utilisateurs n’auront aucune permission dessus et donc n’auront aucun accès. Lorsqu’il créera un nouveau répertoire, il pourra afficher, modifier et explorer son contenu. Alors que son groupe et les autres utilisateurs n’auront aucune permission dessus et donc n’auront aucun accès.

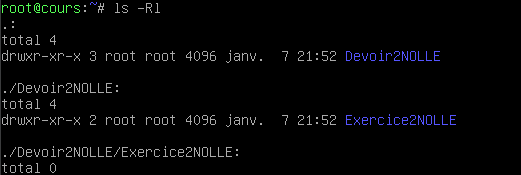
Dans les trois cas, il s’agit de restreindre l’accès aux fichiers et aux répertoires par défaut pour le partage. Dans le premier cas (044), les fichiers pourront être modifié et le contenu des répertoires pourront être explorer et modifier par les autres utilisateurs et ceux du groupe. Dans le second cas (066), ils ne pourront qu’explorer les répertoires mais ne pourront pas modifier le contenu ni l’afficher et n’auront aucuns accès aux fichiers. Dans le dernier cas (077), les fichiers et les répertoires ne seront accessibles que par l’utilisateur propriétaire.

Exercice 2)

Q1 :

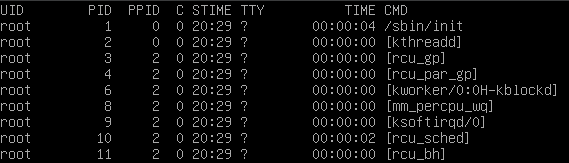
Création du répertoire de travail, soit du répertoire « Devoir2NOLLE » et dans ce dernier, du répertoire « Exercice2NOLLE ». Le tout dans notre répertoire d’accueil avec la commande « mkdir » :



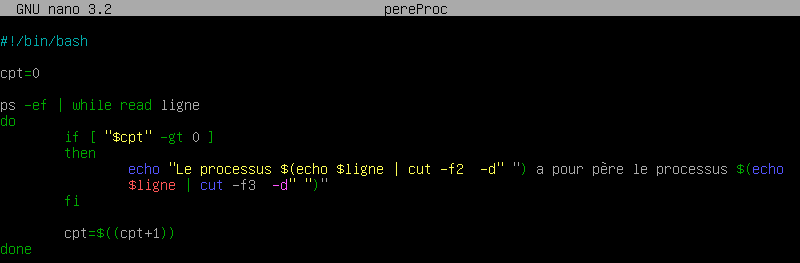


Q2 :

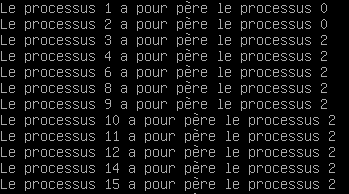
1. Pour afficher le PID et le PPID de chaque processus actif sur la machine, on peut utiliser la commande « ps -ef » :



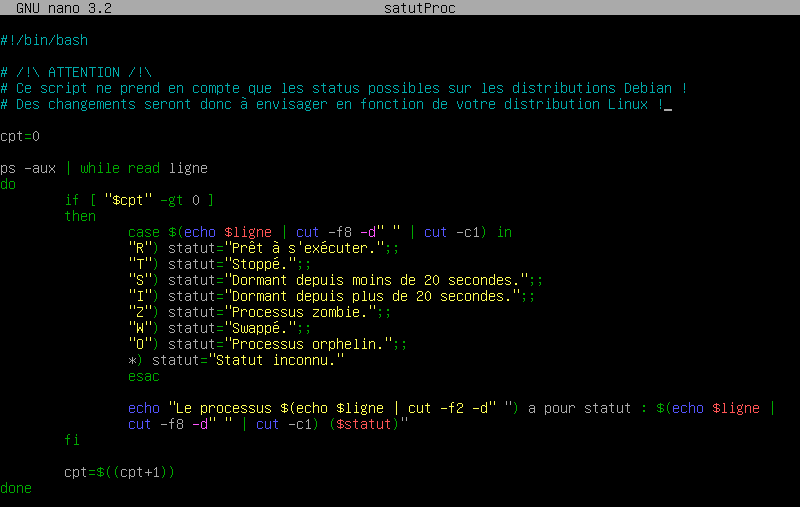
1. Script « pereProc » permettant d’afficher pour chaque processus actif sur la machine, le processus père sous la forme « Le processus [PID] à pour père le processus [PPID] » :



Résultat :



Q3 : Script « statutProc » permettant d’afficher pour chaque processus actif sur la machine, son statut et à quoi il correspond :

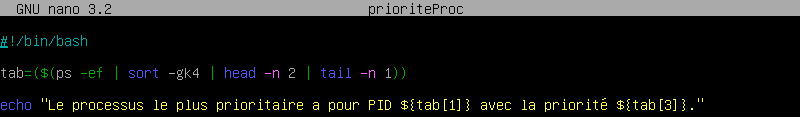


Résultat :



Q4 :

1. Script permettant d’afficher le PID du processus le plus prioritaire avec sa priorité :



Résultat :



1. Pour afficher le PID du processus le plus prioritaire et sa priorité, on peut utiliser la commande « ps –efo "%p %n" | sort -gk2 | head -n 2 » :



Exercice 3)

Q1 : On sait qu’il y aura 3 sous-réseaux : SR1 (la direction et le marketing), SR2 (l’unité de production) et SR3 (le bureau d’études). Chaque sous-réseaux contiendra un routeur.

* Pour SR1 : Il y aura 31 postes. Donc il faut 31 adresses IP pour ces postes + 1 adresse IP pour le routeur + 2 adresses IP réservés (correspondant à l’adresse réseau et de broadcast). Ce qui fait au total : 31 + 1 + 2 = 34 adresses IP qui seront utilisées.
* Pour SR2 : Il y aura 247 postes. Donc il faut 247 adresses IP + 1 adresse IP pour le routeur + 2 adresses IP réservés. Ce qui fait au total : 247 + 1 + 2 = 250 adresses IP qui seront utilisées.
* SR3 : Il y aura 450 postes. Donc il faut 450 adresses IP + 1 adresse IP pour le routeur + 2 adresses IP réservés. Ce qui fait au total : 450 + 1 + 2 = 453 adresses IP qui seront utilisées.

On sait que la taille d’un sous-réseau correspond au nombre d’adresses IP au total - 2 = nombre d’adresses IP disponibles.

Le nombre d’adresses IP total du sous-réseaux correspond à : 2nombre de 0 du masque.

Par exemple : Si dans le masque nous avons que deux 0, on aura alors 22 = 4 adresses IP au total dans le sous-réseaux. Sois 4 – 2 = 2 adresses IP disponibles.

* Pour SR1 : 26 = 64 > 34 adresses IP nécessaires mais 25 = 32 < 34 adresses IP nécessaires. On retient donc 26 car sinon on n’aura pas assez d’adresses IP. Le sous-réseau aura donc une taille de 64 adresses IP au total pour 62 adresses IP disponibles pour les postes et le routeur.
* Pour SR2 : 28 = 256 > 250 adresses IP nécessaires car 27 = 128 < 250. Le sous-réseau aura donc une taille de 256 adresses IP au total pour 253 adresses IP disponibles pour les postes et le routeur.
* Pour SR3 : 29 = 512 > 453 car 28 = 256 < 453. Le sous-réseau aura donc une taille de 512 adresses IP au total pour 510 adresses IP disponibles pour les postes et le routeur.

Q2 : Le réseau 172.18.0.0 est de classe B car 128 < 172 < 191, son masque par défaut est donc 255.255.0.0.

Le réseau 10.0.0.0 est de classe A car 0 < 10 < 127, son masque par défaut est donc 255.0.0.0.

Le réseau 192.168.4.0 est de classe C car 192 = 192 < 223, son masque par défaut est donc 255.255.255.0.

Puisqu’il faut découper le réseau de base en 3 sous-réseaux, il faut donc déjà que ce réseau puisse accueillir les 64 + 256 + 512 = 832 adresses IP nécessaires pour se découpage.

Le premier réseau de base (Classe B) :

Le masque 255.255.0.0 donne : 11111111 11111111 00000000 00000000 en base 2. On sait que 2nombre de 0 du masque correspond au nombre d’adresse IP total du réseau, donc : 216 = 65536 > 832 adresse IP nécessaires.

Le deuxième réseau de base (Classe A) :

Le masque 255.0.0.0 donne : 11111111 00000000 00000000 00000000 en base 2. On sait que 2nombre de 0 du masque correspond au nombre d’adresse IP total du réseau, donc : 224 = 16777216 > 832 adresse IP nécessaires.

Le troisième réseau de base (Classe C) :

Le masque 255.255.255.0 donne : 11111111 11111111 11111111 00000000 en base 2. On sait que 2nombre de 0 du masque correspond au nombre d’adresse IP total du réseau, donc : 28 = 256 < 832 adresse IP nécessaires.

Le deuxième réseau de base (Classe A) à un nombre d’adresses IP total trop supérieur au nombre d’adresses nécessaires. A l’inverse, le troisième réseau de base (Classe C) à un nombre d’adresses IP total trop inférieur au nombre d’adresses nécessaires.

Le réseau retenu sera donc le premier réseau de base (Classe B), soit 172.18.0.0 /16.

Q3 :

Le masque du réseau est 255.255.0.0 soit 11111111 11111111 00000000 00000000 avec 172.18.0.0 comme adresse réseau.

* SR1 :

Puisque 26 = 64 et que dans le masque de base on a : 16 bits à 0. Il faut donc ajouter 16 – 6 = 10 bits à 1 pour obtenir le masque de sous-réseau : 11111111 11111111 11111111 11000000, soit 255.255.255.(27 + 26 = 128 + 64 = 192) en décimal pointé, soit /26 en notation CIDR (qui correspond au nombre de bits à 1 du masque).

En partant du début : 172.18.0.0 correspond à l’adresse IP de ce premier sous-réseau.

Alors 172.18.0.(0 + 64 = 64) correspond à l’adresse du prochain réseau. Donc 172.18.0.(64 – 1 = 63) correspond à l’adresse de diffusion (ou de broadcast).

La première adresse disponible est donc 172.18.0.(0 + 1 = 1).

La dernière adresse disponible est donc 172.18.0.(63 – 1 = 62).

* SR2 :

Puisque 28 = 256 et que dans le masque de base on a : 16 bits à 0. Il faut donc ajouter 16 – 8 = 8 bits à 1 pour obtenir le masque de sous-réseau : 11111111 11111111 11111111 00000000, soit 255.255.255.0 en décimal pointé, soit /24 en notation CIDR.

On a déterminé précédemment le masque du prochain sous-réseau : 172.18.0.64 correspond à l’adresse IP de ce second sous-réseau.

Alors 172.18.(0 + 1 = 1).(64 + 256 = 320 sauf qu’il n’est pas possible d’avoir 320 comme valeur pour un octet, on fait donc 320 – 256 = 64 et on ajoute 1 à l’octet précédent, soit le troisième), ce qui nous donne : 172.18.1.64, correspond à l’adresse du prochain réseau. Donc 172.18.1.(64 – 1 = 63) correspond à l’adresse de diffusion (ou de broadcast).

La première adresse disponible est donc 172.18.0.(64 + 1 = 65).

La dernière adresse disponible est donc 172.18.1.(63 – 1 = 62).

* SR3 :

Puisque 29 = 512 et que dans le masque de base on a : 16 bits à 0. Il faut donc ajouter 16 – 9 = 7 bits à 1 pour obtenir le masque de sous-réseau : 11111111 11111111 11111110 00000000, soit 255.255.(27 + 26 + 25 + 24 + 23 + 22 + 21 = 128 + 64 + 32 + 16 + 8 + 4 + 2 ⬄ 255 – 1 = 254).0 en décimal pointé, soit /23 en notation CIDR.

On a déterminé précédemment le masque du prochain sous-réseau : 172.18.1.64 correspond à l’adresse IP de ce troisième sous-réseau.

Alors 172.18.(1 + 2 = 3).(64 + 512 = 576 ce qui n’est pas possible comme valeur pour un octet, donc 576 – 256 = 320 ce qui n’est toujours pas possible, donc 320 – 256 = 64 et on ajoute 2 à l’octet précédent, soit le troisième), ce qui nous donne : 172.18.3.64, correspond à l’adresse du prochain réseau. Donc 172.18.3.(64 – 1 = 63) correspond à l’adresse de diffusion (ou de broadcast).

La première adresse disponible est donc 172.18.1.(64 + 1 = 65).

La dernière adresse disponible est donc 172.18.3.(63 – 1 = 62).