

Parcours de formation SIN

**Déploiement d’une interface de communication web**

Module SIN412 : COMMANDES LINUX ET PROGRAMMATION C

SOUS LINUX

**Eléments de correction**

Durée : 3h

Objectifs à atteindre : Acquérir la connaissance des commandes fondamentales des systèmes d'exploitation Unix et Linux.

Niveau des connaissances envisageable : BAC

Pré requis : - Maitrise de l’environnement et du système de fichier de linux Ubuntu.

- Le langage C

Logiciels utilisés : - VMware Workstation et/ou Player.

- Dernière version stable de Linux Ubuntu

- Putty

- gcc

Outils mobilisés, le cas échéant : tous les tutoriels disponibles sur internet

Webographie :

<http://www.siteduzero.com/tutoriel-2-96-linux.html>

<http://www.ubuntu-beginner.com/>

<http://doc.ubuntu-fr.org/terminal>

<http://www.linux-france.org/article/debutant/debutant-linux.html>

<http://doc.ubuntu-fr.org/tutoriel/console_ligne_de_commande>

<http://doc.ubuntu-fr.org/tutoriel/script_shell>

<http://doc.ubuntu-fr.org/console>

Illustrations :

Copies d’écran Linux Ubuntu – Terminal – M. Silanus.

# Les principales commandes linux

## Le shell

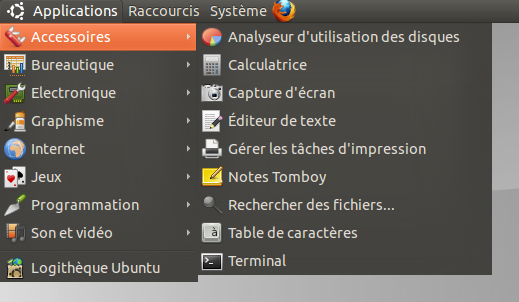
## Le terminal *shell*

* **Ouvrez le terminal de linux Ubuntu.**

**Menu Application**

**-> Accessoires**

**-> Terminal**

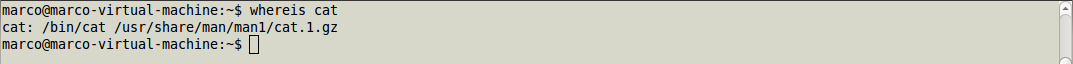
****

### Qu’est-ce qu’une commande ?

* **Recherchez les dossiers qui contiennent la commande ls : A l’invite de commande saisissez  whereis ls**

****

* **Recherchez les dossiers qui contiennent la commande cat**

****

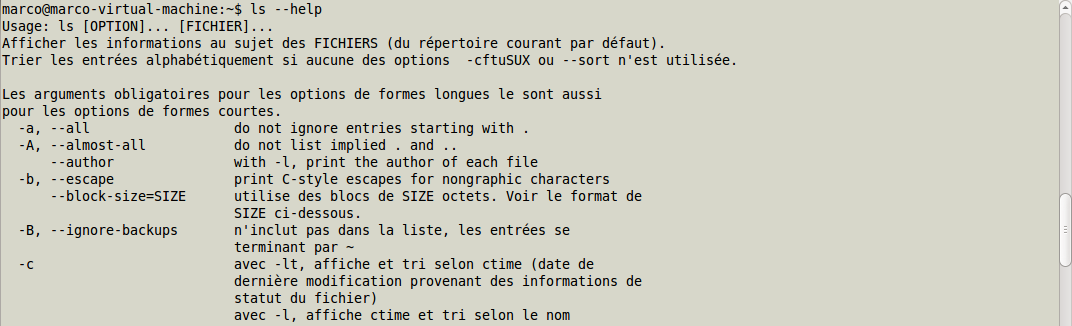
* **Identifiez les chemins d’accès aux commandes qui vous sont accessibles : echo $PATH**

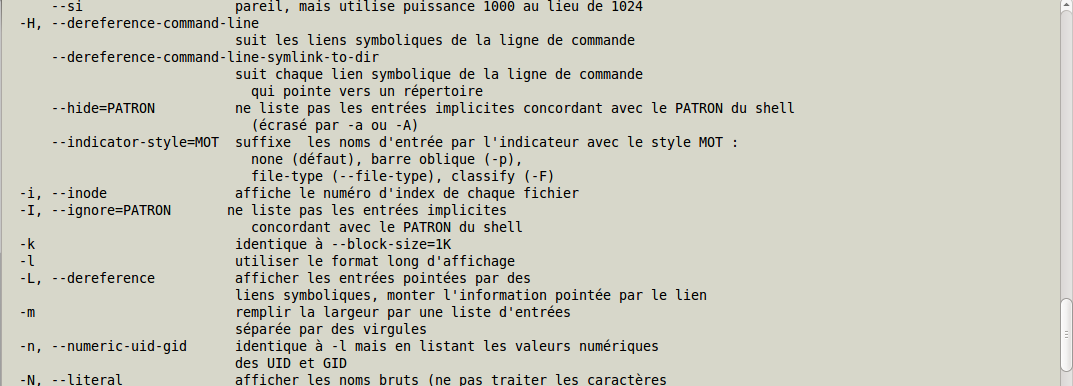
****

### Format d’une commande

* **Saisissez la commande ls --help et cherchez les options nécessaires pour afficher en format long la liste des fichiers contenu dans le dossier personnel de l’utilisateur courant (/home/nom\_utilisateur), y compris les fichiers cachés (qui commencent par « . » ex : .texte.txt )**

**ls – al /home/nom\_utilisateur ou ls – al ~**

****

****

# Les utilisateurs et les groupes

|  |
| --- |
| * **Quelle option de la commande useradd permet de créer le dossier personnel du nouvel utilisateur.**      * **Octroyez-vous les droits *root* : sudo su (le mot de passe est celui de l’utilisateur courant).**      * **Créez un nouvel utilisateur nommé *stagiaire* et créez son dossier personnel (/home/stagiaire).** * **Créez son mot de passe : passwd stagiaire** * **Créez un dossier *test* dans le dossier personnel de *stagiaire*.** * **Listez avec les détails les fichiers et les dossiers de l’utilisateur *stagiaire*.**     Groupe  Propriétaire  **Constatez que le propriétaire du dossier test est *root* et bien que ce dernier soit placé dans le dossier de stagiaire, celui-ci ne peut pas y écrire. Seul *root* possède tous les droits (rwx : read, write, execution) sur ce dossier.**   * **Déconnectez-vous de *root* : exit** * **En tant que simple utilisateur, tentez de supprimer le dossier *test*.** * **En tant que simple utilisateur, tentez de supprimer l’utilisateur *stagiaire* ainsi que son dossier personnel.**     Le dossier *test* appartient à *root* et un utilisateur autre que *root* ne peut pas le modifier ou le supprimer. Le dossier personnel de *stagiaire* lui appartient et un autre utilisateur ne peut pas le modifier ou le supprimer. Dans les deux cas, la permission n’est pas accordée.   * **Refaite de même en tant que *root*.**      * **Pour la suite du travail, créez un second utilisateur (avec le nom de votre choix).** |

## Les redirections

# Le système de fichier

## Organisation

|  |
| --- |
| * **En tant que simple utilisateur, créez un dossier dans votre dossier personnel, puis listez en le contenu.**     Groupe  Propriétaire   * **Qui est le propriétaire par défaut du dossier créé ? Qui est le groupe par défaut ?**   Propriétaire : utilisateur connecté  Groupe : groupe principal de l’utilisateur connecté   * **Quels sont les droits d’accès du dossier actuel en notation numérique octal ?**   rwxr-xr-x : 755   * **Reconnectez-vous au système en utilisant un autre utilisateur. Vérifiez que vous ne pouvez pas accéder en écriture au dossier précédemment créé.**   **Remarque** : l’utilisateur toto a été créé avec la commande standard linux useradd. Son interface Terminal est sommaire (pas de repérage du dossier en cours, complétion des commandes indisponible, …    Ubuntu dispose d’une commande plus complexe pour la création d’utilisateur système : addsuer. Elle permet de créer des utilisateurs avec toutes les options et la création du dossier personnel et son interface Terminal est complète.   * **Modifiez les droits pour que tout le monde puisse avoir un accès complet au dossier. (attention, vous devez être *root* ou le propriétaire du dossier)**     Toto n’appartient pas au groupe sudoers qui correspond aux utilisateurs pouvant bénéficier des droits d’administration. Il faut utiliser un autre utilisateur appartenant à ce groupe. C’est le cas de l’utilisateur créé lors de l’installation d’Ubuntu et de ceux créés avec la commande **adduser**.     * **Testez à nouveau l’accès en écriture du dossier.**      * **Attribuez à *root* la propriété du dossier. (attention, vous devez être *root*).** |

# Les commandes réseau

|  |
| --- |
| * **Quelle est l’adresse IP et le masque de votre interface réseau ?**      * **Désactivez votre interface réseau, puis affichez ses paramètres. Vérifiez l’accès au réseau.**      * **Réactivez-la et vérifiez de nouveau l’accès réseau.** |

# Exercices

## Manipuler un dossier ou un fichier

|  |
| --- |
| Déplacez-vous dans le dossier Documents/ et listez le. Créez un nouveau dossier appelé ExoShell et listez à nouveau le dossier Documents/ .  Commandes utilisées :  **$ cd ~/Documents**  **$ ls –l**  **$ mkdir ExoShell**  **$ ls –l** |

|  |
| --- |
| Déplacez-vous dans le dossier ExoShell et lister son contenu. Essayez de le supprimer tout en restant dedans. Que se passe t-il ?  Commandes utilisées :  **$ cd ExoShell**  **$ ls –l**  **$ rm –r . (le . indique le répertoire courant)**  Faites en sorte de pouvoir le supprimer. **(On ne peut pas supprimer le dossier dans lequel on se trouve)**  Commandes utilisées :  **$ cd ..**  **$ rm –r ExoShell**  **$ ls –l** |

|  |
| --- |
| Créer le fichier texte.txt et insérez-y le texte suivant : (Copier-coller)  *Linux, ou GNU/Linux, est un système d’exploitation compatible POSIX de type UNIX. Linux est fondé sur le noyau Linux, logiciel libre créé en 1991 par Linus Torvalds. Il est souvent utilisé avec les logiciels du système d’exploitation libre GNU.*  *Développé sur Internet par des milliers d’informaticiens bénévoles et salariés, Linux fonctionne maintenant sur du matériel allant du téléphone mobile au superordinateur. Il existe de nombreuses distributions Linux indépendantes, destinées aux ordinateurs personnels et aux serveurs informatiques, pour lesquels Linux est très populaire. Elles incluent des milliers de logiciels issus de la communauté du logiciel libre et fréquemment quelques logiciels propriétaires. Linux est également populaire sur système embarqué. La mascotte de Linux est le manchot Tux.*  Commandes utilisées :  **$ echo "Linux, … " > texte.txt**  Ajoutez à la fin du fichier la source de cet article : *Linux - définition par Wikipedia*  Commandes utilisées :  **$ echo "** **Linux - définition par Wikipedia " >> texte.txt**  Affichez le contenu du fichier.  Commandes utilisées :  **$ cat texte.txt** |

|  |
| --- |
| Créez un dossier appelé texte et copiez-y le fichier texte.txt précédant. Listez le contenu de ce dossier.  Commandes utilisées :  **$ mkdir texte**  **$ cp texte.txt texte/**  Renommez le fichier texte.txt en linux.txt  Commandes utilisées :  **$ mv texte/texte.txt texte/linux.txt** |
| Supprimez le fichier texte.txt  Commandes utilisées :  **$ cd ~/Documents**  **$ rm texte.txt** |

## Modification des droits d’accès

|  |
| --- |
| Qui est le propriétaire du dossier texte ?  Commandes utilisées :    **Propriétaires**  Attribuez la propriété du dossier texte à *root* et interdisez l’accès à ce dossier à tous les utilisateurs (sauf *root*).  Commandes utilisées :  **# chown root texte**  **# chmod 700 texte**  Qui est le propriétaire du fichier linux.txt contenu dans le dossier texte ?  Commandes utilisées :  **# cd /home/utilisateur/Documents/texte**  **# ls –l**  **=> Linux est toujours la propriété de l’utilisateur qui l’a créé.**  **La modification de la propriété ou des droits d’accès d’un dossier n’entraine pas la modification de son contenu. Il faut ajouter l’option –R (Récursif) pour modifier les propriétés d’un dossier et son contenu.**    En tant qu’utilisateur propriétaire de ce fichier, pouvez-vous lire son contenu ? Pourquoi ?  Commandes utilisées :  **# exit**  **$ cat /home/utilisateur/texte/linux.txt**  **=>Permission non accordée**  **En effet, bien que vous soyez propriétaire du fichier linux.txt et que vous ayez tous les droits sur lui,il se trouve dans un dossier qui vous est interdit. Le problème n’est donc pas les droits sur le fichier mais sur le dossier qui le contient.**    Modifiez les droits d’accès nécessaire pour que seuls *root* et le propriétaire du fichier puissent le lire et le modifier.  Commandes utilisées :  Le propriétaire du fichier est le seul à appartenir à son groupe. Par défaut, lors de la création d’un nouvel utilisateur, linux créer un groupe à son nom si aucun groupe principal n’est imposé.  L’idée consiste donc à attribuer tous les droits à son groupe sur le dossier.  **# chmod 770 texte**    **groupes** |

## Rechercher un fichier ou un dossier

|  |
| --- |
| En tant que super-utilisateur (*root*), chercher l’emplacement du dossier texte à partir de la racine.  Commandes utilisées :  **# find / –name texte**  A partir de la racine, chercher l’emplacement du fichier linux.txt  Commandes utilisées :  **# find / –name linux.txt** |

## Rechercher un mot dans un fichier

|  |
| --- |
| Recherchez le mot « linux » dans le fichier linux.txt indépendamment de la casse. (placez-vous dans le dossier où se trouve le fichier linux.txt).  Commandes utilisées :  La commande à utiliser est la commande **grep**. Pour connaitre la manière de l’utiliser, utilisez l’aide :  **# grep --help**  L’aide nous indique qu’il faut utiliser l’option **–i** pour rendre la recherche insensible à la casse : |

## Suppression d’un dossier non vide

|  |
| --- |
| Supprimez le dossier texte et tout ce qu’il contient en une seule opération.  Commandes utilisées :  **# rm –r /home/utilisateur/Documents/texte**  **(il faut être en dessous de texte dans l’arborescence des dossiers)** |

# Programmation en C sous linux

## Premier programme, compilation et exécution

Modifiez votre programme pour qu'il affiche « **Bonjour tout le monde** » avec un retour à ligne à la fin.

|  |
| --- |
| #include<stdio.h>  main()  {  printf("Hello World\n");  printf("Bonjour tout le monde\n");  } |

## Exercices

* Un nom est saisi au clavier (ex : robert) puis l’ordinateur affiche " Bonjour robert ", (utiliser **scanf** de la librairie **stdlib.h** et **printf**)

|  |
| --- |
| /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \* nom : bonjour.c  \* auteur : Marc Silanus  \* date : 14/02/2011  \*  \* Formation STI2D-SIN / Programmation en C sous linux  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  #include <stdio.h>  #include <stdlib.h>  int main ()  {  char prenom[20];  printf ("Quel est ton prénom ?\n");  scanf ("%19s",prenom);  printf("Bonjour %s...\n",prenom);  return 0;  } |

* Après avoir entré la longueur et la largeur le programme retourne le périmètre,   
  (utiliser une fonction : **float perimetre(float l,float L) ;**)

|  |
| --- |
| /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \* nom : perimetre.c  \* auteur : Marc Silanus  \* date : 14/02/2011  \*  \* Formation STI2D-SIN / Programmation en C sous linux  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  #include <stdio.h>  #include <stdlib.h>  float L,l;  float perimetre(float a, float b);  int main(void)  {  printf("Calcul du périmètre d'un rectangle\n");  printf("\nSaisissez la longueur : ");  scanf("%f",&L);  printf("\nSaisissez la largeur : ");  scanf("%f",&l);  printf("\nCalcul du périmètre : %5.2f\n",perimetre(L,l));    }  float perimetre(float a, float b)  {  return (a+b)\*2;  } |

* Recherchez la racine réelle d'une équation du premier degré.

|  |
| --- |
| /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \* nom : degre1.c  \* auteur : Marc Silanus  \* date : 14/02/2011  \*  \* Formation STI2D-SIN / Programmation en C sous linux  #include <stdio.h>  #include <stdlib.h>  int main ()  {  float a,b,c,x;  a=0;  printf ("Résolution d'une équation du premier degré : ax+b=c\n");  printf ("Entrez la valeur de a (différent de 0) : ");  while(a==0) scanf ("%f",&a);  printf ("Entrez la valeur de b : ");  scanf ("%f",&b);  printf ("Entrez la valeur de c : ");  scanf ("%f",&c);  x=(c-b)/a;  printf("La racine est : x=%f\n",x);  return 0;  } |

* Recherchez les racines réelles d'une équation du second degré.   
  (utilisez la librairie **Math.h** et l’option **–lm** pour la compilation)

|  |
| --- |
| /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \* nom : degre2.c  \* auteur : Marc Silanus  \* date : 14/02/2011  \*  \* Formation STI2D-SIN / Programmation en C sous linux  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  #include <stdio.h>  #include <stdlib.h>  #include <math.h>  int main ()  {  float a,b,c,x1,x2,delta;  a=0;  printf ("Résolution d'une équation du second degré dans R : ax²+bx+c=0\n");  printf ("Entrez la valeur de a (différente de 0) : ");  while(a==0) scanf ("%f",&a);  printf ("Entrez la valeur de b : ");  scanf ("%f",&b);  printf ("Entrez la valeur de c : ");  scanf ("%f",&c);  delta=((b\*b)-(4\*a\*c));  if(delta>0)  {  x1=(-b-sqrt(delta))/(2\*a);  x2=(-b+sqrt(delta))/(2\*a);  printf("Les racines sont : \nx1=%f\nx2=%f\n",x1,x2);  }  else if(delta==0)  {  x1=(-b)/(2\*a);  printf("Une racine unique : \nx=%f\n",x1);  }  else  printf("Pas de racines réelles\n");  return 0;  } |
| **# gcc degre2.c –o degre2 –lm** (-l : lier avec une libraire, m : raccourci pour math)  **# ./degre2** |

* Recherchez un nombre aléatoire entre 0 et 999, à chaque essai le programme indique « trop grand » ou « trop petit » et en cas de réussite le nombre d’essais, (utiliser **if**, **do while**, **rand**)

|  |
| --- |
| /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \* nom : chiffre-secret.c  \* auteur : Marc Silanus  \* date : 14/02/2011  \*  \* Formation STI2D-SIN / Programmation en C sous linux  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  #include <stdio.h>  #include <stdlib.h>  #include <time.h>  int main ()  {  int iSecret, iTente, Compt;  /\* initialisation de la fonction rand \*/  srand ( time(NULL) );  /\* génération du nombre secret \*/  iSecret = rand() % 999;  /\* initialisation du compteur de coups \*/  Compt=0;  do  {  Compt++;  printf ("Entrez un nombre (0 to 999): ");  scanf ("%d",&iTente);  if (iSecret<iTente) printf("Le nombre secret est plus petit !\n");  else if (iSecret>iTente) printf("Le nombre secret est plus grand !\n");  } while (iSecret!=iTente);  printf("Félicitation !\nVous avez trouvé en %d coups\n",Compt);  return 0;  } |