

1) Sessions multiples et écrans virtuels

Que réalise les associations de touches suivante : <Ctrl>+<Alt>+<F1> à <Ctrl>+<Alt>+<F6> et <Ctrl>+<Alt>+<F7>

2) Quelques commandes

Que réalise la commande "cal" et analyser les options de cette commande? Afficher le mois actuel avec le mois précédent et suivant, afficher les mois de l'année actuelle, le mois actuel avec les 6 précédents et les 6 suivants, le mois de Juillet de l'année prochaine. Quelles sont les options de la commande "date", proposer un affichage de format "YYYY-MM-DD" et peut on modifier la date du système avec cette commande ?

3) Parcourir et lister les répertoires, définition de chemin

En mode console, que réalise les commandes suivantes : (analyser et commenter)

Manuel : **'man man'** ('q' pour quitter)

Print Working Directory : 'pwd'. Remarque : le chemin peut être absolu (indiqué à partir de la racine) ou relatif (à partir du répertoire courant)

Lister le contenu de répertoire : 'ls', 'ls .', 'ls ..', 'ls -l', 'ls -a', 'ls -la', 'ls /etc'.

Changer de répertoire : 'cd -help', 'cd ..', 'cd /', comment précise-t-on le répertoire actuel, 'cd', 'cd ~', 'cd \$HOME', 'echo \$HOME'

MaKe/ReMove DIRectory : 'mkdir TP', 'cd TP', 'mkdir rep1', 'rmdir rep1'

ReMove : pour la commande 'rm' que signifie les options -f, -i et -R

Déterminer le type de contenu d'un fichier : 'file fichier' en utilisant un fichier existant, voir 'file *'. Quel est le rôle de l'option '-z' ?

Déplacer ou renommer : 'mv ressource1 ressource2' que signifie l'option -R

Créer un sous répertoire de travail '~/.TP/TP1' pour la suite du TP à partir du répertoire de base de l'utilisateur(home-directory).

4) Manipulation d'éditeur de texte mode console et vérification de la compilation facile sous linux.

Échantillons des Raccourcis claviers pour	sous 'vi' en console texte	sous 'emacs' texte ou graphique	Sous 'nano' en console texte
Charger un fichier	:r fichier	Ctrl x Ctrl f	Ctrl r
Sauvegarder sans quitter	:w	Ctrl x Ctrl s	Ctrl o
Sauvegarder et quitter	:wq	Ctrl x Ctrl s Ctrl x Ctrl c	
Sauvegarder sous	:w file	Ctrl x Ctrl w	
Recharge le fichier	:e!		
Insérer du texte derrière le curseur	a puis Echap quitte insertion		
Insérer du texte devant le curseur	i puis Echap quitte insertion		
Insérer du texte à la fin de la ligne	A puis Echap quitte insertion		
Insérer du texte en début de ligne	I puis Echap quitte insertion		
Copier		Début 'Ctrl espace' puis 'Alt w' pour fin	
Couper		Début 'Ctrl espace' puis 'Ctrl w' pour fin	
Coller		Ctrl y	
Annuler	u	Ctrl	
Supprime 1 caractère	x	Suppr	
Supprime 1 mot	dw	Alt d	
Supprime n mots	dnw		
Supprime ligne	dd	Ctrl k	
Supprime n lignes	ndd		
Copie une ligne	Y		
Copie n lignes	nY		
Colle les lignes avant le curseur	P		
Colle les lignes après le curseur	p		
Quitter	:q ou pour forcer :q!	Ctrl x Ctrl c	Ctrl x

- Pour concevoir un exécutable en mode console :

- créer un fichier source en C "**nano aff.c**" contenant le code suivant :

```
#include <stdio.h> //Sous Ubuntu, verifier l'installation de libc6-dev
int main (int argc, char * argv[])
{
    int i;
    printf ("%s en execution\n",argv[0]);
    for (i=0;i<argc;i++) printf ("argv[%d] = \"%s\"\n",i,argv[i]);
    return 0;
}
```

- Exécuter et analyser : 'man printf', 'gcc aff.c', './a.out', './a.out param1 -l -help', 'cat aff.c', 'cat -n aff.c', 'file a.out', 'gcc -S aff.c', 'file aff.s' et 'more aff.s'

- Le même exercice en C++ :

- créer un fichier source en C++ "**nano aff.cpp**" contenant le code suivant :

```
#include <iostream>
using namespace std;
int main (int argc, char * argv[]){
    int i;
    cout << argv[0]<< " en execution" << endl;
    for (i=0;i<argc;i++) cout << "argv["<<i<<"] = \""<< argv[i] << "\"<<endl;
    return 0;
}
```

- Exécuter et analyser les commandes : 'g++ aff.cpp -o aff', './aff', './aff param1 second'

5) Fichiers et commandes associés aux supports de stockage :

- Visualisez `'/proc/filesystems'` (man filesystems)
- Visualiser le fichier `/var/log/dmesg` (ou lancer la commande `'dmesg'`) et commenter le (ou la).
- Préparation de montage manuel d'un support déjà formaté :
 - Insérer un support mémoire usb (clé ou disque)
 - Repérer avec la commande `'dmesg'`, la désignation de ce nouveau support `'/dev/sd_1'` détecté
 - Observer l'évolution du contenu du fichier `'/etc/mtab'` à chaque montage/démontage.
 - Créer un répertoire pour préparer le montage, ex : `'~/mnt'`
 - Analyser la commande `'mount'` et activer l'accès à ce nouveau support amovible `'/dev/sd_1'` avec le point de montage `'~/mnt'`. Donnez toutes les options nécessaires pour ce montage.
- Pour enlever un support, il faut absolument le démonter du système (tampon R/W). Préciser la commande permettant le démontage et ses options.
- Pour simplifier le montage de support non amovible (sata, ide, scsi, ...), il est possible de préconfigurer le montage grâce au fichier `'/etc/fstab'` : Observer `'/etc/fstab'` permettant d'indiquer comment monter les supports du système par rapport : aux utilisateurs, en rw, sur quel point de montage, en automatique, avec vérification, ...

`<peripherique>` `<pointDeMontage>` `<typeFS>` `<options>` `<dump>` `<pass>`
 dump(filesystem backup): Ce champ est utilisé par la commande dump pour déterminer quels systèmes de fichiers doivent être sauvegardés. La valeur par défaut est zéro (ne pas sauvegarder) s'il n'est pas présent.

Pass: Ce champ est utilisé par la commande fsck pour déterminer l'ordre dans lequel les vérifications du système de fichiers sont effectuées au démarrage. Le système de fichiers racine doit être spécifié avec un numéro à 1. Les autres systèmes de fichiers doivent avoir un numéro à 2. La valeur par défaut est zéro (pas de vérification) s'il n'est pas présent.

- Visualiser le contenu du fichier `'/etc/mtab'` (commenter les)
- Expliquer les commandes `'mkfs'`, `'mkfs.____'`
- Préparer le montage/démontage d'un **CDROM** manuellement. Que réalise la commande `'eject'` et avec l'option `'-t'` ou `'-T'`

6) Un peu plus de matériel

- Analyser les commandes `'df'`, `'free'`
- Analyser le contenu du fichier `'/proc/cpuinfo'`
- Analyser la commande `'sudo dmidecode'` et répondre aux questions suivantes :
 - Comment connaître la marque et le modèle de la carte mère?
 - Comment connaître la version du BIOS?
 - Combien de barrettes mémoire sur la machine?
 - Combien de slots PCI sont utilisés?
 - Analyser le script shell suivant et l'exécuter :

```
for obj in bios-vendor bios-version bios-release-date system-manufacturer system-product-name system-version system-serial-number system-uuid system-family baseboard-manufacturer baseboard-product-name baseboard-version baseboard-serial-number baseboard-asset-tag chassis-manufacturer chassis-type chassis-version chassis-serial-number chassis-asset-tag processor-family processor-manufacturer processor-version processor-frequency
```

```
do
a=`dmidecode -s ${obj}`
echo "${obj} : $a"
done
```

- Analyser la commande `'lspci'` ou `'scanpci'`
- Analyser la commande `'/usr/sbin/lusb'` et éventuellement l'utilitaire graphique `'usbview'`
- Analyser la commande `'ifconfig'`
 - Comment observer les interfaces actives ?
 - Comment observer les interfaces même si elles ne sont pas actives ?
 - Comment observer une interface ?
 - Comment changer l'adresse IP et le masque d'une interface ?
 - Comment activer/désactiver une interface ?
- Analyser la commande `'ip [OPTIONS] OBJECT{ COMMAND | help}'`
 - Comment observer les adresses des interfaces ?
 - Comment afficher la table de routage réseau IPv4 du système ?
 - Pour avoir plus de précision sur le paramètre COMMAND d'un OBJECT : `'ip <objet> help'`, exemple `'ip address help'`
 - Que réalise la commande : `'ip link help vlan'` ?
 - Comment activer/désactiver une interface ?
 - Comment observer les statistiques des interfaces ?
 - Comment ajouter/enlever une adresse IP et le masque d'une interface ?

- Analyser la commande '**iwconfig**'
 - Quels modes peut on affecter à l'interface Wifi?
 - Comment peut on choisir le mode de l'interface Wifi ?
 - Comment peut on choisir le canal Wifi ?
 - Comment activer/désactiver la radio d'une interface ?

- Analyser la commande '**hdparm**'. Que réalise :
 - '**sudo hdparm -i /dev/sda**' ou autre support présent dans le résultat de la commande df
 - '**sudo hdparm -t /dev/sda**'
 - Comment obtenir la géométrie du disque ?
 - Comment obtenir la température Celsius du disque ?

- Analyser la commande '**lshw**' (list hardware)
 - Il y a t il une différence lorsque cette commande est exécutée en mode 'super user' ou pas ?
 - Comment choisir un affichage texte complet, texte condensé, html, xml ou json ?
 - Comment exporter le résultat dans un fichier html ?
 - Sachant que les classes commencent avec le caractère '*', comment sélectionner que les interfaces réseau ?
 - Comment sélectionner les composants multimédia ?

- Analyser la commande '**uname**' avec l'option '-a'
 - Donner le numéro de la version du Kernel
 - Donner le nom réseau de la station
 - Donner le type de processeur