



Guide succinct de GNU/Linux pour le débutant

11 novembre 2013

Autiwa

Résumé

Recueil des commandes et astuces que j'ai glané à droite à gauche au fil de mon apprentissage de Linux. Ceci s'adresse plutôt aux utilisateurs d'Ubuntu étant donné que ces astuces et commandes ont été faites et trouvées sous Ubuntu, mais certaines choses s'appliquent de manière générale.

Table des matières

1	Astuces Générales	5
1.1	Connexion distante (XDMCP)	5
1.2	Créer un .iso en ligne de commande	5
1.3	Créer des modèles de documents	5
1.4	Dossiers Utiles	6
1.5	Exécuter des tâches de manière périodique	6
1.5.1	cron et anacron	6
1.6	Fichiers Utiles	7
1.7	Forcer la suppression des fichiers de la corbeille	7
1.8	La stégano pour les nuls	7
1.9	Raccourcis	8
1.9.1	Les raccourcis standards sous gnome	8
1.10	Rajouter des polices de caractères	8
1.11	Sélectionner des fichiers avec motif (dans Nautilus)	8
1.12	Réaliser des opérations sur un fichier .odt à la main	9
2	Configuration du système	9
2.1	Restaurer un système RAID1 « dégradé »	9
2.2	Regénérer le menu de multiboot de Grub	10
2.3	Quelques commandes pour débloquer le système	10
2.3.1	Activation / Désactivation de SysRq	11
2.3.2	Les commandes	11
3	Équivalents windows	11
3.1	Calcul formel : Xcas	11
3.1.1	Xcas	11
3.1.2	Wxmaxima	12
3.1.3	Octave/scilab	13
3.2	Client Mail : Evolution	13
3.3	Comptabilité : Grisbi	13
3.4	Dessin Vectoriel : Inkscape	13
3.5	Digitizer : Engauge	13
3.6	Éditeur \LaTeX : Kile	13
3.7	Éditeur de Texte : Kate	14
3.8	Encodeur Vidéo : Avidemux	14
3.8.1	Options de montage et de rognage	14
3.8.2	Encoder un DVD	14
3.9	Gestionnaire de photo : Digikam	15
3.10	IDL : GDL	15
3.10.1	fonctions mathématiques standard	15
3.10.2	Les tableaux	15
3.10.3	les boucles	17
3.10.4	Les variables aléatoires	17
3.10.5	Astuces	18
3.10.6	Erreurs	18
3.11	Lecteur de musique : Quod Libet	18
3.12	Lecteur PDF : Okular	18
3.13	Lecteur Vidéo : SMPlayer	19
3.14	Manipulateur de Fichiers : Métamorphose	19
3.15	Matlab : Octave	19
3.15.1	Ajouter des fichiers .m dans octave	19
3.15.2	Indexage logique	19
3.15.3	Tracer un damier	20
3.15.4	Lettre grecques dans les labels	20
3.15.5	Changer les symboles et styles de ligne	21
3.15.6	Exporter des figures en .svg	21
3.15.7	subplot	22

3.15.8 Options avancées de plot	22
3.16 Navigateur Web : Firefox	23
3.16.1 Reprendre un téléchargement interrompu sous Firefox avec wget	23
3.17 Retouche Photo : The Gimp	23
3.17.1 Retouche d'Éléments gênants avec Gimp	23
3.17.2 Brosse dynamique	24
3.17.3 Créer sa propre brosse	24
3.18 Tableur : Gnumeric	24
3.19 Visionneur D'Images : gThumb	24
4 L^AT_EX sous GNU/Linux	24
4.1 Installation	24
4.2 Astuces	25
5 La ligne de commande	25
5.1 Astuces générales pour la console	25
5.1.1 Arrêter une commande en cours d'exécution	25
5.1.2 Raccourcis claviers	25
5.1.3 Rechercher une commande : Ctrl + R	25
5.1.4 Stocker dans un fichier : >	25
5.1.5 Stocker à la suite d'un fichier : »	26
5.1.6 Exécuter en tâche de fond : &	26
5.1.7 Passer les arguments entre commandes : 	26
5.1.8 &&	26
5.1.9 Argument précédent : !\$	26
5.1.10 Le bash {} expansion	26
5.1.11 Faire des maths : \$(())	27
5.1.12 Substitution de Commande	27
5.1.13 Mettre une commande en background a posteriori	27
5.2 Présentation des commandes usuelles	27
5.2.1 cd	27
5.2.2 chmod	27
5.2.3 dir	28
5.2.4 du	28
5.2.5 find	28
5.2.6 finger	28
5.2.7 htop	28
5.2.8 kill	28
5.2.9 killall	28
5.2.10 locate	28
5.2.11 ln	29
5.2.12 ls	29
5.2.13 man	29
5.2.14 mkdir	29
5.2.15 pidof	29
5.2.16 ps	29
5.2.17 pwd	29
5.2.18 qui	29
5.2.19 rm	29
5.2.20 rwho	29
5.2.21 rwrite	29
5.2.22 shutdown	30
5.2.23 sort	30
5.2.24 top	30
5.2.25 uniq	30
5.2.26 vim	30
5.2.27 wget	30
5.2.28 ytalk	31
5.3 Présentations de programmes plus spécifiques	31

5.3.1	alltray	31
5.3.2	Mencoder	31
5.3.3	ffmpeg	31
5.3.4	pdftk : Manipuler les fichiers .pdf	32
5.3.5	record : enregistrer des sons	32
5.3.6	sed : manipuler les chaînes de caractères	33
5.3.7	convert : Convertir un .pdf en .png	33
5.3.8	inkscape en ligne de commande	33
5.3.9	zenity	34
5.4	Actions faciles à faire en ligne de commande	35
5.4.1	Effacer des fichiers en masse	35
5.4.2	Changer les droits des dossiers de manière récursive	35
5.4.3	Convertir un .ps en .pdf	35
5.4.4	Déplacer des fichiers dans le répertoire parent	35
5.4.5	Changer l'encodage d'un texte	35
5.4.6	Inverser l'ordre des lignes	35
5.4.7	Lire le début ou la fin d'un fichier	36
5.4.8	Extraire un DVD avec mplayer	36
5.4.9	Extraire les sous-titres d'un fichier .mkv	36
5.4.10	Rechercher/Remplacer de manière récursive dans plusieurs fichiers	37
5.5	Mettre de la couleur dans la console	37
5.5.1	Trier les lignes d'un fichier par ordre alphabétique	37
5.6	Manipuler un fichier texte	37
5.6.1	Expressions régulières regexp	37
5.6.2	Rechercher-remplacer regexp	38
5.6.3	Supprimer des lignes vides	39
5.6.4	Extraire certaines colonnes d'un fichier	39
6	Faire des scripts Bash	39
6.1	Boucle for	39
6.1.1	Boucle sur des fichiers	39
6.2	Manipuler les chaînes de caractères	40
6.2.1	<code>\${chaine#souschaine}</code>	40
6.2.2	<code>\${chaine##souschaine}</code>	40
6.2.3	<code>\${chaine%souschaine}</code>	40
6.2.4	<code>\${chaine%%souschaine}</code>	40
6.3	Faire une pause dans un script bash	40
6.4	Découpage des chemins	41
6.5	Tester si un fichier existe	41
6.6	Tester si un binaire existe	41
6.7	jpg en png (script compact)	41
6.8	jpg2pdf.sh	41
6.9	flv2mp3	42
6.10	dvdchapter2vob	42
	Glossaire	44

1 Astuces Générales

1.1 Connection distante (XDMCP)

Il m'est venu la lubie de me connecter à mon ordinateur fixe depuis l'extérieur, via son adresse IP. j'ai mis du temps à comprendre ce qu'il fallait faire, vu qu'au début, j'ai cherché du côté de vnc.

Ce qui suit permet d'arriver à l'écran de connection et de s'authentifier sur la machine distante. Ceci de deux manières d'ailleurs, soit en session complète (la session du client va être une session distante), soit dans une fenêtre (la session distante sera simplement une fenêtre de la session locale du client).

Pour pouvoir se connecter à distance, il faut dans un premier temps activer la connexion à distance sur le serveur.

Remarque : J'appelle **serveur** l'ordinateur fixe sur lequel on souhaite se connecter, et **client** l'ordinateur sur lequel on veut accéder à l'ordinateur distant.

Pour cela, il faut aller dans Système > Administration > Fenêtre de connexion et activer la connection à distance dans l'onglet prévu à cet effet. Choisissez l'option qui permet d'avoir une connection distante identique à la connection locale.

Si vous êtes derrière un routeur comme c'est de plus en plus le cas avec les box, vous devez créer une règle qui va router le port 177 (UDP) du routeur vers le port 177 du serveur. Il se peut qu'il soit pratique de donner une IP locale fixe au serveur.

Ensuite, côté client, il suffit, à l'écran de connexion de sélectionner "connexion distante via XDMCP" et de donner l'IP locale ou distante du serveur.



Activer la connexion distante présente un sérieux risque de sécurité, en effet, n'importe qui connaissant l'IP du serveur peut accéder à l'écran de connexion. C'est pour cela que j'ai désactivé ça en attendant de savoir le faire passer via ssh.

Pour se connecter au serveur dans une simple fenêtre, il faut installer **Xnest**. Ensuite, il faut sélectionner le protocole XDMCP dans l'application **tsclient**. Si vous n'installez pas **Xnest**, le protocole XDMCP ne sera tout simplement pas disponible dans la liste déroulante. Pour lancer **tsclient**, il faut généralement aller dans Applications > Internet > Terminal Server Client

Apparemment, il semble possible de lancer une application en particulier, et ce simplement avec ssh. Je n'ai pas encore testé, mais ça pourrait être intéressant.

1.2 Créer un .iso en ligne de commande

```
mkisofs -v -r -J -o image.iso path_of_your_file
```

Remarque : La ligne que j'utilise est celle-ci (en me plaçant dans le répertoire où je veux que l'iso soit enregistrée) :

```
mkisofs -v -r -J -o mon_fichier.iso /media/cdrom0
```

1.3 Créer des modèles de documents

Pour créer des modèles de documents, il suffit de créer un dossier **Modèles** dans le répertoire utilisateur, le chemin de ce dossier sera donc **~/Modèles/**

Dans ce dossier, il suffit de créer des fichiers, ceux-ci seront considérés comme des modèles et s'ajouteront dans la liste déroulante lors de la création d'un nouveau fichier (par le clic droit par exemple).

Remarque : La première fois que j'ai fait cette manipulation, je n'ai eu besoin de rien modifier. Toutefois, j'ai dû le faire la deuxième fois. Il faut pour cela éditer le fichier **~/config/user-dirs.dirs** et modifier la variable **XDG_TEMPLATES_DIR** pour avoir la ligne suivante : **XDG_TEMPLATES_DIR="\$HOME/Modèles"**

1.4 Dossiers Utiles

Il y a certains dossiers où il devient utile de connaître le chemin tant ils sont utilisés.

- `/usr/share/pixmaps` : contient les images des icônes des programmes
- `/usr/bin` : contient des exécutables (ce qu'on lance dans la console) et vous pouvez en rajouter d'autres manuellement ici.
- `/var/www` : est le répertoire qui contient tout les dossiers et pages web du serveur web (apache ou autre)
- `/usr/share/icons` : contient à peu près tout les icones du systèmes. Les sous dossiers **scalables** contiennent les versions vectorielles en **.svg**. J'avoue que j'ai mis ça dans un coin pour aller me servir quand je veux certains trucs pour mes dessins vectoriels.

Les dossiers les plus utiles généralement, ce sont les dossiers de configuration des programmes, qui sont habituellement des dossiers cachés au nom relativement explicite dans votre dossier utilisateur. Seulement voilà, la norme, ou du moins la recommandation, c'est de mettre ces dossiers dans le dossier `/home/-nom_utilisateur/.config`. Donc il arrive qu'on ne puisse pas trouver certains dossiers de configuration. Pensez donc à regarder dans **.config**, ou dans les dossiers pour gnome. Voici maintenant une liste de dossier et programme pour lesquels le nom du dossier n'est pas le nom du programme :

Pidgin .purple

1.5 Exécuter des tâches de manière périodique

1.5.1 cron et anacron

Il existe pour cela **cron**, que l'on peut éditer en ligne de commande via **crontab**. Il existe une interface graphique à ce programme que l'on peut installer :

```
sudo apt-get install gnome-schedule
```

L'interface graphique étant suffisamment explicite, et la documentation étant accessible en tapant

```
man crontab
```

je ne détaille pas plus son fonctionnement. À noter tout de même que je n'ai pas réussi à lancer une tâche journalièrement. En effet, **@daily** permet de lancer une tâche tout les jours, mais celle-ci est lancée à heure fixe, c'est à dire 00h00. Dans mon cas, il arrive que mon ordinateur soit éteint à minuit, et donc la tâche ne se lance pas.

Pour remédier à celà, il existe une « sur-couche » nommée **anacron**. Celle-ci se base sur les tâche **cron**, et les lance si leur lancement était à un instant où l'ordinateur n'était pas allumé.

Pour éditer les lignes de lancement de tâches, il faut faire :

```
crontab -e
```

Remarque : On peut aussi utiliser **sudo crontab -e**, les tâches seront alors exécutées en tant que root. Celà dit, le répertoire par défaut sera le répertoire root, et le home sera /root. Penser donc à ne pas utiliser mais des chemins absolus pour avoir des fichiers accessibles. Si les deux disques gèrent les droits utilisateurs, **rsync** permet de copier des fichiers root même en tant que simple utilisateur (j'en suis presque sûr, mais pas totalement certain non plus. Après test, il se trouve qu'**rsync** ne le fait pas chez moi.)

Les lignes sont de la forme suivante :

```
mm hh jj MMM JJJ tâche > log
```

mm : représente les minutes (de 0 à 59)
hh : représente l'heure (de 0 à 23)
jj : représente le numéro du jour du mois (de 1 à 31)
MMM : représente le numéro du mois (de 1 à 12) ou l'abréviation du nom du mois (jan, feb, mar, apr, ...)
JJJ : représente l'abréviation du nom du jour ou le chiffre correspondant au jour de la semaine (0 représente le dimanche, 1 représente le lundi, ...)
tâche : représente la commande ou le script shell à exécuter
log : représente le nom d'un fichier dans lequel stocker le journal des opérations. Si la clause **> log** n'est pas spécifiée, **cron** enverra automatiquement un courriel de confirmation. Pour éviter cela il suffit de spécifier **> /dev/null**

Pour chaque unité de temps (minute/heure/...) les notations sont possibles :

***** : à chaque unité de temps
2-5 : les unités de temps (2,3,4,5)
***/3** : toutes les 3 unités de temps (0,3,6, ...)
5,8 : les unités de temps 5 et 8

J'utilise cette ligne de commande, qui me lance le backup à 12h et à 19h tout les jours, et anacron s'occupera de la lancer si le lancement prévu était durant une période où l'ordinateur était éteint :

```
00 12,19 * * * /home/autiwa/scripts/backup_journalier.sh >/home/autiwa/backup_journalier.log
```

Pour rajouter cette tâche en ligne de commande, il faut faire :

```
echo "00 */5 * * * autiwa /home/autiwa/scripts/cron/backup_journalier_desktop.sh"|sudo tee -a /etc/crontab
```

Remarque : Notez la présence supplémentaire de "autiwa" qui signifie que le script sera lancé par moi. On peut aussi mettre "root" si on veut.

1.6 Fichiers Utiles

/ect/fstab : contient la table des partitions, avec les options de mount (dont celle qui les fait apparaître au boot de l'ordi)
/boot/grub/menu.lst : est le fichier qui contient les options de grub, le programme qui gère le multi boot. Entre autre, on peut rajouter des OS qui n'apparaissent pas encore ou modifier le temps d'attente avant le boot de l'OS par défaut.
/etc/X11/xorg.conf : contient la configuration du serveur X et en particulier la config et le nom du pilote graphique utilisé.
/etc/apt/sources.list : contient la liste des dépôts pour l'installation et la mise à jour des programmes.

1.7 Forcer la suppression des fichiers de la corbeille

Je me suis retrouvé avec un dossier que je n'arrivais pas à supprimer. Avec les droits root en mode graphique, j'ai fait planter mon ordi plusieurs fois. Alors qu'avec une petite ligne de commande bien choisie, ça m'a pris 3 secondes.

J'ai d'abord mis du temps à trouver où était la corbeille. La corbeille du dossier home se trouve dans le dossier **~/.local/share/Trash/files**

```
rm -r -f ~/.local/share/Trash/files/*
```

Remarque : Il est possible de mettre **sudo** devant, en cas de réticence du système à supprimer. Mais il faut bien faire attention, on peut facilement faire des bêtises avec un **sudo** mal placé.

1.8 La stégano pour les nuls

La stéganographie est un procédé qui permet de cacher un message « secret » dans un autre message sans éveiller les soupçons d'autrui. L'avantage de la stégano sur la crypto et qu'avec la crypto tout le

monde voit que c'est crypté, ce qui va encore plus attirer l'œil. Si on cache le message crypté dans une image anodine cela devient parfait.

Pour cela on peut jouer sur une propriété intéressante des fichiers ZIP. Leur « header » se trouve à la fin du fichier. Ainsi si on ouvre un **.zip** il va commencer par aller lire la fin du fichier, regarder la taille qu'indique le header et ignorer tout ce qui se trouve avant. La plus part des autres formats fonctionnent à l'opposé : le header est au début et tout ce qui se trouve après est ignoré. C'est le cas des **jpeg**, **png** et **gif**. La solution est donc simple :

pour « cacher » : `cat image_anodine.png secret_data.zip > image_avec_code.png`
 si vous « lancez » l'image, vous verrez votre bête image et rien d'autre.
 Pour récupérer vos données, faites simplement : `unzip image_avec_code.png`

Remarque : Veillez à ce que la taille de vos données reste assez petite par rapport à la taille de l'image. Car une image de 300x200 pixels qui fait 10 Mio ça passe pas inaperçu.

1.9 Raccourcis

Ctrl + Alt + Del	quitte le serveur X (il faut rester appuyé sur Ctrl + Alt et appuyer deux fois successivement sur Del)
Ctrl + Alt + F2	Passes en mode console (donc plus de serveur X)
Ctrl + Alt + F7	Repasses en mode graphique quand on est en mode console.
Alt + F2	Affiche une fenêtre pour lancer un programme soit préenregistré soit personnel. D'une manière générale, ça permet de lancer une commande.

1.9.1 Les raccourcis standards sous gnome

Si vous allez ensuite faire un tour dans **Système/Préférences/Raccourcis clavier**, vous allez pouvoir modifier le comportement standard du clavier Gnome. La manipulation est très simple, par exemple pour ouvrir une boîte de dialogue de lancement d'exécutable sur la combinaison **Ctrl** + **Alt** + **R**, il suffit d'aller sur la bonne ligne, de cliquer sur la colonne de droite, et de presser les touches correspondantes.

En revanche, cela se corse pour la fameuse touche **Win**. Si vous faites la même manipulation, seul le premier appui sur cette touche est mémorisé. Du coup, si vous vouliez que ce soit finalement **Win** + **R** qui lance votre boîte d'exécution, elle va être lancée dès que vous appuyez sur **Win**... La solution est donc :

1. Sélectionnez la ligne voulue (ex. Lancer une application).
2. Cliquez sur la colonne de gauche.
3. Pressez la touche **Win**, ne relâchez pas.
4. Cliquez une seconde fois sur la colonne de gauche.
5. Pressez la seconde touche (ici **e**), relâchez!!

1.10 Rajouter des polices de caractères

On peut les rajouter pour tout les utilisateurs, mais l'inconvénient majeur, pour moi qui réinstalle relativement souvent mon système, c'est qu'il faut le refaire à chaque fois. Un moyen de contourner, c'est de mettre les fichiers au format **.ttf** dans le dossier **.fonts** du répertoire utilisateur (à savoir /home/login)

Remarque : S'il n'existe pas, créez le.

1.11 Sélectionner des fichiers avec motif (dans Nautilus)

en rentrant des expressions du style ***.gif** on sélectionne tout les fichiers avec l'extension **.gif**

- **"ac*"** : Tous les fichiers qui commencent avec "ac"
- **"a*.txt"** : Tous les fichiers qui commencent avec "a" et qui finissent avec ".txt"
- **"*a*x"** : Tous les fichiers qui contiennent le caractère "a" n'importe où, et qui finissent avec "x"

1.12 Réaliser des opérations sur un fichier .odt à la main

Un fichier .odt est simplement un fichier .zip avec des choses dedans. À partir de là il est possible de modifier manuellement le fichier.

Un exemple : Je voulais redimensionner toutes les images d'un énorme fichier .odt parce que ce n'avait pas été fait. La solution est de

1. renommer le fichier *toto.odt* en *toto.zip* puis le décompresser.
2. dans le dossier *toto*, cherchez le dossier *Pictures*.
3. Sélectionnez toutes les images trop grosses, puis clic droit *Redimensionner les images*. Dans la fenêtre qui s'ouvre, sélectionnez le format voulu (j'ai pris 800x600) et n'oubliez pas de cocher *la case Ne pas changer le nom*.
4. Une fois fait, il faut recompresser. Pour ça, attention, il ne faut pas compresser le dossier, mais bien les fichiers à l'intérieur. Pour faire plus clair, placez vous DANS le dossier *toto*, sélectionnez tout, et compressez en .zip.
5. Renommez le .zip en .odt et ouvrez le, normalement c'est bon.

2 Configuration du système

2.1 Restaurer un système RAID1 « dégradé »

Dans mon cas, /dev/sda est un SSD, et les deux partitions (une seule partition par disque) utilisés pour le RAID sont /dev/sdb1 et /dev/sdc1.

On vérifie déjà l'état du RAID via la commande :

```
sudo mdadm --query --detail /dev/md0
```

Le retour de la commande sera quelque chose du style :

```
/dev/md0:
  Version : 1.2
  Creation Time : Sun Nov  3 22:07:32 2013
    Raid Level : raid1
    Array Size : 1465006336 (1397.14 GiB 1500.17 GB)
  Used Dev Size : 1465006336 (1397.14 GiB 1500.17 GB)
    Raid Devices : 2
  Total Devices : 1
 Persistence : Superblock is persistent

Update Time : Sun Nov 10 16:55:56 2013
  State : clean, degraded
Active Devices : 1
Working Devices : 1
Failed Devices : 0
Spare Devices : 0

   Name : autiwa-SSD:0
   UUID : 2eee2afa:c66bc1b4:83fcfd73:b652df7c
 Events : 118

Number   Major   Minor   RaidDevice State
   0         0         0         0    removed
   1         8        33         1    active sync    /dev/sdc1
```

On constate qu'un seul des disques est dans le RAID1, et que ce dernier est dégradé, c'est à dire qu'il n'y a plus qu'un seul disque qui fonctionne.



À partir de là, je considère que vous avez remplacé le disque défectueux par un autre disque s'il y a lieu afin de pouvoir remettre le système en état. Notez que durant cette période, plus rien ne vous protège contre la perte de données si le 2e disque dur crashe.

Au cas où ce ne soit pas déjà effectué de manière automatique, on peut déclarer un disque du système RAID comme défectueux (si la partition a déjà été supprimée, on peut passer cette étape; c'était mon cas) :

```
sudo mdadm /dev/md0 --fail /dev/sdb1 --remove /dev/sdb1
```

Remarque : Avant de s'inquiéter de la récupération, il se pourrait que vous deviez formater le nouveau disque pour être parfaitement identique au premier disque encore présent dans le système RAID.

Ensuite, pour remettre le système RAID en état, il suffit de rajouter de nouveau la partition qui nous intéresse dans le système :

```
$ sudo mdadm /dev/md0 --add /dev/sdb1
mdadm: added /dev/sdb1
```

À partir de là, le système RAID va petit à petit se remettre en état. Comptez environ 6h pour remettre en état un RAID 1 d'1.5Tio.

On peut surveiller l'avancement soit par :

```
$ sudo mdadm --query --detail /dev/md0
/dev/md0:
    Version : 1.2
  Creation Time : Sun Nov 3 22:07:32 2013
    Raid Level : raid1
    Array Size : 1465006336 (1397.14 GiB 1500.17 GB)
  Used Dev Size : 1465006336 (1397.14 GiB 1500.17 GB)
    Raid Devices : 2
  Total Devices : 2
 Persistence : Superblock is persistent

 Update Time : Sun Nov 10 17:15:50 2013
   State : clean, degraded, recovering
 Active Devices : 1
Working Devices : 2
 Failed Devices : 0
  Spare Devices : 1

Rebuild Status : 1% complete

    Name : autiwa-SSD:0
   UUID : 2eee2afa:c66bc1b4:83fcfd73:b652df7c
 Events : 218
```

Number	Major	Minor	RaidDevice	State	
2	8	17	0	spare rebuilding	/dev/sdb1
1	8	33	1	active sync	/dev/sdc1

ou à l'aide de la commande suivante, très pratique soit dit en passant :

```
watch -n3 cat /proc/mdstat
```

2.2 Régénérer le menu de multiboot de Grub

`sudo update-grub` génère un nouveau fichier `/boot/grub/menu.lst`.

2.3 Quelques commandes pour débloquer le système

Ce petit guide explique quelques combinaisons de touches à utiliser lorsqu'un système Linux est complètement bloqué.

La plupart des commandes citées ci-dessous utilisent la touche Magic SysRq Key. Cette dernière correspond à la touche Impr écran de votre clavier :

2.3.1 Activation / Désactivation de SysRq

Il faudra absolument que le noyau de votre Linux soit compilé avec l'option "Magic SysRq Key" pour que cette touche fonctionne. La plupart des distributions récentes font cela par défaut aujourd'hui.

On pourra ensuite vérifier l'activation de cette touche via la commande suivante :

```
# cat /proc/sys/kernel/sysrq
0
```

Ici, zéro signifie que cette dernière n'a pas été activée¹. On pourra changer dynamiquement cette valeur comme ceci :

```
# echo "1" > /proc/sys/kernel/sysrq
```

Si l'on veut activer l'utilisation du SysRq automatiquement au démarrage de la machine, on pourra éditer le fichier `/etc/sysctl.conf` et mettre l'option `sysrq` à 1 :

```
kernel.sysrq = 1
```

2.3.2 Les commandes

[Alt] + [SysRq] + [r] : met le clavier en mode 'raw'. Ceci est pratique notamment lorsque le clavier et la souris sont complètement bloqués. En passant sur ce mode, on peut de nouveau utiliser les touches pour saisir des commandes.

[Ctrl] + [Alt] + [F1] à [F7] : permet de naviguer à travers les consoles en mode texte (appelées aussi tty)

[Ctrl] + [Alt] + [Backspace] : arrête l'interface graphique, on pourra la relancer ensuite avec la commande `startx`

[Alt] + [SysRq] + [s] : écrit toutes les données en cours sur le disque dur afin d'éviter les pertes ou les corruptions de fichiers

[Alt] + [SysRq] + [u] : permet de remonter tous les systèmes de fichiers en lecture seule pour éviter la corruption de données

[Alt] + [SysRq] + [e] : tue tous les processus normalement (le processus `init` n'est pas concerné)

[Alt] + [SysRq] + [i] : force le kill de tous les processus (le processus `init` n'est pas concerné)

[Alt] + [SysRq] + [l] : même chose que la précédente, sauf que le processus `init` est concerné ici

[Alt] + [SysRq] + [b] : redémarre le système

[Alt] + [SysRq] + [o] : arrête le système

3 Équivalents windows

En fait, rigoureusement, j'aurais pu nommer la section autrement. Vu que je vais regrouper les programmes par catégorie et dire lesquels je préfère. Mais ça sera plus utiles pour ceux qui cherchent des équivalents windows que pour les autres. **La liste ne concerne que mes choix, je ne parle pas de tout, le meilleur moyen de trouver ses préférences est de tester soi même et de choisir en fonction de ses propres exigences**

3.1 Calcul formel : Xcas

3.1.1 Xcas

Maple est un logiciel de calcul formel qui n'est plus à présenter, mais malheureusement payant (et peut-être non disponible sous GNU/Linux, en tout cas je ne me suis pas posé la question). **Xcas** semble être une sorte d'alternative libre et disponible notamment sous GNU/Linux. Pour l'installer, on peut ajouter un dépôt :

```
deb http://www-fourier.ujf-grenoble.fr/~parisse/debian/ stable main
```

1. dans la plus part des distributions, l'utilisation de la touche SysRq n'est pas activée par défaut pour des raisons de sécurité. En effet, quelqu'un qui aurait un accès physique à la machine, pourrait facilement la redémarrer via une combinaison de touches et compromettre cette dernière ensuite.

puis taper la commande dans une console :

```
sudo apt-get update
sudo apt-get install giac
```

source : http://www-fourier.ujf-grenoble.fr/~parisse/install_fr#packages

3.1.2 Wxmaxima

C'est le logiciel que j'utilise pour l'instant, j'ai mis un peu de temps à comprendre, vu qu'à force d'utiliser 50 logiciels différents pour les mêmes fonctions, on finit par confondre les syntaxes. Voici quelques exemples de bases d'utilisation, que ce soit affectation de variables, utilisation de variable, ou tracé de courbe 2D et 3D.

Constantes prédéfinies Il existe dans maxima des constantes prédéfinies. Elles se distinguent des autres variables par le fait qu'on les utilise avec le signe « % » devant.

π	%pi
e (nombre de néper)	%e
i tel que $i^2 = -1$	%i

Définir une variable et l'utiliser Pour définir une variable x , il suffit de taper :

```
x:3.5e-9
```

Remarque : Par exemple, si ici x représente pour nous une longueur, on aurait donc x qui vaut 3,5 nm.

Pour utiliser x dans une expression, il suffit de rentrer :

```
cos(x*%pi)
```

Remarque : Ici, %pi est la variable π . Certaines constantes sont directement définies dans maxima, voir partie (3.1.2).

Définir une fonction Pour définir $f(x)$:

```
f(x):=x^2
```

Pour tracer une courbe en 2D Voici un exemple de commandes à taper pour avoir une courbe en 2D :

```
plot2d (sin(x), [x, -5, 5])
plot2d (sec(x), [x, -2, 2], [y, -20, 20], [nticks, 200])
```

Pour tracer une courbe en 3D Voici un exemple de commandes à taper pour avoir une courbe en 3D :

```
expr_1: cos(y)*(10.0+6*cos(x));
expr_2: sin(y)*(10.0+6*cos(x));
expr_3: -6*sin(x);
plot3d ([expr_1, expr_2, expr_3], [x, 0, 2*%pi],
[y, 0, 2*%pi], ['grid, 40, 40]);
```

Pour enregistrer cette courbe dans un fichier, il faut taper :

```
expr_1: cos(y)*(10.0+6*cos(x));
expr_2: sin(y)*(10.0+6*cos(x));
expr_3: -6*sin(x);
plot3d ([expr_1, expr_2, expr_3], [x, 0, 2*%pi],
[y, 0, 2*%pi], ['grid, 40, 40], [gnuplot_term, ps],
[gnuplot_out_file, "/home/autiwa/plot3d.eps"]);
```

Remarque : J'ai essayé de tracer directement dans un `.svg`, mais il semble que le `.svg` soit corrompu. À confirmer donc.

J'avais utilisé les options suivantes :

```
[gnuplot_term, svg], [gnuplot_out_file, "/home/autiwa/plot3d.svg"]
```

3.1.3 Octave/scilab

Voir la section [§ 3.15 page 19]

3.2 Client Mail : Evolution

Thunderbird me plaît beaucoup. Celà dit, tout comme firefox, certaines extensions ne marchent pas sous GNU/Linux, notamment *minimize to tray*. Il existe aussi *evolution* et après l'avoir testé, je le trouve beaucoup mieux intégré à gnome que thunderbird. Je le conseille sous gnome. Mais je suis passé récemment à *KDE*, et j'ai préféré prendre un client mail KDE (*Kmail*) pour une question de rapidité et d'intégration, et depuis quelques temps, j'en viens presque à le préférer qu'*evolution*. Finalement repassé sous gnome, ayant été déçu par KDE que je trouve trop compliqué et fouilli, et buguant (surtout KDE4 à sa sortie stable), je reprends evolution.

Remarque : Pour faire une sauvegarde complète vous devez copier ces répertoires :

```
— ~/.evolution/
— ~/.gconf/apps/evolution/
— ~/.gnome2_private/Evolution
```

3.3 Comptabilité : Grisbi

J'utilisais *Money* dans le temps sous windows, puis un logiciel libre dont j'ai oublié le nom, et quand je suis passé sous GNU/Linux, j'ai parcouru les logiciels disponibles, et j'ai opté pour *Grisbi*. Je l'avais essayé sous windows dans le temps et il ne m'avait pas plu du tout. Sous GNU/Linux, le courage et le sens de l'effort sous le bras, je m'y suis mis plus sérieusement, et j'ai bien fait, passé une semaine pour comprendre comment ça fonctionne et que sans faire les rapprochement tout les mois avec les relevés, le logiciel n'est pratiquement pas utilisé, je suis content d'avoir insisté.

3.4 Dessin Vectoriel : Inkscape

J'utilisais pas de logiciel de dessin vectoriel sous windows, mais petit à petit, pour les besoins de \LaTeX , j'ai voulu faire des schémas propres, et je me suis mis à *Inkscape* pour exporter ensuite mes schémas en `.pdf`.

Ses possibilités sont nombreuses et j'ai même dû faire un tutoriel à part tellement j'en avais rajouté. Je vous le conseille chaudement, surtout si vous avez souvent besoin de faire des schémas ou d'en modifier.

3.5 Digitizer : Engauge

Logiciel permettant de digitaliser des courbes (utiles pour les études scientifiques parfois. Pour l'installer :

```
sudo apt-get install engauge-digitizer
```

3.6 Éditeur \LaTeX : Kile

Pour \LaTeX j'utilise *Kile* qui est celui qui me plaît le plus de tout ceux que j'ai essayé (il existe *TeXmaker* que j'utilisais sous windows).

3.7 Éditeur de Texte : Kate

Il en existe vraiment énormément, il y en a pour tous les goûts, toutes les utilisations et je ne vais pas tous les énumérer. Certains ne jurent que par *emacs*. Un éditeur pratique pour moi est *gedit* mais pour mes textes, j'ai retenu *Kate* que je trouve excellent, bourré de fonctionnalités, très pratique et très complet.

Kwrite est aussi très bien, mais souffre de l'inconvénient majeur de ne pas gérer les onglets, on aura donc autant de fenêtres que de fichiers, ce qui est tout sauf pratique je trouve.

geany est aussi pas mal. C'est même celui que j'utilise actuellement, très pratique pour du code, notamment grâce à l'affichage des fonctions, classes et variables dans une structure sur la gauche. En maintenant **Ctrl** appuyé et en sélectionnant, on peut sélectionner des colonnes de caractère, comme dans *emacs*.

gedit est trop simple à mon goût. Pas assez de fonctionnalités (il n'y a qu'à comparer le nombre de choses que l'on peut faire avec *kate*, comme fusionner les lignes, modifier la casse et cie).

3.8 Encodeur Vidéo : Avidemux

Sous windows, c'était *virtual dub*, ici j'ai testé *avidemux* et je le trouve très simple à utiliser.

Pour réencoder une vidéo, il suffit de l'ouvrir. Ensuite, pour ma part, je prend le codec vidéo x264 (qualité à 22²) et le codec audio AAC. Puis, ça dépend de ma vidéo, mais maintenant, j'aime bien sélectionner une taille préférentielle, et en double passe.

Puis j'ajoute des filtres. Le premier, **crop** (que l'on trouve dans la rubrique « transform » pour rogner s'il y a des bandes noires ou des logos qui dépassent.

Ensuite, dans la rubrique « noise », je prend le filtre **hqdn3d** et je laisse les valeurs par défauts.

Puis dans la rubrique « sharpness », je prend le filtre **sharpn**.

Et enfin, si la vidéo est en noir et blanc, je prend dans la rubrique « couleur » le filtre **luma only** qui encode en niveau de gris.

Si j'encode un anime, je n'utilise pas le même filtre. À la place de **hqdn3d**, j'utilise **MSmooth** par **Donald Graft** qui est fait exprès pour les anime, et je le trouve très efficace.

3.8.1 Options de montage et de rognage

J'utilisais depuis longtemps les fonctions A et B qui permettent de sélectionner le début et la fin de la vidéo à encoder. En gros, j'étais persuadé qu'on ne pouvait rogner que le début ou la fin de la vidéo ajoutée.

En fait, c'est beaucoup plus général que ça. On sélectionne un passage en prenant le début et la fin grâce à A et B. Une fois fait, on peut, soit encoder juste ce passage en sélectionnant « enregistrer la vidéo », soit on peut couper, copier, coller ou supprimer la sélection. Ainsi, on peut modifier beaucoup plus que ce que je croyais.

Pour rogner certains passages au milieu de la vidéo, on sélectionne donc grâce à A et B uniquement le passage qu'on veut virer, puis on fait Edition>Supprimer.

3.8.2 Encoder un DVD

À partir d'un .vob préalablement extrait, on peut encoder un DVD, mais il y a pour la plupart d'entre eux une manipulation à effectuer. En effet, la forme des pixels n'est pas forcément standard. Je sais que je vais mal l'expliquer mais en gros, il faut redimensionner la vidéo pour que le ratio à l'écran soit correct.

Pour cela, il faut sélectionner le filtre vidéo **Mplayer Resize**. Sélectionner *source* 16 :9 et *destination* 1 :1 puis cocher l'option **16 round up** (arrondir le nombre de pixels à un multiple de 16). Puis sur cliquez sur le curseur qui permet de sélectionner le ratio, déplacez le un peu à gauche et revenez tout à droite. Vous aurez ainsi la taille en pixel maximum possible et respectant les conditions qu'on vient d'imposer. Pour ma vidéo qui est en 720*576, la vidéo finale devient en 720*400

2. Si je met la qualité à 20, le fichier prendra plus de place et la qualité sera meilleure. Inversement, la qualité par défaut, 26, est moins bonne que la qualité que je prend moi

3.9 Gestionnaire de photo : Digikam

J'ai pas trouvé mieux. Pourtant, je crois que je les ai quasiment tous essayés.

Pour renommer mes photos, j'applique le schéma suivant :

```
%Y-%m-%d_%H:%M:%S.[e-]
```

je coche "ajouter la date", "date format" et enfin "use extra symbol".

3.10 IDL : GDL

L'équivalent libre sous GNU/Linux d'**IDL**, c'est **GDL**. Une petite recherche sur internet vous en dira plus. Ce que par contre j'ai mis du temps à comprendre, c'est qu'on peut l'installer automatiquement de la façon suivante :

```
sudo apt-get install gnudatalanguage
```

Ensuite, il faut définir les variables d'environnements de GDL. J'ai mis du temps à comprendre et trouver comment il fallait faire, ma méthode n'est en aucun cas la meilleure vu qu'apparemment je définit à chaque démarrage d'une console les variables au lieu qu'elles soient créées une bonne fois pour toute, mais ça a le mérite de marcher.

J'ai donc créé un répertoire `.gdl` dans mon répertoire utilisateur. Dans ce répertoire, j'ai créé un fichier startup qui me sert à définir les commandes à exécuter au démarrage de GDL. Pour le moment, le fichier est vide, mais quand je saurais quoi mettre dedans, je pourrais le faire facilement (peut-être qu'il faut les droits d'exécutions au fichier).

J'ai créé dans ce dossier deux autres dossiers, `lib` et `routines` dans lesquels je stocke les `.pro` que j'ai récupéré ici et là et qui me permettent de compléter mon utilisation de GDL.

Maintenant, pour créer les variables d'environnements, j'ai rajouté les lignes suivantes à mon fichier `/.bashrc` :

```
##### Variables d'environnements pour GDL #####
export GDL_PATH=~/.gdl/lib :~/.gdl/routines
export GDL_STARTUP=~/.gdl/startup
```

J'ai aussi eu besoin d'installer un paquet de plus pour pouvoir utiliser un plot dans une console. Sinon, je suis obligé de me limiter à l'exportation en `.ps`, c'est un peu réducteur comme utilisation.

```
sudo apt-get install plplot9-driver-xwin python-numarray python-matplotlib
```

3.10.1 fonctions mathématiques standard

nom : fonction
alog : logarithme népérien (ou naturel)
alog10 : logarithme décimal
exp : exponentiel
factorial : factoriel ($n!$)
abs : valeur absolue
sqrt : racine carrée (pour l'anglais square root)
cos : cosinus (attention, tous les angles sont en radians : $p \text{ radians} = 180^\circ$)
sin : sinus
tan : tangente (rappelons que $\cotangente = 1/\text{tangente}$)
acos : arc cosinus (fonction inverse de cosinus)
asin : arc sinus (fonction inverse de sinus)
atan : arc tangente (fonction inverse de tangente)
sinh : sinus hyperbolique ($\sinh(x) = [\exp(x) - \exp(-x)]/2$)
cosh : cosinus hyperbolique ($\cosh(x) = [\exp(x) + \exp(-x)]/2$)
tanh : tangente hyperbolique ($\tanh(x) = \sinh(x)/\cosh(x)$)

3.10.2 Les tableaux

Il existe plusieurs types de tableaux, celui que j'utilise le plus est :

```
a=dblarr(12,3)
b=dblarr(10)
```

qui définit a comme un tableau de 12 colonnes et 3 lignes et b comme une liste de 10 éléments, les éléments de ces deux tableaux étant des réels double précision.



La numérotation commence à 0, ainsi, $b[0]$ est le premier élément de b , et $b[9]$ est le dernier élément

Il existe un moyen très pratique de générer des listes, notamment quand on veut tracer des courbes, c'est à l'aide de la fonction suivante :

```
f=indgen(10)
dx=1.d-2
x=dx*indgen(10)
```

Ceci définit f comme une liste contenant les entiers de 0 à 9, et x comme la même chose, sauf qu'on a défini un pas, dx qui nous permet de resserrer les points. On peut ainsi rajouter une valeur minimale, faire une échelle logarithmique et plein de choses très pratique.

Remarque : À noter que si on exécute :

```
a=0.1*!dpi*indgen(10)
b=sin(a)
```

alors on va faire le sinus de chaque élément de a et le stocker dans b qui sera un tableau de même taille que a . $!dpi$ est la variable qui contient la valeur de π en réel double précision.

Il existe des commandes spéciales pour manipuler des tableaux, qu'il ne faut surtout pas reprogrammer soit même, vu que le temps de calcul serait plus long :

```
total(a)
max(a)
min(a)
n_elements(a)
size(a)
```

qui définissent respectivement la somme, le maximum, le minimum, le nombre d'éléments et les dimensions (je ne connais pas le détail de cette dernière commande, vu que je ne l'utilise jamais). À noter que ce sont des fonctions, donc on les définit de la façon suivante :

```
somme=total(a)
```

Remarque : On peut aussi récupérer l'indice de l'élément le plus grand ou le plus petit du tableau. Pour cela, il suffit d'écrire :

```
energie_max=max(energie,indice_max)
energie_min=min(energie,indice_min)
print,energie[indice_max],energie[indice_min]
```

Dans le cas présent, ce n'est pas très utile, mais ça permet, dans d'autres cas, quand on a plusieurs tableaux, de récupérer cet indice, et d'afficher les autres caractéristiques correspondant à ce même indice, c'est à dire rayon, masse et vitesse d'un objet dont l'énergie est maximale, par exemple.

La commande `sort(a)` permet de sortir une liste d'indice qui permet de trier la liste dans l'ordre croissant. Ainsi

```
print,a[sort(a)]
```

affiche la liste a avec les éléments triés.

3.10.3 les boucles

for

```
for i=0,n_elements(a)-1 do begin
  a[i]=3*i+1
endfor
```

if

```
if (a lt b) then begin
  print,"a est inférieur à b"
endif
```

ou encore, si on veut faire un else :

```
if (a lt b) then begin
  print,"a est inférieur à b"
endif else begin
  print,"a est supérieur ou égal à b"
endelse
```

On a comme ça plusieurs paramètres de test :

Commande : relation

lt : inférieur strictement

le : inférieur ou égal

gt : supérieur strictement

ge : supérieur ou égal

eq : égal

ne : différent

and : permet de faire deux conditions. Par exemple (a lt 5 and a gt 4)

where

```
a=indgen(10)
ind=where(a lt 5,count)
print,a[ind]
```

grâce à la commande **where**, on stocke dans **ind** les indices des valeurs de *a* qui vérifient $a < 5$. Ça permet de sélectionner rapidement et efficacement certaines données pour les isoler.

count, quant à lui, est une variable qui va stocker le nombre d'éléments vérifiant la condition que l'on a donné. En particulier, ça permet de ne faire des opérations que si **count** est différent de 0.

3.10.4 Les variables aléatoires

Il existe au moins deux types de variables aléatoires implémentées dans GDL (i) distribution uniforme (ii) distribution normale que l'on appelle respectivement par :

```
a=randomu(seed,10)
b=randomn(seed,10)
```

où 10 est le nombre de valeurs que l'on veut, et **seed** le nombre qui sert à générer la séquence aléatoire. On peut tout à fait (et c'est conseillé si on veut des résultats reproductible pour vérification) spécifier la valeur de **seed**, du style :

```
a=randomu(182751824562,10)
```

Remarque : Pour un **seed** donné, la séquence de nombre aléatoire générée est toujours la même.

Pour éviter les corrélations entre les générations de nombres aléatoires, il faut en fait utiliser le même **seed** quand on fait plusieurs tirages, et les nombres générés sont à la suite (quand les générations se font au sein du même programme, sans réinitialisation du **seed**).

Exemple :

```
seed=1001L
print,randomu(seed,10)
print,randomu(seed,10)
```

les deux suites de nombres seront différentes.

```
seed=1001L
print,randomu(seed,10)
seed=1001L
print,randomu(seed,10)
```

les deux suites de nombres seront les mêmes.

3.10.5 Astuces

- En tout cas dans GDL, je ne sais pas si c'est le cas pour IDL, on n'appelle pas le nom du fichier, mais du programme. C'est à dire le nom qu'il y a dans la première ligne de définition, du style :

```
pro transit
```

le nom du programme est `transit`.

- Pour appeler une commande du shell, du style pour lister les fichiers, il faut utiliser un \$ devant la commande. Ainsi, pour lister les fichiers, on utilisera la commande :

```
GDL> $ls
back      cooling.pro~  energy.pro~  lhb.pro~    tutos      warm.pro~
cooling.pro  energy.pro    lhb.pro    LHB_temp_ocean.ps  warm.pro
```

3.10.6 Erreurs

Il refusait catégoriquement de lire mon fichier, quoi que je fasse. Après plusieurs heures de recherches éfrénées, il s'est avéré que ça venait du fait que la fin de ligne de mon fichier était une fin de ligne mac. Après avoir changé et mis une fin de ligne Unix/windows, tout est rentré dans l'ordre.

3.11 Lecteur de musique : Quod Libet

J'en ai essayé vraiment beaucoup, j'utilisais *iTunes* sous windows et je voulais à peu près un équivalent, je n'en ai pas trouvé. Je voulais retrouver la possibilité d'avoir des listes de lecture intelligentes, de pouvoir modifier dans les propriétés d'une chanson la position du début et de la fin de celle ci, je n'ai pas retrouvé. J'ai essayé *Amarok*, *Song Bird* et bien d'autres. *Song Bird* n'est pas encore assez abouti à mon goût même si il se rapprochait beaucoup d'*iTunes*. J'ai finalement opté pour *Quod Libet*. Je ne fais pas de liste de lecture avec lui car je suis obligé de changer d'affichage, mais l'affichage de la bibliothèque est tellement personnalisable que je n'en ai plus besoin.

3.12 Lecteur PDF : Okular

Remarque : Anciennement, Okular s'appelait Kpdf.

Il existe un visualiseur par défaut des fichiers PDF, mais je trouvais que le rendu était vraiment moche, ok il affichait tout les fichiers, tels que PDF, PS et cie, mais pour les PDF, les caractères étaient très mal affichés je trouvais, donc j'ai cherché un remplaçant et j'ai opté pour *Okular* qui affiche très bien les PDF et me convient tout à fait, pour moi l'équivalent de *Foxit Reader* que j'utilisais sous windows.

Petite astuce, on peut ajouter dans la barre d'outils les très pratiques « Back in the document » et « Forward in the document ». Pourquoi c'est pratique ?

Prenons un exemple : Je clique sur un lien interne qui m'amène à une équation sur une autre page du document. Je clique sur « Back in the document » et ça me ramène à la page où j'avais cliqué sur le lien vers l'équation, ainsi je reviens à mon calcul du départ sans devoir m'embêter à retrouver la page.

3.13 Lecteur Vidéo : SMPlayer

J'ai mis beaucoup de temps à en trouver un. On m'a conseillé *Totem* que j'ai conservé un moment, mais pour certaines vidéos, le rendu n'est pas top. *MPlayer* n'est pas pratique je trouve, tout comme *VLC*. Puis j'ai essayé *Kaffeine* au début, je n'arrivait carrément pas à lire de vidéos, mais après quelques manip via un tutoriel, ça marchait et j'aime bien. ça se rapproche un peu de *Window Media Player*. Mais en dernier lieu, celui que je préfère c'est *SMPlayer* un lecteur basé sur *MPlayer* mais avec une interface graphique plus pratique. Autant le dire aussi, je n'ai pas trouvé d'équivalent à *Windows Media Player* avec la liste de lecture sur le côté et un bon rendu des vidéos. En clair, je jongle entre *Kaffeine* et *SMPlayer*.

Étant passé sous KDE, j'ai voulu essayé *KMPlayer*, mais je le trouve bizarre, et moins pratique qu'*SMPlayer*.

3.14 Manipulateur de Fichiers : Métamorphose

Sous windows, j'utilisais le couple *Antrenamer* et *mp3 tag* pour renommer mes fichiers et mes musiques. Sous GNU/Linux j'ai eu le besoin d'avoir la même chose. Il en existe plusieurs, mais j'ai réussi à en faire fonctionner un seul, nommé *Métamorphose*. Je l'avais essayé sous windows et il ne m'avait pas plu, sous GNU/Linux, je l'ai réessayé et j'ai appris à l'aimer. Il permet de renommer à la fois des fichiers et des dossiers, possède des fonctions très complètes même si j'ai mis du temps à comprendre qu'il fallait non seulement les configurer via par exemple l'onglet **numérotation** mais ensuite il fallait les appeler en tant que variable dans le nom à renommer via par exemple **:numr** : pour la numérotation. Il ne lui manque pour moi qu'une seule fonction, celle de pouvoir renommer les fichiers à partir d'une liste chose que j'appréciais avec mes précédents programmes (par exemple la liste des noms de chansons d'un album, et vous renommez d'un coup les 10 chansons nommées track 1 à 10 de la bonne façon.

Sous KDE, il existe *KRename* qui a l'air pas mal, avec une système étape par étape pour les débutants : À tester donc.

3.15 Matlab : Octave

Il existe deux logiciels à ma connaissance qui sont susceptible de remplacer *matlab* : *Octave* et *Scilab*. La version que j'ai testé de scilab ne m'a pas plu du tout, sans doute en GTK2. De plus, la syntaxe scilab n'est pas la même que celle de matlab, ne serait-ce que pour le caractère permettant d'écrire des commentaires, et j'ai trouvé ça un peu osé que pour un programme censé être un « équivalent » de matlab, il n'aie même pas la même manière d'écrire les commentaires.

Bref, j'ai rapidement laissé tomber scilab, et je me suis tourné avec octave qui lui s'évertue beaucoup plus à être compatible avec matlab. Par défaut, c'est un programme en ligne de commande, mais il existe un front-end, et il existe pléthore de paquets pour rajouter des fonctions, et à première vue, ça a l'air assez complet. Pour installer le minimum, je conseille de faire :

```
sudo apt-get install octave octave-general qt octave
```

3.15.1 Ajouter des fichiers .m dans octave

Pour celà, il suffit, dans une session octave, de taper les commandes suivantes :

```
addpath(genpath("~/octave/"));
savepath;
```

où `~/octave/` est le chemin absolu du dossier où vous souhaitez stocker vos fichiers **.m**.

3.15.2 Indexage logique

Il existe une troisième forme d'indexage basée sur les conditions logiques. On la désigne par indexage logique et on l'utilise principalement avec les opérateurs relationnels et les opérateurs logiques. Ce type d'indexage permet d'améliorer l'efficacité des codes en évitant l'utilisation de fonctions supplémentaires (principalement **find**).

L'indexage logique est souvent utilisé avec les fonctions **any** et **all**.

Par exemple, on souhaite trouver toutes les valeurs supérieures à 3 (soit 8 et 7) dans la matrice suivante :

```
>>X=[8 1 ; 2 7]
X =
8 1
2 7
```

La méthode classique consiste à utiliser **find** comme ceci :

```
>> X=[8 1 ; 2 7]
X =
8 1
2 7
>> idx=find(X>3) % Indexage linéaire
idx =
1
4
>> X(idx)
ans =
8
7
```

On remarque que `idx` correspond aux indices linéaires de $X > 3$

L'indexage logique consiste simplement à se passer de la fonction **find** :

```
>> X=[8 1 ; 2 7]
X =
8 1
2 7
>> idx = (X>3) % Indexage logique
idx =
1 0
0 1
>> X(idx)
ans =
8
7
```

On remarque maintenant que `idx` est une matrice contenant des valeurs logiques (0 ou 1).

J'avoue ne pas encore avoir bien saisi la différence entre les deux et surtout dans quels cas les utiliser, mais ça a l'air pratique, donc faut garder ça en mémoire.

3.15.3 Tracer un damier

```
l=20;b=zeros(1,1);
b(1:2:end, 1:2:end)=255;
b(2:2:end, 2:2:end)=255;
imshow(b);
```

3.15.4 Lettre grecques dans les labels

Il est possible d'afficher des lettres grecques et d'utiliser la syntaxe \LaTeX dans les labels et titres d'un plot. Sous matlab, on peut utiliser directement `\pi`, `\sigma` etc... Par contre, sous octave, il faut utiliser une syntaxe différente. Pour faire π il faut rentrer le code suivant : `{/Symbol p}`.

Ensuite, sous matlab comme sous octave, on peut mettre des indices et des exposants à l'aide de `^` et `_`. Il convient d'utiliser les groupes si on veut mettre plusieurs caractères en indices et exposants, exemple :

```
title("{/Symbol s} section efficace (cm^{-2})")
```

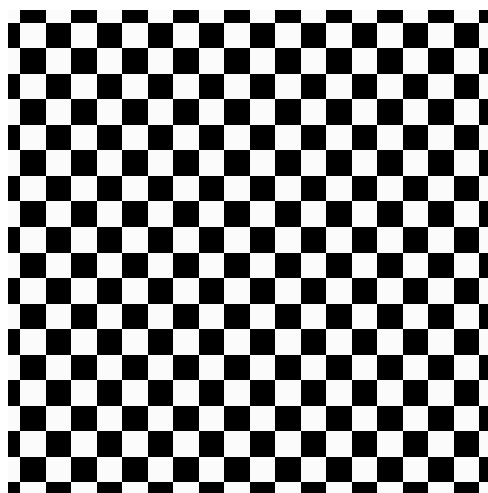


FIGURE 1 – Rendu du damier tracé par matlab

3.15.5 Changer les symboles et styles de ligne

Le type de ligne ou de symbole pour les plots peut être changé en ajoutant un troisième argument optionnel dans la commande plot. Par exemple :

```
>> plot(x,y,'o');
```

affiche les données dans les vecteurs x et y en utilisant des cercles dans la couleur par défaut (jaune) et

```
>> plot(x,y,'r:');
```

affiche les données dans les vecteurs x et y en reliant les points entre eux par une ligne pointillée rouge.

Le 3^e argument de la commande plot est une chaîne de caractères composé de un, deux ou trois symboles de la forme “cs” où “c” est un caractère unique indiquant la couleur, et “s” une chaîne de un ou deux caractères qui indique le type de symbole ou de ligne. Les couleurs et types de ligne ou symboles sont répertoriés dans le tableau [TABLE 1].

“c”	couleur du symbole ou de la ligne	“s”	type de ligne ou de symbole
y	jaune	.	point
m	magenta	o	cercle
c	cyan	x	croix
r	rouge	+	plus
g	vert	-	ligne continue
b	bleu	*	étoile
w	blanc	:	ligne pointillée
k	noir	-.	ligne (tiret-point)

TABLE 1 – Couleur, symbole et type de ligne pour les graphiques 2D

source : <http://web.cecs.pdx.edu/~gerry/MATLAB/plotting/symbolPlots.html>

3.15.6 Exporter des figures en .svg

Il suffit pour celà de rentrer la commande suivante :

```
plot(x,x)
title("premiere bissectrice")
grid on
axis on
print('-dsvg','figure.svg')
```

Remarque : Après ouverture du .svg, je me suis rendu compte que la grille et les axes n'apparaissaient pas. Pourtant ils sont bien là, mais ils sont transparents. Soit vous pouvez faire **Ctrl**+**A** et mettre un contour noir, soit le faire à la main.

3.15.7 subplot

```
x=0:0.01:2*pi;

figure(1)
subplot(2,2,1)
plot(x,sin(x))
title("sinus")

subplot(2,2,2)
plot(x,cos(x))
title("cosinus")

subplot(2,2,3)
plot(x,tan(x))
title("tangente")

subplot(2,2,4)
plot(x,x)
grid on
```

Pour cela, on a donc la fonction `subplot(L,C,N)` où L est le nombre de ligne, C le nombre de colonnes, et N le numéro de la figure qu'on s'apprête à afficher. En clair, Dans l'exemple ci-dessus, on a 2 lignes et 2 colonnes, ce qui nous permettra d'afficher au maximum 4 plots différents.



Il faut faire attention à mettre le même nombre de ligne et de colonnes pour les subplots d'une même figure, sous peine d'effacer certaines figures.

3.15.8 Options avancées de plot

`xlabel`, `ylabel` et `title` permettent de définir les titres des axes et du graphique.

On peut définir des graphiques en échelle log, ou semi-log en remplaçant `plot` par `semilogx`, `semilogy` ou `loglog`.

On peut afficher une grille et une sous-grille via les options suivantes

```
grid on
grid("minor")
```

Un exemple un peu plus élaboré :

```
figure(2)
subplot(2,2,1)
title("fluence en sortie du laser")
xlabel('temps (ns)')
ylabel('fluence J/cm^2')
hold off

subplot(2,2,2)
title('Energie en sortie de fibre laser')
xlabel('temps (ns)')
ylabel('Energie')
hold off

subplot(2,2,3)
plot(liste_longueur_fibre_1,dt)
```

```

title("evolution de la duree de l'impulsion")
xlabel('longueur de la fibre (cm)')
ylabel('duree (ns)')
grid on

subplot(2,2,4)
plot(lambda*1e9,fpompe)
title("spectre gaussien de la pompe")
xlabel("\lambda (nm)")
ylabel("intensite")
axis([970 980 1e7 3.2e8])
grid on
print('-deps','figure2.eps')

```

3.16 Navigateur Web : Firefox

Firefox Il reste mon préféré car hautement personnalisable.

Opéra existe sous GNU/Linux, une bonne alternative à firefox, même si j'ai à déplorer quelques plantages, notamment lors de l'affichage de la fenêtre **enregistrer sous** quand je clique sur un fichier à télécharger.

Epiphany est un navigateur très léger et bien intégré à gnome. Celà dit, je lui reproche, tout comme à opéra, de planter un peu trop facilement, chose qui sera peut-être réglée dans quelques temps.

Konqueror est LE navigateur de l'environnement KDE. Il est très réputé, et je compte le re-tester sous peu, vu qu'à mon premier test il y a de ça quelques mois, je n'avais pas du tout aimé.

3.16.1 Reprendre un téléchargement interrompu sous Firefox avec wget



Si vous supprimez le téléchargement dans la liste de firefox, ça supprimera les fichiers qu'il avait créé, donc il ne faut surtout pas le faire ; ou alors renommez le .part autrement ou copiez le ailleurs pour qu'il ne le supprime pas !

Renommez le fichier **.part** en le nom exact du fichier semi-téléchargé :

```
mv ubuntustudio.iso.part ubuntustudio.iso
```

Puis, utilisez *wget* avec l'option **-c** pour calculer la taille du fichier et reprendre le téléchargement interrompu :

```
wget -c http://www.truc.net/ubuntustudio.iso
```

(voir la section [§ 5.2.27 page 30] pour plus de détails

3.17 Retouche Photo : The Gimp

The Gimp apparaît comme la meilleure alternative sous GNU/Linux pour photoshop qui reste une référence sous windows. Seulement, il faut avouer que de ma propre expérience, je trouve *The Gimp* beaucoup moins intuitif que photoshop, et qui plus est moins puissant. Sur le traitement de CMJN ou des formats photos. *The Gimp* traite très mal les formats RAW je trouve, du moins celui qui me concerne, alors que photoshop les gère très bien. Il existe une version de Gimp, *gimpshop*, dont l'interface est copiée sur photoshop, mais je n'ai pas testé

3.17.1 Retouche d'Éléments gênants avec Gimp

Installation Cette astuce utilise le plugin *resynthesizer* the Gimp. Assurez-vous donc qu'il est installé avant d'aller plus loin. Avec Ubuntu, comme souvent, il suffit d'installer le paquet **gimp-resynthesizer** qui se trouve dans les dépôts *universe*.

```
sudo apt-get install gimp-resynthesizer
```

Utilisation Niveau utilisation, c'est assez simple. Il suffit de faire une sélection des éléments à faire disparaître avant d'appeler le script.

Pour faire la sélection plus facilement, nous allons utiliser un nouveau calque. Créez donc un nouveau calque. Mettez la transparence de celui-ci à 80% pour voir ce que vous faites.

Peignez alors avec une couleur les parties de l'image que vous voudriez voir disparaître.

Puis sur le calque, faites un clic droit, suivi de « Alpha vers sélection ». Sélectionnez le calque avec votre photo, puis rendez vous dans le menu **Script Fu > Enhance > Smart remove selection**, puis cliquez sur Suivant.

3.17.2 Brosse dynamique

Rien de plus ennuyeux que de devoir changer de taille de brosse toutes les 3 secondes. La solution à ce problème épineux consiste à aller dans les préférences, “Périphérique d'entrée > Contrôleur d'entrée”. Sélectionnez la molette de la souris (en anglais “main mouse wheel”) puis cliquez sur “configurer le contrôleur sélectionné” (une petite icône avec une clé d'outillage). Il ne reste plus qu'à assigner à **défilement haut (Maj)** et **défilement bas (Maj)** les actions **context-brush-radius-increase**, **context-brush-radius-decrease** ou des actions s'y apparentant.

Remarque : Pour ma part, j'ai une petite astuce. Je n'assigne pas les mêmes actions. C'est à dire que j'assigne **context-brush-radius-decrease-percent** et **context-brush-radius-increase**. Ainsi, la diminution du rayon de la brosse se fera en pourcentage du rayon actuel, ce qui a un effet énorme pour des rayons énormes, et un très petit effet pour des petits rayons. Ceci a pour conséquence qu'avec une énorme brosse, on lui fait reprendre une taille normale assez rapidement, et on est plus précis sur les petits rayons quand on diminue. À l'inverse, quand on va augmenter le rayon, l'augmentation sera toujours la même quelle que soit le rayon, ceci évitant un manque de précision dans la taille du rayon pour des gros rayons et une augmentation rapide du rayon quand celui-ci est tout petit (alors qu'en pourcentage, ça prend un temps fou)

3.17.3 Créer sa propre brosse

Pour cela, il suffit de créer une image (la brosse supporte le RVB et le niveau de gris mais pas l'indexation. Cela dit, si vous indexez, il vous proposera l'une ou l'autre des solutions) et de l'enregistrer sous le format .gbr dans le dossier qui contient les brosses. Ce dossier se situe dans le répertoire utilisateur. Pour moi c'est `/.gimp-2.6/brushes`.

3.18 Tableur : Gnumeric

Je dois avouer que le tableur d'*OpenOffice* m'a déçu. Vraiment en dessous d'excel pour moi. Ou de kaleidagraph (mais payant aussi celui là, donc pas un bon exemple. En clair, j'ai été agréablement surpris par *gnumeric* et il m'a bien aidé. Je n'ai pas encore tout testé, mais c'est le programme que je recommande pour faire des traitements de données sous GNU/Linux.

3.19 Visionneur D'Images : gThumb

J'utilise pour cela *gThumb* qui est relativement léger, permet de renommer les images en masse, et d'autres fonctionnalités pratiques. Il y en a bien sûr d'autres tels que *F-Spot*.

4 L^AT_EX sous GNU/Linux

4.1 Installation

Pour installer L^AT_EX sous GNU/Linux, il faut tout d'abord installer les commandes qui serviront à compiler vos code source. Pour cela, il y a *tetex*, et *tex-live* qui sont les deux principaux. Tetex n'étant plus mis à jour, je conseille tex-live. Vous pouvez l'installer via *synaptic*, ce qui vous permettra de compléter les 3 paquets que j'installe ici par des modules plus spécifiques.

Sinon, en ligne de commande, ceci devrait suffire à avoir ses premiers textes :

```
sudo apt-get install texlive texlive-lang-french texlive-latex-extra
```


Une fois fait, il faut maintenant un éditeur pour nous faciliter la vie. Il y en a pas mal, comme *TeXmaker* qui existe aussi sous windows. Il y a aussi *Winefish*, un éditeur L^AT_EX basé sur *Bluefish* qui est pas mal. Mais j'ai opté pour ma part pour *Kile* que je trouve très bien fait, très complet et que l'on peut installer comme ceci :

```
sudo apt-get install kile
```

4.2 Astuces

Pour mettre à jour la base de donnée des packages quand on a rajouté manuellement un **.sty**, il suffit de taper en console **sudo texhash**

À noter que le chemin où se trouvent les packages, et donc, là où on en rajoute manuellement est : `/usr/share/texmf-texlive/tex/latex/`

5 La ligne de commande

5.1 Astuces générales pour la console

5.1.1 Arrêter une commande en cours d'exécution

Pour arrêter la commande en cours du terminal ouvert, il suffit d'appuyer sur **Ctrl** + **C**

Pour quitter une page de manuel ou une documentation quelconque, il suffit bien souvent d'appuyer sur la touche **Q**.

Pour quitter certains visualiseurs de texte dont vim et vimdiff, il faut faire **:q** ou **:q!** si on veut forcer à quitter sans enregistrer les modifications (et valider en appuyant sur **Entrée**).

5.1.2 Raccourcis claviers

Ctrl + **A** pour aller au début de la ligne.

Ctrl + **E** pour aller à la fin de la ligne.

Ctrl + **U** pour effacer tout ce qu'il y a avant le curseur.

Ctrl + **K** pour effacer tout ce qu'il y a après le curseur.

Ctrl + **W** pour effacer un mot à gauche de la position du curseur (n'efface que la partie gauche du mot si le curseur est au milieu).

Ctrl + **→** et **Ctrl** + **←** permettent dans certains terminaux de se déplacer d'un mot à gauche ou à droite.

5.1.3 Rechercher une commande : **Ctrl** + **R**

Il est possible de rechercher dans l'historique avec le raccourci **Ctrl** + **R**. Cette recherche est incrémentale, cela signifie qu'au fur et à mesure de la frappe les résultats correspondant apparaissent, une fois la bonne commande trouvée, il suffit de taper sur **Entrée** pour l'exécuter ou sur la flèche droite pour copier cette commande sur la ligne courante si on veut y ajouter des paramètres.

Remarque : En appuyant de nouveau sur **Ctrl** + **R**, on se déplace dans la liste des différentes commandes qui correspondent à la recherche effectuée.

En moins pratique, il y a **!** qui permet de retrouver la dernière commande commençant par le texte tapé :

```
!sudo
```

va retrouver la dernière commande commençant par **sudo**

5.1.4 Stocker dans un fichier : **>**

commande > fichier.txt : permet de stocker le résultat de la commande placée avant ceci dans le fichier **fichier.txt**. Celui-ci sera créé s'il n'existe pas, ou remplacé s'il existe.

5.1.5 Stocker à la suite d'un fichier : »

`commande >> fichier.txt` : permet de stocker le résultat de la commande placée avant ceci à la fin du fichier **fichier.txt**. Le résultat sera placé à la suite du contenu déjà existant du fichier.

5.1.6 Exécuter en tâche de fond : &

`commande &` : permet de créer un processus dans lequel le programme va s'exécuter et de pouvoir dans le même temps continuer d'utiliser le terminal, ça lance donc en quelque sorte la commande en tâche de fond.

5.1.7 Passer les arguments entre commandes : |

`commande1|commande2` : permet d'exécuter plusieurs commandes à la suite. En fait, permet aussi de passer l'argument de sortie de la commande précédente dans la commande suivante, mais j'ai pas encore réellement compris le fonctionnement.

Par exemple

```
ls -l *.f90 | grep "10:10"
```

permet de lister les fichiers ayant l'extension **.f90** et dont la ligne contient l'expression **10:10**. Dans la pratique, ça affichera les fichiers dont la dernière modification s'est faite à 10h10.

5.1.8 &&

`commande1 && commande2` permet d'exécuter `commande1`, puis exécuter `commande2` si `commande1` s'est exécutée sans retourner d'erreur.

C'est très pratique quand on a une succession de commande à lancer.

Je lance par exemple

```
(cd .. && Makefile.py) && mercury-clean.sh && mercury && element
```

Je vais dans le répertoire parent tant que je reste dans la parenthèse; je compile. Si la compilation réussit, je lance le script de nettoyage des fichiers de la simulation, je lance la simulation, et je récupère les fichiers de sortie.

5.1.9 Argument précédent : !\$

Qui n'a jamais rêvé de reprendre l'argument précédemment fourni pour le réutiliser dans la commande actuelle? `!$` le permet.

```
ls Desktop
cd !$
```

et vous voici arrivé dans le répertoire **Desktop**.

Remarque : Notez que si vous cherchez la commande précédente, ce sera `cd Desktop` et non `cd !$`.

5.1.10 Le bash {} expansion

Petite fonction sympa de bash qui va nous éviter de réécrire certaines info. Par exemple, vous voulez faire un backup de votre `/etc/postgresql/8.3/main/postgresql.conf` mais vous êtes dans votre `/home`.

utilisez alors le `/chemin/du/fichier{,votreajout}`
donc en gros :

```
cp /etc/postgresql/8.3/main/postgresql.conf{,.bak}
```

va en fait faire un

```
cp /etc/postgresql/8.3/main/postgresql.conf /etc/postgresql/8.3/main/postgresql.conf.bak
```

Et ça marche aussi pour `mv` ou `diff` par exemple :

```
diff fichiermodifie{,.old}
```

5.1.11 Faire des maths : `$(())`

Besoin de faire des maths ? `\$(())` vous permettra de faire une petite opération vite fait. Par exemple

```
echo $(( 5 + 7 ))
```

5.1.12 Substitution de Commande

Vous pouvez utiliser le résultat d'une commande comme paramètre d'entrée pour une autre ; en d'autres termes pour une substitution de commande. Une substitution de commande a lieu lorsque vous encadrez une commande avec des apostrophes inversées. Par exemple :

```
cat `find . -name aaa.txt`
```

va afficher à l'écran le contenu de tout les fichiers nommés **aaa.txt** dans le dossier de travail courant ou ses sous-dossiers.

5.1.13 Mettre une commande en background a posteriori

Prenons un exemple. Vous lancez une commande dans le terminal sans la lancer dans un processus séparé avec l'esperluette (&). Pas besoin de tuer le processus pour reprendre la main. Il suffit de faire `Ctrl` + `Z` puis de taper dans la console `bg` + `Entrée` (background).

De la même manière, on peut mettre une commande en avant plan en taper `fg` (foreground).

5.2 Présentation des commandes usuelles

5.2.1 cd

`cd` : permet de changer le répertoire courant

`cd` tout court est équivalent à `cd ~` qui nous place dans le dossier de l'utilisateur local

`cd ..` répertoire parent

`cd -` répertoire précédent

5.2.2 chmod

Méthode alphabétique `chmod [1] [2] [3] fichier`

[1] Prend les valeurs suivantes :

- u user : définit les permissions pour le propriétaire du fichier
- g group : définit les permissions pour le groupe auquel appartient le fichier
- o other : définit les permissions pour tout les autres

[2] + signifie qu'on va ajouter les permissions définies après ce signe
 - signifie qu'on va enlever les permissions définies après ce signe
 = signifie qu'on définit exactement les permissions à celles citées après ce signe

[3] r read : permission de lecture du fichier ou dossier
 w write : permission d'écriture (modification/suppression/création) du fichier ou dossier
 x execute : permet d'exécuter le fichier

Exemples

```
chmod go-wx fichier
```

```
chmod u+x fichier
```

La première commande enlève les droit d'écriture et d'exécution sur **fichier** pour le groupe et les autres. La deuxième commande ajoute les droit d'exécution sur **fichier** au propriétaire de celui-ci (user)

Méthode numérique On définit une suite de 3 chiffres définissant les droits sur le fichier dans l'ordre **propriétaire / groupe / autres**.

Au chiffre 1 correspond le droit d'exécution

Au chiffre 2 correspond le droit d'écriture

Au chiffre 4 correspond le droit de lecture

Il suffit pour définir plusieurs droits d'additionner les chiffres. On peut expliquer ces chiffres d'une manière binaire qui explique relativement bien les choses.

Considérons les droits dans l'ordre suivant **lecture/écriture/exécution**. Si on donne à chaque droit une valeur binaire (1 pour oui, et 0 pour non), on retrouve la valeur décimale que l'on doit donner pour associer les droits. Exemple :

Je veux donner lecture, écriture et exécution au propriétaire, seulement lecture et écriture au groupe, et uniquement lecture aux autres.

Pour le propriétaire : **lecture/écriture/exécution** = 111_(binaire) \Rightarrow 7_(décimal)

Pour le groupe : **lecture/écriture/exécution** = 110_(binaire) \Rightarrow 6_(décimal)

Pour les autres : **lecture/écriture/exécution** = 100_(binaire) \Rightarrow 4_(décimal)

Ainsi, on entrera la commande suivante :

```
chmod 764 fichier
```

5.2.3 dir

dir liste les fichiers du répertoire courant.

5.2.4 du

Commande que j'ai retrouvée par hasard suite à un besoin de copier un gros répertoire, il vous affiche la taille mémoire utilisée du répertoire et des sous répertoires.

Utilisation :

```
du -h /repertoire
```

5.2.5 find

find -type d >Musiques.txt Permet de lister les répertoires et sous répertoires d'un dossier

find >Films.txt Permet de lister les fichiers, répertoires et sous répertoires (et fichiers inclus) d'un dossier

5.2.6 finger

Affiche les informations sur les utilisateurs

5.2.7 htop

htop est une version de « top » améliorée. Il y a de la couleur, on peut se déplacer sur les processus via les flèches du clavier, et effectuer des actions via les touches « F1-10 »

5.2.8 kill

kill -9 PID : Permet de tuer le processus avec le PID fourni

5.2.9 killall

killall irssi

Tue les processus à partir de leur nom.

5.2.10 locate

locate fichier.txt

Recherche toutes les copies de **fichier.txt** sur le disque dur. Pratique pour retrouver un fichier dont on connaît le nom mais pas l'emplacement.

5.2.11 ln

La commande ci-dessous permet de créer un lien symbolique dans le dossier dans lequel on est placé actuellement. Par exemple, en ce plaçant dans `/var/www` ça permet de faire un lien symbolique vers le dossier des fichiers d'un site web, et de pouvoir les modifier facilement. À noter que dans le cas du site web, il faut une permission 755 sur le dossier du site web. De plus, dans le localhost, il n'apparaîtra pas, il faudra taper manuellement le dossier pour se retrouver dans celui-ci.

```
sudo ln -s /home/votre_identifiant/répertoire_pour_votre_site mon_site
```

5.2.12 ls

`ls` : liste les fichiers et dossiers du répertoire courant

`-a` affiche aussi les fichiers cachés

`-l` liste en affichant en tableaux les permissions propriétaires et cie

`-F` affiche les fichiers uniquement

`-R` affiche récursivement le contenu des sous répertoires

5.2.13 man

`man` permet de donner une aide sur une commande.

Exemple : `man wine`

Pour quitter l'aide de la console et pouvoir à nouveau taper des commandes, il faut appuyer sur "**Q**"

5.2.14 mkdir

`mkdir` : crée un répertoire

`-p` permet de créer aussi les répertoires parents qui n'existent pas non plus et donc de créer plusieurs répertoires imbriquées en une seule fois à l'aide de `"/"`

5.2.15 pidof

Trouve l'ID d'un programme en cours d'exécution

```
pidof fetchmail
```

5.2.16 ps

`ps -A` : permet de lister les processus

5.2.17 pwd

`pwd` : affiche le chemin du répertoire courant

5.2.18 qui

Affiche le plan des salles (souligné = tuteur)

5.2.19 rm

`rm` : permet d'effacer des fichiers

`-r` permet de supprimer aussi des dossiers

5.2.20 rwho

Affiche personnes loguées sur une machine de la salle

5.2.21 rwrite

`rwrite personne@machine` envoie un message

5.2.22 shutdown

Permet d'éteindre l'ordinateur en ligne de commande.

`shutdown -h now` Éteint l'ordinateur immédiatement

`shutdown -r now` Redémarre l'ordinateur immédiatement

`shutdown -h 10 :00` programme l'extinction à 10h00 du matin (donné en format 24h)

`shutdown -h +10` Programme l'extinction de l'ordinateur dans 10 minutes

`shutdown -t 30` Programme l'extinction de l'ordinateur dans 30 secondes

5.2.23 sort

Permet de trier. Je ne sais pas dans quelle mesure il trie, mais je sais l'utiliser pour qu'il me trie dans l'ordre alphabétique les lignes d'un fichier, et ça, c'est super pratique :

```
sort -d ~/autiwa.cwl>autiwa2.cwl
```

5.2.24 top

la commande `top` permet de lister les processus les plus gourmands. Cette commande affiche un tableau qui est mis à jour régulièrement, la commande ne se termine donc pas toute seule comme `ps`. Voir aussi [§ 5.2.7 page 28]

5.2.25 uniq

Pour enlever les doublons de lignes, pratique quand vous faites une liste rapidement par copier coller ou par redirection de fichiers.

Utilisation :

```
uniq fichier_source > fichier_destination
```

5.2.26 vim

`vim` : permet d'éditer un fichier dans la console.

Vim possède plusieurs mode, au départ, on est dans le mode commande. Pour rentrer dans le mode édition, il faut taper sur `'i'`, et sur **Echap** pour en sortir. Dans le mode commande voici les raccourcis usuels :

`:w` pour enregistrer

`:q` pour quitter

`:q!` pour quitter sans enregistrer (si des modifications ont été faites, la commande `:q` ne marchera pas si vous n'avez pas enregistré au préalable.

`:x` pour sauvegarder et quitter.

Remarque : Pour ma part, je préfère nano que je trouve plus simple à utiliser quand on a besoin d'un éditeur en console.

5.2.27 wget

Permet de télécharger en ligne de commande.

```
wget http://lesite.com/lefichier.pdf
```

Pour reprendre un téléchargement interrompu :

```
wget -c http://lesite.com/lefichier.pdf
```

Pour spécifier le nom du fichier que l'on veut

```
wget -c http://lesite.com/lefichier.pdf -O "lenomquejeveux.pdf"
```

5.2.28 ytalk

ytalk personne@machine propose une discussion à la personne considérée

5.3 Présentations de programmes plus spécifiques

5.3.1 alltray

Programme qui permet de réduire dans le systray n'importe quelle application. Pour automatiser ce processus par exemple avec des icônes³, il suffit de rajouter le préfixe **alltray** avant le nom du programme.

Exemple : thunderbird devient alltray thunderbird

5.3.2 Mencoder

Découper une vidéo sans encoder

```
mencoder fichier_trop_long.wmv -oac copy -ovc copy -ss 00 :00 :00 -endpos 00 :xx :xx \
-o fichier_de_la_bonne_longueur.wmv
```

Joindre des vidéos sans ré-encoder Avant tout, il faut que les vidéos utilisent les mêmes codecs, aient la même taille, bref, semblables. Pour les joindres, et ce, (à priori) quel que soit les codecs :

```
mencoder fichier1.wmv fichier2.wmv fichier3.wmv -oac copy -ovc copy -o fichier_joint.wmv
```

Remarque : Sur le papier, c'est censé joindre parfaitement, mais pour une raison que j'ignore, le test que j'ai effectué donnait une qualité d'image moindre au fichier de sortie...

Convertir .rm en .avi

```
mencoder toto.rm -ovc lavc -lavcopts vcodec=mpeg4 :vbitrate=384
-vf scale=640 :480 -oac mp3lame -o toto.avi
```

Extraire les frames en tant qu'images

```
mplayer -vo gif89a -vf scale=640 :480 in.mpg
```

où **gif89a** peut être remplacé par **jpg** ou **png**. et où **in.mpg** désigne bien entendu la vidéo d'entrée. PS : à noter qu'on utilise *MPlayer* et non *Mencoder*.

Reconstruire l'index d'un .avi de façon permanente

```
mencoder -forceidx -oac copy -ovc copy -o out.avi in.avi
```

5.3.3 ffmpeg

Fait en gros la même chose que *Mencoder*

Extraire le son d'une vidéo .flv

```
ffmpeg -i flashvideo.flv -ar 44100 -ac 2 -ab 128k output.mp3
```

Extraire un morceau de la bande son d'une vidéo La commande ci-dessous permet d'extraire, à partir de 18 minutes et 33 secondes de film, la bande son des 15 prochaines secondes et d'enregistrer le tout dans un fichier mp3. Bien entendu, on peut spécifier le début en secondes, et inversement, la durée en heure :minutes :secondes.

```
ffmpeg -i "video.avi" -f mp3 -ss 00 :18 :33 -t 15 audio.mp3
```

3. ceci marche aussi pour le démarrage automatique en début de session

Convertir .rm en .avi

```
ffmpeg -i toto.rm -acodec aac -ab 128 -vcodec \
mpeg4 -b 384kb -mbd 2 -flags +4mv+trell -aic 2 -cmp 2 -subcmp 2 -s \
320x180 -title "Toto" toto.avi
```

5.3.4 pdftk : Manipuler les fichiers .pdf

Fusionner des pdf

```
pdftk fichier1.pdf fichier2.pdf fichier3.pdf cat output fichier.pdf
```

Fusionner des parties de pdf

```
pdftk A=fichier1.pdf B=fichier2.pdf C=fichier3.pdf cat A1-2 B5 C output fichier.pdf
```

Dans cet exemple, on prend les pages 1 et 2 du premier pdf, la page 5 du 2^e pdf, et tout le 3^e pdf.

Pivoter des pdf `pdftk A=entree.pdf cat A-W output sortie.pdf`

Ici, on applique une rotation à tout le pdf **A**. Pour désigner la rotation à effectuer on utilise la première lettre du point cardinal en anglais :

- North : 0°
- East : 90°
- South : 180°
- West : 270°

Résumé des 3 parties précédentes On peut combiner les commandes précédentes comme dans l'exemple qui suit :

```
pdftk A=entree.pdf cat A1-4 A5-S A6-47 output sortie.pdf
```

Dans cet exemple, on ne prend qu'un pdf, mais on peut bien entendu en prendre plusieurs. Cet exemple recopie les 4 premières pages de l'entrée, puis effectue une rotation de 180° à la 5^e page (en gros, elle la renverse), et enfin recopie les autres pages (mon exemple possédait 47 pages). Ceci m'a permis de remettre à l'endroit une page qui avait son texte dans le mauvais sens.

Protéger un .pdf par mot de passe

```
pdftk mondocument.pdf output mondocument.128.pdf owner_pw foo user_pw monmotdepasse
```

Il suffit de remplacer **monmotdepasse** parce le mot de passe souhaité, **mondocument.pdf** par le nom du pdf en entrée et de remplacer **mondocument.128.pdf** par le nom du nouveau **.pdf** avec mot de passe qui sera créé. Il faut cependant comme d'habitude se placer dans le répertoire où se trouve le **.pdf**.

5.3.5 record : enregistrer des sons

J'ai voulu récemment enregistrer des sons via l'entrée micro de mon PC. j'ai essayé audacity, celui-ci me propose enregistrement, enregistrement1 et digital comme entrée audio, mais ne me propose pas mic (microphone) qui se trouve être la bonne entrée, c'est assez facheux. Même soucis du côté de l'enregistreur de son par défaut.

Et puis il y a *record*. Alors bien entendu, c'est un programme en ligne de commande, on a pas trop le choix pour le nom du fichier voire l'encodage du son (**record.wav** et **wave** respectivement). Et encore que, je dis qu'on ne peut pas parce que je ne sais pas le faire, si ça se trouve ces configurations sont possibles. Quoi qu'il en soit, je lance le programme dans la console via :

```
record -i mic
```

C'est à dire que je peux spécifier le périphérique d'entrée, très pratique. L'interface est dans la console et toutes les commandes sont expliquées, j'active le réglage automatique du volume (**A**), et hop, un coup d'**Espace** et ça enregistre. Pour information, le fichier s'enregistre par défaut dans le répertoire utilisateur, c'est à dire `/home/login/` mais si ça se trouve, c'est simplement dans le répertoire courant, mais j'ai pas testé.

5.3.6 sed : manipuler les chaînes de caractères

La syntaxe de la commande substitution est la suivante :

```
sed -e 's<séparateur><chaîne à remplacer><séparateur><chaîne remplaçante><séparateur><options>'
```

<séparateur> : ici j'ai utilisé le caractère / par habitude, ça fonctionnerait tout aussi bien avec , ou ; etc...

<chaîne à remplacer> : ici j'ai utilisé \.mp3\$

.mp3 c'est l'extension que l'on veut enlever.

\$ permet de dire que je ne veux enlever que le .mp3 final ce qui évite de supprimer autre chose que l'extension. Imaginons un fichier nommé coucou.mp3 par titi.mp3 si tu ne mets pas le \$ tu vas remplacer par coucou par titi.mp3 au lieu de coucou.mp3 par titi

dans sed, le caractère . étant un caractère joker, il faut annuler son interprétation avec un \

<chaîne remplaçante> : ici on veut effacer donc je n'ai rien mis

<options> : ici on n'en a pas besoin donc je n'en ai pas mis.

Pour d'autres utilisation à venir, les options que j'utilise le plus souvent sont :

un chiffre : pour dire je veux remplacer l'occurrence x

g : pour dire je veux remplacer toutes les occurrences

I : permet de rendre le remplacement insensible à la casse.

Exemples `sed -r 's/\<./\U&/g'` : Permet de mettre en majuscule la première lettre de chaque mot de la chaîne de caractère qu'on lui passe en paramètre. Si on spécifie le nom d'un fichier après cette commande, alors il remplacera dans tout le fichier. En rajoutant `>nom_dun_fichier.txt`, il stockera le résultat dans ce fichier.

5.3.7 convert : Convertir un .pdf en .png

En ligne de commande Convertir un fichier PDF en image avec *convert* fourni avec *ImageMagick*. La commande suivante permet de convertir un fichier pdf au format png.

```
convert MonFichier.pdf MonFichier.png
```

À l'aide de Gimp Je rajoute cette section parce que je viens de me rendre compte que convert ne faisait pas un .png très propre. J'ai voulu avoir une image .png d'un tableau excel (open office, mais .xls) et la conversion .pdf en .png ne me permet pas de lire les caractères proprement quand je zoome sur l'image alors que dans le .pdf original (choisi en 1200 dpi dans open office) je pouvais zoomer sans que les caractères ne pixellisent.

J'ai donc voulu chercher une autre méthode. J'en ai trouvé une avec Gimp. Dans un premier temps, il faut ouvrir le .pdf avec *The Gimp*, dans la fenêtre qui s'affiche, vous pouvez choisir les pages qui vous intéressent. Au niveau des réglages, j'ai pris **900 dpi** qui me permet d'avoir les traits bien propres.

Ensuite, je converti mon image en couleurs indexées vu que des fichiers textes ont peu de couleurs, on gagne de la place. réduire la taille de l'image est aussi une option⁴ vu qu'en mettant 900 dpi, je me suis retrouvé, dans mon exemple, avec une image de 8000px par 5000px

5.3.8 inkscape en ligne de commande

Nettoyer les définitions Ça permet de nettoyer les définitions inutilisées d'un fichier .svg. Je le fais généralement en lançant l'interface graphique, mais cette ligne va me permettre de faire un script pour automatiser tout ça dans un dossier, bref, c'est super pratique :

```
inkscape fichier.svg --vacuum-defs
```



Je ne sais pas si ça a été résolu, mais cette fonction, au moment où je l'ai utilisée, avait le facheux défaut de supprimer tout les marqueurs des chemins. En gros, mes flèches devenaient des simples traits.

4. Attention cependant, si vous réduisez trop l'image, pour que celle-ci reste propre, il peut être nécessaire de repasser en RVB, de réduire la taille aux dimensions voulues et ensuite de repasser aux couleurs indexées, sinon les écritures pourraient être moches...

Exporter en .png

```
inkscape fichier.svg -e fichier.png -w 300 -h 500
```

-e indique qu'il s'agit d'un export png -w largeur du bitmap à exporter -h hauteur du bitmap à exporter

Remarque : Je ne sais pas encore comment on fait pour spécifier de garder les proportions, peut-être en ne spécifiant qu'une seule dimension.

5.3.9 zenity

zenity permet d'avoir facilement une interface graphique, et donc d'afficher des informations à l'écran pour des scripts.

Pour afficher du texte à l'écran, on utilise la syntaxe suivante :

```
zenity --info --text="Le Backup a été effectué avec succès." --title="Informations"
```

On peut mettre en forme ce texte pour mettre de la couleur ou des mises en formes basiques :

```
zenity --info --text "
<span color=\"red\">red</span>
<span color=\"green\">green</span>
<span color=\"blue\">blue</span>
<span color=\"yellow\">yellow</span>
<span color=\"magenta\">magenta</span>
<span color=\"white\">white</span>
<span color=\"black\">black</span>
<span color=\"gray\">gray</span>
<span color=\"lightblue\">lightblue</span>
<span color=\"lightgray\">lightgray</span>
<b><big>ETC...</big></b>
<b>style</b>
<big>style</big>
<i>style</i>
<s>style</s>
<sub>style</sub>
<sup>style</sup>
<small>style</small>
<tt>style</tt>
<u>style</u>
"
```

D'autres exemples d'utilisation ci dessous :

- En rajoutant | **zenity progress pulsate** à droite d'une commande, on peut avoir une barre de « progression » tant que la commande s'exécute.
- **zenity question title "Alert" text "Êtes-vous sur?"** affiche une alerte. Si on valide, alors ça retourne 0, si on annule, ça retourne 1. (en gros, 0 pour oui, et 1 pour non)
- **zenity title="Select a file to remove" fileselection** permet de sélectionner un fichier, et son chemin complet sera retourné.

Remarque : Pour stocker le résultat dans une variable, on procède comme suit : **variable='zenity ***'**, ça permet d'exécuter la commande et stocker son argument.

5.4 Actions faciles à faire en ligne de commande

5.4.1 Effacer des fichiers en masse

À l'aide de la commande **find** on peut créer en ligne de commande la recherche de fichiers que l'on efface par la suite :

```
find . -name "*.db" -exec rm -f {} \;
```

Dans cet exemple, on recherche les fichiers de type **.db** (**-name "*.db"**, ce qui permet, en pratique, de supprimer les fichiers **Thumbs.db**, en supprimant tout les fichiers avec cette extension — pour palier les éventuels renommage —) du répertoire courant (**.**) que l'on va effacer. le **-f** permet d'enlever la confirmation de suppression.

5.4.2 Changer les droits des dossiers de manière récursive

La commande

```
find -type d -exec chmod 755 {} \;
```

permet de modifier les droits des dossiers et sous-dossiers du répertoire courant. (le propriétaire a les droits d'écriture, lecture et exécution tandis que les autres ont les droits de lecture et exécution)



Il est nécessaire d'avoir les droits d'exécution sur un dossier pour pouvoir voir le contenu de ce dernier

Remarque : La même chose est possible pour les fichiers uniquement en faisant :

```
find -type f -exec chmod 644 {} \;
```

5.4.3 Convertir un .ps en .pdf

La commande est simple, mais l'option est importante. Par défaut, la rotation à effectuer sur une page est déterminée par l'orientation du texte le plus nombreux. Ceci peut parfois poser des problèmes, et c'était mon cas. Afin de désactiver la rotation il faut faire

```
ps2pdf -dAutoRotatePages=/None fichier.ps
```

Remarque : Si non spécifié, le nom du fichier **.pdf** de sortie sera le même que le nom du fichier **.ps**.

5.4.4 Déplacer des fichiers dans le répertoire parent

On se place dans le répertoire contenant les différents dossiers qui contiennent les images. Celles-ci seront déplacées et mises dans le répertoire courant. Il est aussi possible de spécifier un autre répertoire en donnant son emplacement, relatif ou absolu. À noter qu'on peut aussi garder cette ligne de commande et se placer dans le bon répertoire via le **cd** approprié

```
find . -name "* - Page*" -exec mv {} "'pwd'" \;
```

5.4.5 Changer l'encodage d'un texte

```
iconv -f iso-8859-1 -t utf-8 entree.txt>sortie.txt    # vers UTF-8
iconv -f utf-8 -t iso-8859-1 entree.txt>sortie.txt    # vers latin-1
```

5.4.6 Inverser l'ordre des lignes

Si on a un fichier texte dont on souhaite inverser l'ordre des lignes, il suffit de faire en ligne de commande :

```
tac texte.txt>texte2.txt
```

ainsi, un texte comme le suivant :

```
ligne 01
ligne 02
ligne 03
```

sera stocké dans le fichier `texte2.txt` sous la forme suivante :

```
ligne 03
ligne 02
ligne 01
```

5.4.7 Lire le début ou la fin d'un fichier

Dans le cas de fichiers extrêmement gros, typiquement plusieurs Gio, il n'est pas possible de les ouvrir entièrement. Ça fait planter tous les éditeurs de textes (en tout cas la plupart), et même **vim** a du mal.

La solution est d'afficher dans le terminal le début ou la fin du fichier, sans avoir à charger le fichier entier.

Pour voir le début du fichier

```
head fichier.txt
```

et pour voir la fin du fichier

```
tail fichier.txt
```

5.4.8 Extraire un DVD avec mplayer

Tout d'abord, il faut trouver le bon titre. En effet, sur un DVD il y a parfois, voire même souvent, plusieurs titres, ne serait-ce que pour une introduction ou autre. On a donc, en plus du découpage en chapitres, une divisions en titre.

Pour cela, rien de plus simple, il suffit de tester les titres pour trouver celui qui nous intéresse via

```
mplayer dvd://1
```

en faisant varier le chiffre (dans mon cas, c'était `mplayer dvd://4` par exemple).

Si on souhaite extraire chaque chapitre séparément, on a besoin d'une étape intermédiaire qui est

```
dvdxchap -t 4 /dev/dvd > chapters.txt
```

où 4 est le numéro du titre qui nous intéresse. Ceci permet de connaître le nombre de chapitres.

Ensuite, pour extraire le DVD en `.vob`

```
mplayer -dumpstream dvd://4 -dumpfile video.vob
```

Pour extraire chapitre par chapitre, voir le script **dvdchapter2vob.sh** (section [§ 6.10 page 42])

Pour encoder, voir l'astuce [§ 3.8.2 page 14].

5.4.9 Extraire les sous-titres d'un fichier .mkv

Il faut d'abord connaître l'ID du fichier de sous-titre à l'intérieur du fichier `.mkv` :

```
$ mkvmerge -I movie.mkv
File 'movie.mkv': container: Matroska [duration:7753608000000]
Track ID 1: subtitles (S_TEXT/UTF8)
[language:und default_track:1 forced_track:0]
Track ID 2: video (V_MPEG4/ISO/AVC)
[language:und display_dimensions:1280x544 default_track:1
forced_track:0 packetizer:mpeg4_p10_video]
Track ID 3: audio (A_AAC) [language:und default_track:1 forced_track:0]
Chapters: 1 entries
```

Ici, c'est le premier ID. On adapte en conséquence et on extrait l'ID numéro 1 pour le stocker dans le fichier de notre choix, que j'appelle **subtitle.srt** dans mon cas :

```
mkvextract tracks "movie.mkv" 1:subtitle.srt
```

5.4.10 Rechercher/Remplacer de manière récursive dans plusieurs fichiers

```
find . -name "*.php" -exec sed -i 's/mot1/mot2/g' {} \;
```

La ligne de commande ci-dessus permet de remplacer toutes les occurrences de **mot1** par **mot2** dans tous les fichiers **php** du dossier courant ou de ses sous-dossiers.

5.5 Mettre de la couleur dans la console

Pour ajouter de la couleur à votre terminal, éditez votre fichier `~/.bashrc` (`gedit ~/.bashrc`) :

- ajoutez un `#` au début des lignes 26 à 33 (les 8 lignes sous `# set a fancy prompt (non-color, unless we know we "want" color)`)
- en dessous de la ligne 36 :

```
#PS1='${debian_chroot :+(debian_chroot)}\[\033[01;32m\]  
\u@\h\[\033[00m\] : \[\033[01;34m\]\w\[\033[00m\]\$ '
```

ajoutez cette ligne :

```
PS1='${debian_chroot :+($debian_chroot)}\[\033[0;31m\]  
\u@\h\[\033[0;0m\] : \[\033[0;33m\]\w\[\033[0;0m\]\$ '
```

5.5.1 Trier les lignes d'un fichier par ordre alphabétique

```
sort -d ~/autiwa.cwl>autiwa2.cwl
```

5.6 Manipuler un fichier texte

5.6.1 Expressions régulières regexp

Grâce à la commande **Rechercher/Remplacer** on peut manipuler très précisément les textes, pour peu qu'on pense à rechercher dans tout le texte, manipuler toutes les occurrences et activer les expressions régulières. Ces commandes marchent avec Geany et Kate, j'ai supposé que ça marchait pour les autres éditeurs de textes.

.	correspond à n'importe quel caractère
(La parenthèse ouvrante marque le début d'une zone de « tag ». C'est à dire que le contenu de cette région sera mémorisé, et pourra être réutilisé dans la section remplacement (cf la partie sur \n pour plus de détails au sujet du remplacement par des zones taggées)
)	Marque la fin d'une zone taggée
\n	Ici, n représente un nombre entier allant de 1 à 9. On utilise cette commande dans la section remplacement. donc \1 fera référence à la première zone taggée. Prenons un exemple. Voici un exemple d'expression régulière : Fred([1-9])XXX . Ainsi, la zone entre parenthèse est la zone taggée numéro 1 (ici il n'y en a qu'une, mais s'il y en avait d'autres, elles auraient un numéro affecté en fonction de leur ordre d'apparition dans l'expression régulière. Supposons qu'on applique cette expression sur la phrase Fred2XXX . Le « 2 » correspond à la zone taggée numéro 1. Ainsi, si dans la section de remplacement, on écrit Sam\1YYY ; la phrase précédente sera remplacée par : Sam2YYY
\<	correspond au début d'un mot
\>	correspond à la fin d'un mot
\x	permet d'utiliser le caractère x (où x est un caractère non défini ici, mais qui correspond à un caractère « protégé », c'est à dire qui est utilisé pour des commandes) comme un caractère normal alors que celui-ci aurait été considéré comme une commande pour une expression régulière. Par exemple pour rechercher un crochet, il faut utiliser \[
[...]	permet de rechercher un caractère parmi une liste. Par exemple [abc] recherchera l'un des trois caractères. [a-zA-Z] recherchera une lettre majuscule ou minuscule.
[^...]	recherche un caractère parmi le complètement aux caractères dans les crochets. [^a-z] recherche tout caractère qui n'est pas une lettre minuscule.
^	recherche le début d'une ligne
\$	recherche la fin d'une ligne
*	Permet de rechercher un nombre compris entre 0 et l'infini du caractère précédent immédiatement *. Par exemple Sa*m permet de trouver Sm, Sam, Saam, Saaam et ainsi de suite. Si juste avant * on a une liste de caractère (via les crochets), alors ça s'appliquera à la liste complète des caractères. ([ab]* recherchera une séquence de a et b dont le nombre de caractère sera compris entre 0 et l'infini.
+	Ce caractère fonctionne comme *, à la différence près que le nombre d'occurrence doit être compris entre 1 et l'infini. Par exemple Sa+m permet de trouver Sam, Saam, Saaam et ainsi de suite

!.*\$

La commande ci dessus permet de sélectionner toute suite de caractère qui se trouve entre un point d'exclamation et la fin d'une ligne.

\([a-zA-Z?!]*\)

La commande ci-dessus permet de rechercher toute suite de caractère minuscule ou majuscule incluant point d'exclamation, point d'interrogation ainsi que l'espace qui se trouve dans deux parenthèses

5.6.2 Rechercher-remplacer regexp

Ci-dessous un exemple d'expressions à rentrer et de ce qu'il fait. l'exemple est assez complet et permet de voir pas mal de commande. Il est à noter qu'en pratique, j'ai eu un problème avec cette expression et pour les doubles pages, il ne me mettait que le numéro de la dernière page.

expression à rechercher	.*v([0-9]+) p.*([0-9-]+)\.([a-z]+)
expression à remplacer	T\1 - Page \2.\3

Exemple de nom de fichier initial et final :

SA0 v02c03 p008.png

T02 - Page 008.png

Pour les explications sur les commandes, la plupart sont expliquées dans la paragraphe précédent sur les expressions régulières. Le point principal que je voulais expliquer ici, c'est le principe des régions "taggées" c'est à dire que les parties entre parenthèses sont sélectionnées et qu'on les appelle dans la section de remplacement via \1 jusqu'à \9 dans l'ordre d'apparition.

5.6.3 Supprimer des lignes vides

sed permet de le faire via la commande suivante (qui utilise les expressions régulières en fait)

```
sed '/^$/d' mon_fichier.txt
```

5.6.4 Extraire certaines colonnes d'un fichier

cut <colonnes> <fich> : extrait certaines parties dans chaque ligne. Les parties à extraire sont spécifiées :

- soit en indiquant leur position avec l'option **-c** : **cut -c5-15,33,37-** :extraire dans chaque ligne les caractères 5 à 15 et 33 et de 37 jusqu'à la fin de la ligne ;
- soit en indiquant un numéro de champ (avec **-f**) et un caractère délimiteur de champ (avec **-d**) : **cut -d"," -f3-5** :extraire les champs 3 à 5 de chaque ligne en utilisant le caractère « , » comme délimiteur de champ.

6 Faire des scripts Bash

Un script Bash doit commencer par cette ligne

```
#!/bin/bash
```

et en dessous, on met les commandes. ensuite, il faut penser à donner les droits d'exécutions à ce fichier.

Voir la section 5.3.9 pour des détails sur **zenity** qui permet de rajouter des interfaces graphiques aux scripts.

Les variables sont précédées du signe \$. Pour « protéger » une variable, il faut écrire `${nom_de_la_variable}`. En effet, si on utilise la variable dans des chaînes de caractères, il arrive bien souvent que bash ne sache pas où s'arrête le nom de la variable et où commence la suite de la chaîne de caractère.

6.1 Boucle for

```
for i in `seq 2 7`; do convert "TP0".$i.".png" "TP0".$i.".pdf" ; done
```

La commande ci-dessus permet de convertir les fichiers TP01.png à TP07.png en .pdf en une seule commande. Les points virgules permettent de simuler une mise à la ligne pour ne pas avoir à revenir à la ligne réellement.

La commande **seq 2 7** renvoie la liste des entiers compris entre 2 et 7, ce qui permet de faire l'itération.

Ou encore :

Exemple :

```
for ((i=1;i<5;i++)) do {
echo $i;
}
done
```

6.1.1 Boucle sur des fichiers

```
for file in *.f
do
    mv $file `basename $file .f`.f90
done
```

Cette boucle for va chercher tous les fichiers ayant l'extension **.f** dans le dossier courant et les renommer pour leur donner l'extension **.f90**.

Remarque : La partie **basename \$file .f** permet d'enlever l'extension **.f** du nom de fichier contenu dans la variable **\$file**

6.2 Manipuler les chaînes de caractères

6.2.1 \${chaine#souschaine}

Supprime la correspondance la plus petite de \$souschaine à partir du début de \$chaine.

Exemple :

```
chaineZ=abcABC123ABCabc
echo ${chaineZ#a*C}      # 123ABCabc
# Supprime la plus petite correspondance entre 'a' et 'C'
# à partir du début de $chaineZ.
```

6.2.2 \${chaine##souschaine}

Supprime la correspondance la plus grande de \$souschaine à partir du début de \$chaine.

Exemple :

```
chaineZ=abcABC123ABCabc
echo ${chaineZ##a*C}     # abc
# Supprime la plus grande correspondance entre 'a' et 'C'
# à partir du début de $chaineZ.
```

6.2.3 \${chaine%souschaine}

Supprime la plus petite correspondance de \$souschaine à partir de la fin de \$chaine.

Exemple :

```
chaineZ=abcABC123ABCabc
echo ${chaineZ%b*c}      # abcABC123ABCa
# Supprime la plus petite correspondance entre 'b' et 'c'
# à partir de la fin de $chaineZ.
```

Remarque : Pour supprimer l'extension (pour un script par exemple). Soit la variable \$file. Voici la commande à exécuter :

```
name=${file%.*}
```

6.2.4 \${chaine%%souschaine}

Supprime la plus grande correspondance de \$souschaine à partir de la fin de \$chaine.

Exemple :

```
chaineZ=abcABC123ABCabc
echo ${chaineZ%%b*c}     # a
# Supprime la plus grande correspondance entre 'b' et 'c'
# à partir de la fin de $chaineZ.
```

6.3 Faire une pause dans un script bash

```
#!/bin/bash
# pause2.sh : appuyer sur une touche pour continuer un script

echo "Appuyer la touche <Entrée> pour continuer..."
read touche
case $touche in
*) echo "Reprise du script..."
```



```
;;
esac

#!/bin/bash
# pause.sh : continuer oui ou non un script

echo "Voulez-vous continuer ? [O/N]"
read mot
while
[ "$mot" = "O" ]
do
echo "Voulez-vous continuer ? [O/N]"
read mot
done
echo "Programme achevé..."
```

6.4 Découpage des chemins

Les scripts shell manipulent souvent chemins (pathnames) et noms de fichiers. Les commandes `basename` et `dirname` sont très commodes pour découper un chemin en deux parties (répertoires, nom de fichier) :

```
$ dirname /un/long/chemin/vers/toto.txt
/un/long/chemin/vers
$ basename /un/long/chemin/vers/toto.txt
toto.txt
```

source : <http://www-gtr.iutv.univ-paris13.fr/Cours/Mat/Systeme/TDTP2003/tp03.html>

6.5 Tester si un fichier existe

```
if [ -e /media/stockage ]; then
    echo "On le tient"
else
    echo "Il est où ?"
fi
```

6.6 Tester si un binaire existe

```
which xcf2png 2>/dev/null
if [ $? != 0 ]
then
zenity --error --title="Attention" --text="Le programme <b>xcf2png</b> \
n'existe pas. Installez le paquet <b>xcftools</b>"
exit 0
fi
```

6.7 jpg en png (script compact)

```
#!/bin/bash

for file in *.jpg; do convert "${file}" "${file%.jpg}.png"; done
```

6.8 jpg2pdf.sh

```
#!/bin/bash

prefix=$1

for i in $prefix*.jpg
```

```

do
  i=${i%.jpg}      #on vire l'extension
  list="$list $i"
  z="mogrify -page A4+0+0 -format pdf $i.jpg"
  echo $z
  $z
done
z="pdftk $(for i in $list ; do echo $i.pdf ; done) cat output $prefix.pdf"
echo $z
$z

```

Il suffit de recopier ce code dans un fichier que l'on appelle **jpgs2pdf.sh**, sans oublier de lui donner les droits d'exécution. Ensuite pour l'utiliser, on procède comme suit :

Si on a deux pages fichier1.jpg et fichier2.jpg, on crée fichier.pdf avec `./jpgs2pdf.sh fichier` dans le bon répertoire. Il faut avoir le script dans le même dossier que les images, et il faut que les images aient le même nom avec le numéro accolé, commençant à 1 (et non à 01).

6.9 flv2mp3

```

#!/bin/bash
#pour convertir un fichier .flv en .mp3
#version 0.45
#Script à utiliser via nautilus, en
#faisant clic droit sur une sélection de fichiers .flv

while [ $# -gt 0 ]; do
  video=$1
  mp3_file='echo "$video" | sed 's/\.\w*$/ .mp3/'
  ffmpeg -i "$video" -ar 44100 -ac 2 -ab 128k "$mp3_file"
  shift
done

```

Il faut avoir installé nautilus-scripts pour pouvoir lancer le script après avoir sélectionné les fichiers .flv que l'on veut convertir.

6.10 dvdchapter2vob

Voici un script pour extraire les chapitres d'un DVD dans des fichiers séparés.

Il faut tout d'abord trouver la piste à extraire. On procède par tâtonnement en lançant

```

mplayer dvd://i

```

où *i* est un chiffre, généralement entre 1 et 4.

Ensuite, pour vérifier si le nombre de chapitres de la piste *i* correspond, on tape

```

dvdchapter -t i /dev/dvd

```

En clair, s'il n'y a qu'un ou deux chapitres, c'est sûrement un menu ou un truc du style, donc c'est pas bon.

```

#!/bin/bash
#pour extraire les épisodes d'un dvd en .vob
#version 0.1alpha

dossier="/media/video/temp"
for ((i=1 ; 11 - $i ; i++)) do {
  echo $i
  mplayer dvd://4 -chapter $i-$i -dumpstream -dumpfile "$dossier/episode_$i.vob"

}
done

```

Dans cet exemple, il y a 10 chapitres à extraire (de 1 à 10) et on a préalablement trouvé dans quel titre du DVD il se trouvaient, ici le numéro 4).

Pour encoder, voir l'astuce [§ 3.8.2 page 14].

Glossaire

Antivirus

Un antivirus – anti-virus – est un logiciel censé protéger un micro-ordinateur contre les programmes néfastes appelés virus, vers, troyens, macrovirus, etc.

Avahi

Avahi est une implémentation de ZéroConf permettant la découverte de services offerts par le réseau comme les imprimantes disponibles, partager de la musique et l'écouter à distance, parler par messagerie Internet aux autres ordinateurs hôtes de votre réseau, et tout cela sans configuration préalable.

BIOS

Le Basic Input Output System – système élémentaire d'entrée/sortie – est un ensemble de fonctions, contenues dans la mémoire morte – ROM – de la carte mère servant à faire des opérations basiques. Celui-ci émet notamment les premières commandes au système durant la phase de démarrage, pour indiquer par exemple sur quel disque et à quel endroit se trouve le MBR.

Boot loader

Terme anglais de « Chargeur d'amorçage ». Se référer à la définition de ce dernier.

Chargeur d'amorçage

Également appelé « Boot loader », ce programme permet de choisir quel système d'exploitation doit être lancé. Dans le cas de Windows, il s'agit du programme NTLDR, dans le cas d'un système en multiboot – possibilité de démarrer plusieurs systèmes d'exploitation sur un même ordinateur – Lilo dans les cas simples – Windows et Linux – ou GRUB dans les cas plus sophistiqués — tous systèmes supportés.

Codec

Le mot-valise « Codec » est construit d'après les mots « codeur » et « décodeur ». Il s'agit d'un procédé permettant de compresser et/ou de décompresser un signal audio ou vidéo, le plus souvent en temps réel. Les codecs peuvent être partagés entre plusieurs logiciels de lecture audio ou vidéo.

Console

Le mot console est un synonyme de « terminal ». Se référer à la définition de ce dernier.

Défragmentation

La défragmentation réorganise physiquement le contenu du disque pour remettre ensemble et dans l'ordre les fragments ou morceaux éparpillés d'un fichier. Ce processus d'élimination essaye également de créer une grande région d'espace libre pour retarder à nouveau la fragmentation du système de fichiers.

Démon

Le terme démon – ou daemon en anglais – désigne un type de programme. Le terme a été créé par les inventeurs d'Unix pour se référer à un processus qui s'exécute en arrière-plan plutôt que sous le contrôle direct d'un utilisateur. Les démons sont souvent démarrés lors du chargement du système d'exploitation, et servent en général à répondre à des requêtes du réseau, à l'activité du matériel ou à d'autres programmes en exécutant certaines tâches.

DHCP

Le Dynamic Host Configuration Protocol est un terme anglais désignant un protocole réseau dont le rôle est d'assurer la configuration automatique des paramètres TCP/IP d'une station, notamment

en lui assignant automatiquement une adresse IP, une passerelle et un masque de sous-réseau.

DNS

Le Domain Name System – ou système de noms de domaine – est un système permettant d'établir une correspondance entre une adresse IP – exemple : 80.82.17.133 – et un nom de domaine – exemple : <http://www.framabook.org> – et, plus généralement, de trouver une information à partir d'un nom de domaine.

Drag'n'Drop

Terme anglais de glisser-déposer, voir la définition de ce dernier.

Driver

Terme anglais de « pilote ». Se référer à sa définition.

DRM

Les Digital Rights Management, terme anglais de Gestion des Droits Numériques – GDN – a pour objectif de contrôler par des mesures techniques de protection, très restrictives et contraignantes, l'utilisation qui est faite des œuvres numériques.

Ethernet (câble)

Le câble ethernet le plus connu est certainement le câble RJ11, équipant la plupart des téléphones. Cette interface physique est souvent utilisée pour terminer les câbles de type paire torsadée, et sa version RJ45 est celle choisie pour équiper la plupart des cartes réseaux actuelles d'ordinateur.

Firewall

Terme anglais de « pare-feu ». Se référer à la définition de ce dernier.

Fork

Mot anglais signifiant « fourche » ou « fourchette », un fork correspond à un nouveau projet créé à partir d'un autre à l'identique, sans détruire ce dernier. Cela implique que les droits accordés par les auteurs le permettent : ils doivent autoriser la modification, l'utilisation et la redistribution du code source. C'est pour cette raison que les embranchements se produisent facilement dans le domaine des logiciels libres.

GDN

Terme français de DRM, largement plus utilisé. Voir la définition plus haut.

Geek

Terme anglais se prononçant [gi :k], un geek est une personne passionnée, voire obsédée, par un domaine précis (la plupart du temps rapporté à l'informatique et aux nouvelles technologies). À l'origine, en anglais le terme signifiait « fada », soit une variation argotique de « fou ».

Glisser-déposer

Pour réaliser un glisser-déposer, il faut d'abord sélectionner un ou plusieurs éléments, puis maintenir appuyé, lors du déplacement, le bouton gauche de la souris et enfin le relâcher sur le point d'arrivée.

Icône

Petit pictogramme représentant une action, un objet, un logiciel, un type de fichier, etc. Les icônes ont dans un premier temps servi d'outils pour rendre les environnements graphiques informatiques plus simple d'utilisation aux novices.

Identifiant

Appelé également « nom d'utilisateur » et « login » en anglais.

L'Internet — et non internet

Un internet est un réseau, l'Internet — ne pas oublier le « l' » et la majuscule — est « le réseau des réseaux ». L'adjectif, lui, est cependant bien « Internet », comme dans site Internet. Ce réseau informatique à l'échelle du monde, reposant sur le protocole de communication IP, rend accessible au public des services comme le courrier électronique et le web. Vous pourrez reprendre un grand nombre de présentateurs télévisés — qui a dit que ce n'est pas la première fois? :-)

IP (adresse)

Une adresse IP — IP pour Internet Protocol — est le numéro qui identifie chaque ordinateur sur l'Internet, et plus généralement, l'interface avec le réseau de tout matériel informatique — routeur, imprimante — connecté à un réseau informatique utilisant le protocole Internet.

Kernel

Terme anglais de « noyau ». Voir plus bas la définition de ce dernier.

Logiciels libres

Un logiciel libre se dit d'un logiciel qui donne le droit à toute personne possédant une copie, de l'utiliser, de l'étudier, de le modifier et de le redistribuer. Ce droit est souvent donné par une Licence Libre. Tout ceci se réfère directement au mouvement du Libre.

Logiciels propriétaires

Ce sont des logiciels dont l'utilisation est limitée — notion opposée à celle des « logiciels libres » — d'une manière très précise par un contrat de licence.

Masque de sous-réseau

Un masque de sous-réseau permet d'identifier un sous-réseau. En l'appliquant sur l'adresse IP de la machine, il permet de déterminer si certaines machines appartiennent, ou non, au même réseau alors qu'elles sont connectées physiquement.

MBR

Le Master Boot Record, également appelé « Zone d'amorçage », est le nom donné par le BIOS au premier secteur adressable d'un disque dur. Ce dernier peut contenir le « Chargeur d'amorçage » ou encore « Boot loader » ou une adresse pour l'atteindre.

Mémoire vive

La mémoire vive, dite mémoire système ou encore mémoire volatile est la mémoire dans laquelle un ordinateur place les données lors de leur traitement. C'est donc cette mémoire d'un accès très rapide qui est utilisée lorsque l'ordinateur est allumé. Celle-ci est complètement vidée à l'extinction de l'ordinateur. Cette notion est à opposer à la mémoire morte.

Mémoire morte

Une mémoire morte est une mémoire non volatile, c'est-à-dire une mémoire qui ne s'efface pas lorsque l'appareil qui la contient n'est plus alimenté en électricité. Le type le plus connu du grand public est, notamment, le disque dur.

Menu contextuel

Ce menu s'obtient en cliquant avec le bouton droit de la souris sur un objet. Il contient un choix de fonctions variant selon l'objet et son contexte.

Noyau

En informatique, un noyau de système d'exploitation abrégé en noyau – kernel en anglais – est la partie fondamentale de tous les systèmes d'exploitation. Elle gère les ressources de l'ordinateur et permet aux différents composants - matériels et logiciels - de communiquer entre eux.

Operating System

Souvent appelé OS. Terme anglais de Système d'exploitation. Voir cette définition.

Pare-feu

Un pare-feu est un élément du réseau informatique, logiciel et/ou matériel, qui a pour fonction de faire respecter la politique de sécurité du réseau, celle-ci définissant les types de communication autorisés ou interdits. Il a pour principale tâche de contrôler le trafic entre différentes zones de confiance, en filtrant les flux de données qui y transitent. Généralement, les zones de confiance incluent l'Internet – une zone dont la confiance est nulle – et au moins un réseau interne — une zone dont la confiance est plus importante.

Passerelle

En informatique, une passerelle – en anglais, gateway – est un dispositif permettant de relier deux réseaux informatiques différents, comme par exemple un réseau local et l'Internet. Ainsi, plusieurs ordinateurs ou l'ensemble du réseau local peuvent accéder à l'Internet par l'intermédiaire de la passerelle. Le plus souvent, elle sert aussi de pare-feu, ce qui permet de contrôler tous les transferts de données entre le local et l'extérieur.

Phishing

Contraction des mots anglais « PHreaking », signifiant le détournement d'un système téléphonique, et de « FISHING » qui est la pêche à la ligne. Le phishing, appelé hameçonnage en français, correspond à une technique utilisée par des fraudeurs pour obtenir des renseignements personnels dans le but de perpétrer une usurpation d'identité. La technique consiste à faire croire à la victime qu'elle s'adresse à un tiers de confiance – banque, administration, etc. – afin de lui soutirer des renseignements personnels : mot de passe, numéro de carte de crédit, date de naissance, etc.

Pilote

Un pilote informatique souvent abrégé en pilote et quelquefois nommé driver est un programme, souvent accompagné de fichiers ASCII – ou fichiers « texte brut » – de configuration, destiné à permettre à un autre programme – souvent un système d'exploitation – d'interagir avec un périphérique. En général, chaque périphérique a son propre pilote. Sans pilote, l'imprimante ou la carte graphique ne pourraient fonctionner.

Port

Il s'agit d'un port logiciel mettant en œuvre un service. Il existe divers numéros de ports : par exemple 25 pour le SMTP, 110 pour le POP, 80 pour le HTTP, etc. Ce terme peut également correspondre à une prise physique permettant de connecter un périphérique.

Protocole

Dans les réseaux informatiques et les télécommunications, un protocole de communication est une spécification de plusieurs règles pour un type de communication particulier.

RAM

Random Access Memory : mémoire à accès aléatoire. Type de mémoire vive. Se référer à la définition de cette dernière.

ROM

Read-Only Memory : mémoire à lecture seule. Type de mémoire morte. Se reporter à la définition de celui-ci.

RSS

Pouvant porter les noms de Rich Site Summary, Really Simple Syndication ou encore RDF Site Summary, RSS est un format de description et de publication pour les contenus des sites Internet.

Secteur d'amorce

Zone particulière d'un disque dur ou d'une disquette permettant de démarrer le système d'exploitation d'un ordinateur. Le démarrage de l'ordinateur est appelé « boot » en anglais, d'où le terme « secteur de boot ».

Services

Terme utilisé sur Microsoft Windows. Équivalent de démon ; voir cette définition.

Système d'exploitation

Désigné par ses initiales – SE – ou, plus communément encore, par le terme anglais OS. Ensemble de logiciels permettant d'utiliser un ordinateur et ses divers périphériques. Les systèmes d'exploitation les plus connus sont Windows, Mac OS et les distributions GNU/Linux.

Terminal

Un terminal, ou « text teletype »— TTY — est une fenêtre de texte dans laquelle on peut entrer des instructions en mode texte. Ils présentent les « sorties »— c'est à dire ses réponses — uniquement sous forme textuelle et disposent simplement d'un clavier pour les « entrées »— terme utilisé pour exprimer la manière dont laquelle vous pouvez communiquer avec lui. Un exemple de terminal texte qui fut répandu en France est le Minitel, lequel est relié aux serveurs par l'intermédiaire de la ligne téléphonique.

Texte brut

Texte sans mise en forme c'est-à-dire, par exemple, sans gras, sans italique ou sans couleur.

Thème

Habillages ou thèmes d'une application, appelés « skins »— des peaux en anglais. Il s'agit de la définition de l'apparence graphique — couleurs choisies, forme des boutons, icônes...

Troll

Le terme troll est utilisé pour désigner une personne, ou un groupe de personnes, participant à un espace de discussion — de type forum par exemple — cherchant à détourner insidieusement le sujet d'une discussion pour générer des conflits en incitant à la polémique et à la provocation.

Wi-Fi

Le Wi-Fi — également orthographié WiFi, Wifi ou encore wifi — est une technologie de réseau informatique sans fil mise en place pour fonctionner en réseau interne et, depuis, devenue un moyen d'accès à haut débit à l'Internet.

ZéroConf

Zero Configuration Networking est le nom d'un ensemble de technologies permettant à plusieurs ordinateurs de communiquer sans configuration. Le but est d'obtenir un réseau IP fonctionnel sans dépendance d'une infrastructure — serveur DHCP, serveur DNS, etc. — ou d'une expertise réseau.

Zone d'amorçage

Synonyme de « MBR ». Se reporter, plus haut, à cette définition.