

# Kit de survie en lignes de commandes ...

pour les nuls.

## Pourquoi la ligne de commande ?

Posons un peu de vocabulaire : Linux est **un noyau** c'est à dire un binaire qui communique avec le matériel. Autour de lui est le système **GNU** (pour GNU is Not Unix). Le système GNU a été pensé dans les années 80 comme étant une alternative Libre au système (gratuit) mais propriétaire Unix.

Dans les années 80, à la préhistoire informatique, les systèmes d'ordinateurs n'avaient pour ainsi dire pas d'interface graphique (et encore moins de souris) : un noyau, un interpréteur de commande (shell) et quelques utilitaires suffisaient à faire un ordinateur fonctionnel.

```
64K High Memory Area is available.

This driver is provided by Oak Technology, Inc..
DTI-91X ATAPI CD-ROM device driver, Rev D91XU352
(C)Copyright Oak Technology Inc. 1987-1997
  Device Name       : 12345678
  Transfer Mode     : Programmed I/O
  Number of drives  : 1

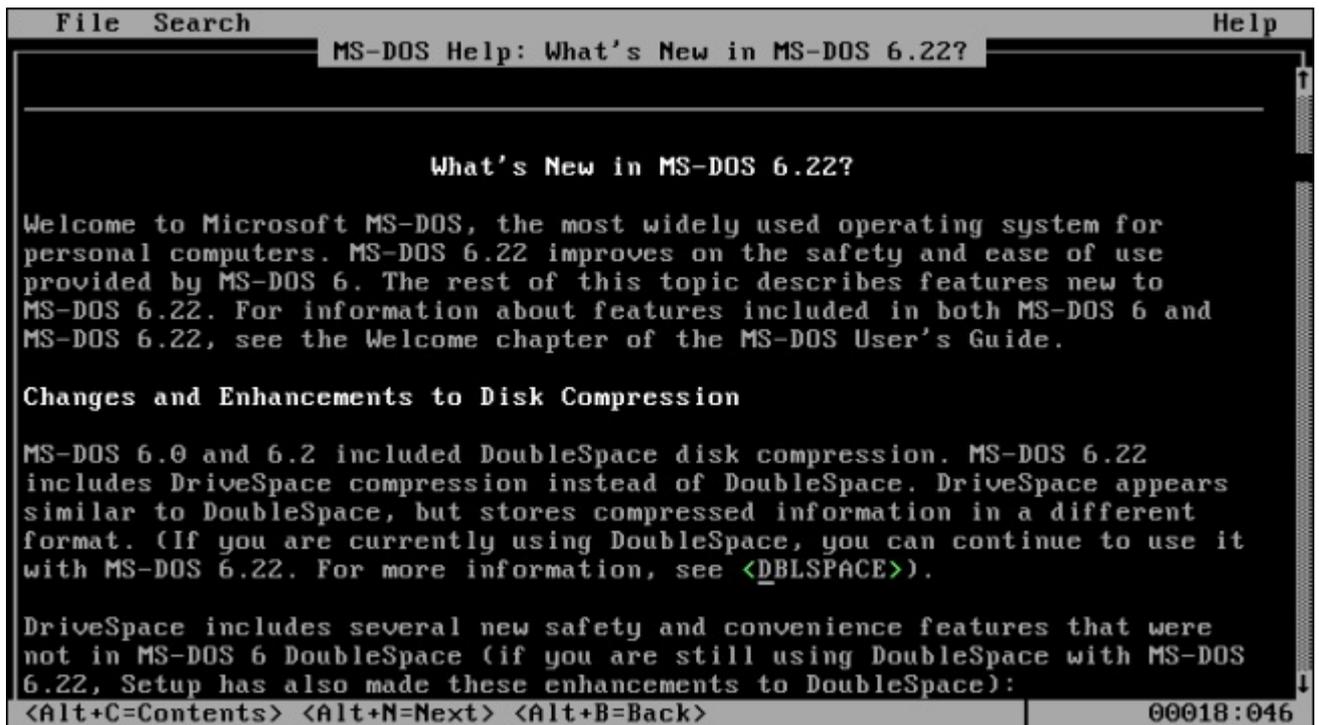
C:\>C:\DOS\SMARTDRV.EXE /X

MODE prepare code page function completed

MODE select code page function completed
MSCDEX Version 2.23
Copyright (C) Microsoft Corp. 1986-1993. All rights reserved.
  Drive D: = Driver 12345678 unit 0

CuteMouse v1.9 [FreeDOS]
Installed at PS/2 port
C:\>
```

*Ici le système Microsoft Disk Operating System fonctionnant avec les fichiers : msdos.sys et io.sys (l'équivalent du noyau), config.sys et autoexec.bat, à partir de là l'interpréteur de commande (ou shell) : COMMAND.COM s'exécute et affiche l'image précédente.*



*Dans MS-DOS on trouve des outils basiques (dont BASIC.COM) qui permettent des tâches de gestion et d'édition simple, l'image affichée ci-dessus est EDIT.COM l'éditeur de textes simple mais toujours utile.*

Rares étaient les interfaces graphiques, avant 1984 on pouvait trouver des stations Xerox, et peut être des amiga (à vérifier), passé 1984 on a eu le "Macintosh classique" d'apple, windows est devenu réellement graphique en 1990/1 (windows 3.0).

Sur le site de Toastytech on trouve d'ailleurs des captures d'écran des systèmes d'exploitation de cette époque : <http://toastytech.com/guis>

D'ailleurs de nos jours, certain·e·s utilisent encore des systèmes en mode texte sans interface graphique :

- Les ingénieur·e·s dans les datacenters
- Les administrateurs / administratrices système travaillant sous Linux, Unix, AIX ...
- Certain·e·s inconditionnels d'anciens systèmes, comme Georges R.R. Martin pour écrire son Game of Thrones (c.f. video youtube)



```

libtool: compile: x86_64-pc-linux-gnu-gcc -DHAVE_CONFIG_H -I. -I. -I../in
clude -I../include -I./ref -I/usr/include/et -pipe -O2 -march=native -D_LA
RGE_FILES=-Wall -Wmissing-prototypes -Wpointer-arith -Wbad-function-cast -Wm
issing-declarations -Wnested-externs -pipe -O2 -march=native -c file.c -fPIC
-DPIC -o .libs/libhx509_la-file.o
/bin/sh ../libtool --tag=CC --mode=compile x86_64-pc-linux-gnu-gcc -DH
AVE_CONFIG_H -I. -I. -I../include -I../include -I./ref -I/usr/include
/et -pipe -O2 -march=native -D_LARGE_FILES=-Wall -Wmissing-prototypes -Wpoi
nter-arith -Wbad-function-cast -Wmissing-declarations -Wnested-externs -pipe
-O2 -march=native -c -o libhx509_la_sel.lo `test -f 'sel.c' || echo './'`sel
.c
libtool: compile: x86_64-pc-linux-gnu-gcc -DHAVE_CONFIG_H -I. -I. -I../in
clude -I../include -I./ref -I/usr/include/et -pipe -O2 -march=native -D_LA
RGE_FILES=-Wall -Wmissing-prototypes -Wpointer-arith -Wbad-function-cast -Wm
issing-declarations -Wnested-externs -pipe -O2 -march=native -c sel.c -fPIC
-DPIC -o .libs/libhx509_la_sel.o
/bin/sh ../libtool --tag=CC --mode=compile x86_64-pc-linux-gnu-gcc -DH
AVE_CONFIG_H -I. -I. -I../include -I../include -I./ref -I/usr/include
/et -pipe -O2 -march=native -D_LARGE_FILES=-Wall -Wmissing-prototypes -Wpoi
nter-arith -Wbad-function-cast -Wmissing-declarations -Wnested-externs -pipe
-O2 -march=native -c -o libhx509_la_sel_gram.lo `test -f 'sel-gram.c' || ech
o './'`sel-gram.c
libtool: compile: x86_64-pc-linux-gnu-gcc -DHAVE_CONFIG_H -I. -I. -I../in
clude -I../include -I./ref -I/usr/include/et -pipe -O2 -march=native -D_LA
RGE_FILES=-Wall -Wmissing-prototypes -Wpointer-arith -Wbad-function-cast -Wm
issing-declarations -Wnested-externs -pipe -O2 -march=native -c sel-gram.c
-fPIC -DPIC -o .libs/libhx509_la_sel_gram.o
HOST=x86_64-pc-linux-gnu*
x86_64_pc_linux_gnu_CFLAGS=-pipe -O2 -march=native*
i686_pc_linux_gnu_CFLAGS=-pipe -O2 -march=native*
case `s(CATEGORY)/$(PRN)` in
NORMAL >> /etc/paludis/bashrc < sh << 9% : 1: 1
[exbull:0] [1:vim] 2:zsh |

```

```

1 | [|||||||||||||||||] 80.9% | Tasks: 46, 8 thr; 1 running
2 | [|||||||||||||||||] 77.5% | Load average: 1.07 0.62
Mem | [|||||||||||||||||] 103/2476 | Uptime: 00:21:04
Swp | [|||||||||||||||||] 7/15/204 |

```

PID	USER	PRI	NI	VIRT	RES	SHR	S	CPU%	MEM%	IORW	TIME+	Command
7583	paludisbu	20	0	8652	2112	1788	S	59.3	0.8	0	1:26.62	sydbox .
72	root	20	0	19132	2452	2304	S	0.0	1.0	0	0:01.11	/usr/lib
271	tureba	20	0	23928	6660	2388	S	0.0	2.6	0	0:04.87	tmux -u2
14177	paludisbu	20	0	6952	2400	1736	S	0.0	0.0	0	0:00.03	make all
12147	root	20	0	480M	19820	15032	S	0.0	7.8	0	0:04.67	cave exe
16659	tureba	20	0	14272	2920	2404	R	0.0	1.2	0	0:00.76	htop
14830	paludisbu	20	0	6980	2316	1656	S	0.0	0.9	0	0:00.04	make all
235	tureba	20	0	76444	3464	2740	S	0.0	1.4	0	0:01.78	sshd: tu
7584	root	20	0	118M	17788	15168	S	0.0	7.8	0	0:00.11	cave per
994	tureba	20	0	29212	8012	4508	S	0.0	3.2	0	0:00.22	vim /etc
26606	root	20	0	118M	17788	15168	S	0.0	7.8	0	0:00.59	cave per

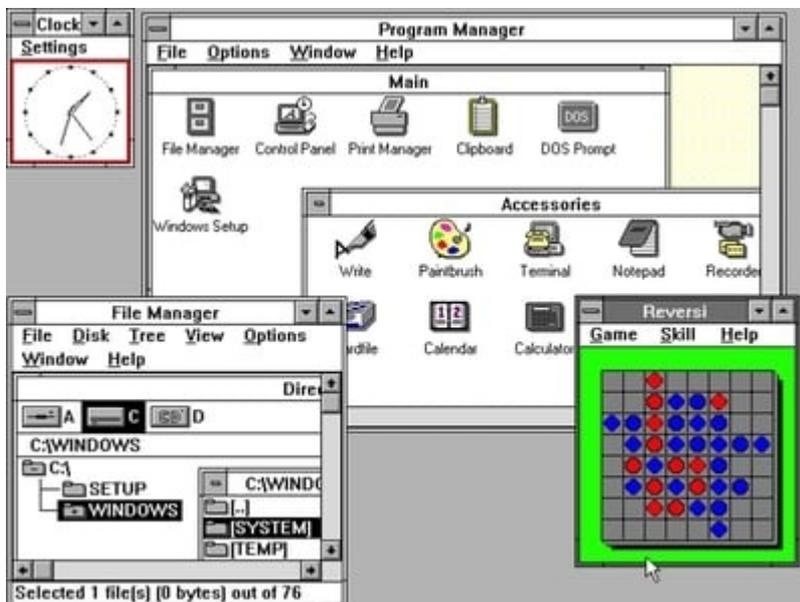
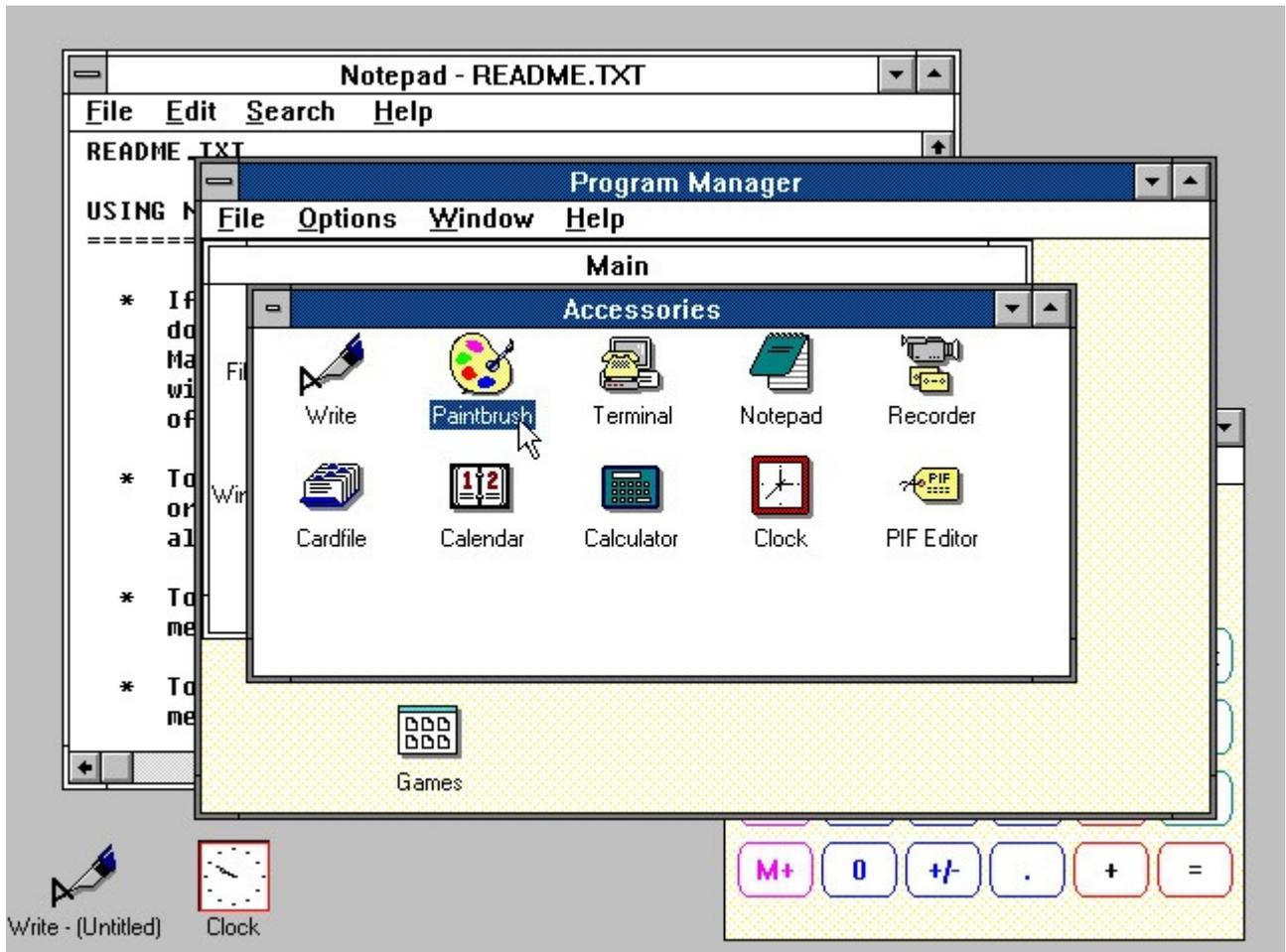
```

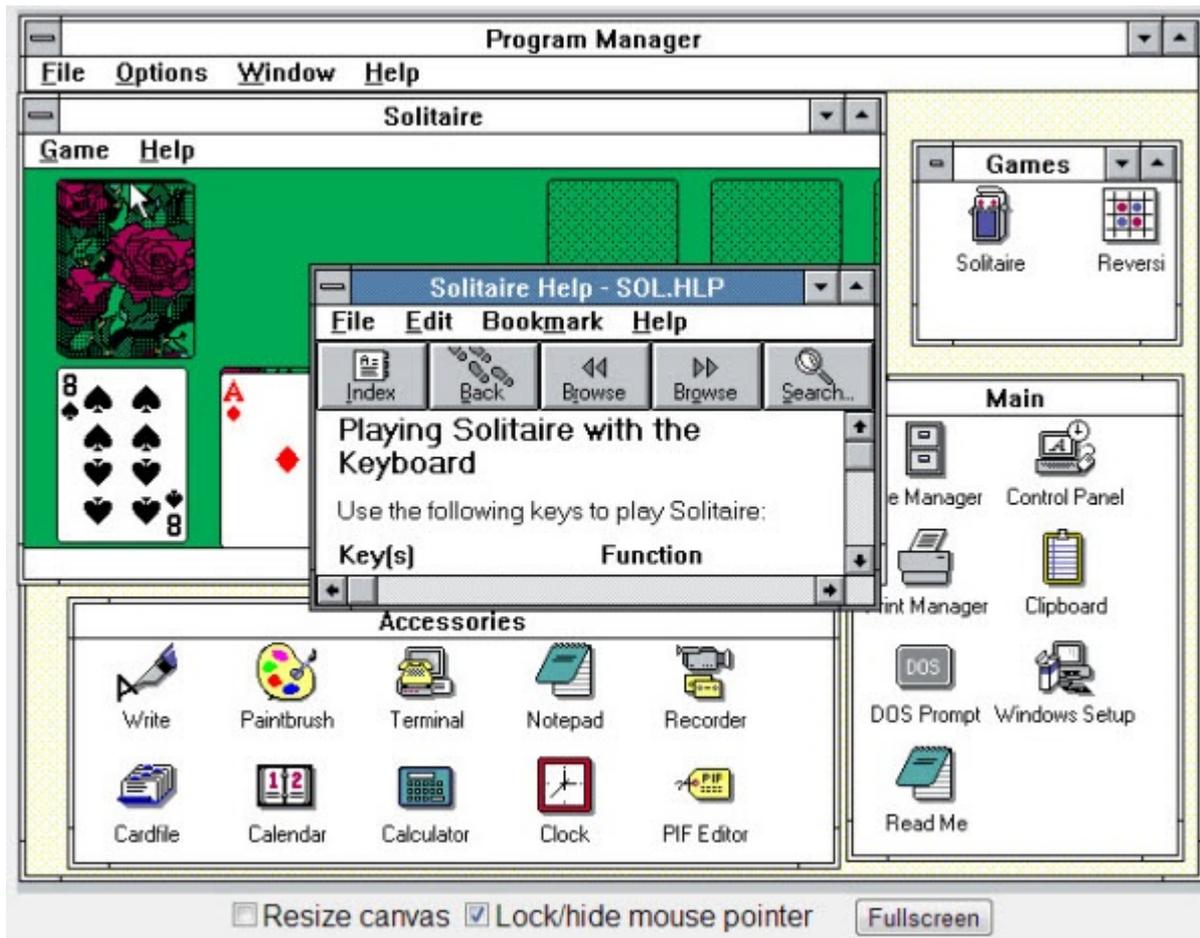
[11:04:40|1023] (tureba@exbull)% cd ../ompi
[11:04:46|1024] (tureba@exbull)% ls
tree
AUTHORS      Makefile.am  VERSION      config.lt    libtool
Doxyfile     Makefile.in  aclocal.m4   config.status mps
HACKING      Makefile.rules autogen.pl   configure.ac  ortc
INSTALL      NEWS         autowrite.cache  configure.ac  ortc
LICENSE      README      contrib       config        oshmc
Makefile     README.JAVA.txt  config.log     examples     test
[11:07:11|1025] (tureba@exbull)%
[1025][11:07:32|1025] (tureba@exbull)%
5 ~/om[11:09:44|1025] (tureba@exbull)%
[11:09:52|1025] (tureba@exbull)%

```

- Des gens qui ont des machines à très faibles capacités et ont des besoins spartiates en terme d'interfaces graphiques voire aucune.

A cette époque, et hormis les macintosh et les amiga (avec GEM) les interfaces graphiques ne sont que des surcouches du système, ainsi les "PC" grand public sont livrés avec MS-DOS (ou d'autres DOS jusqu'à ce que Microsoft bloque le démarrage de windows s'il détecte un DOS autre que le sien en cours d'exécution, c.f. l'affaire "4DOS")... pour un démarrage automatique de windows lors de l'allumage de l'ordinateur il fallait ajouter la ligne "win" à la fin du fichier C:\AUTOEXEC.BAT (oui je sais, ça fait vieux barbu).

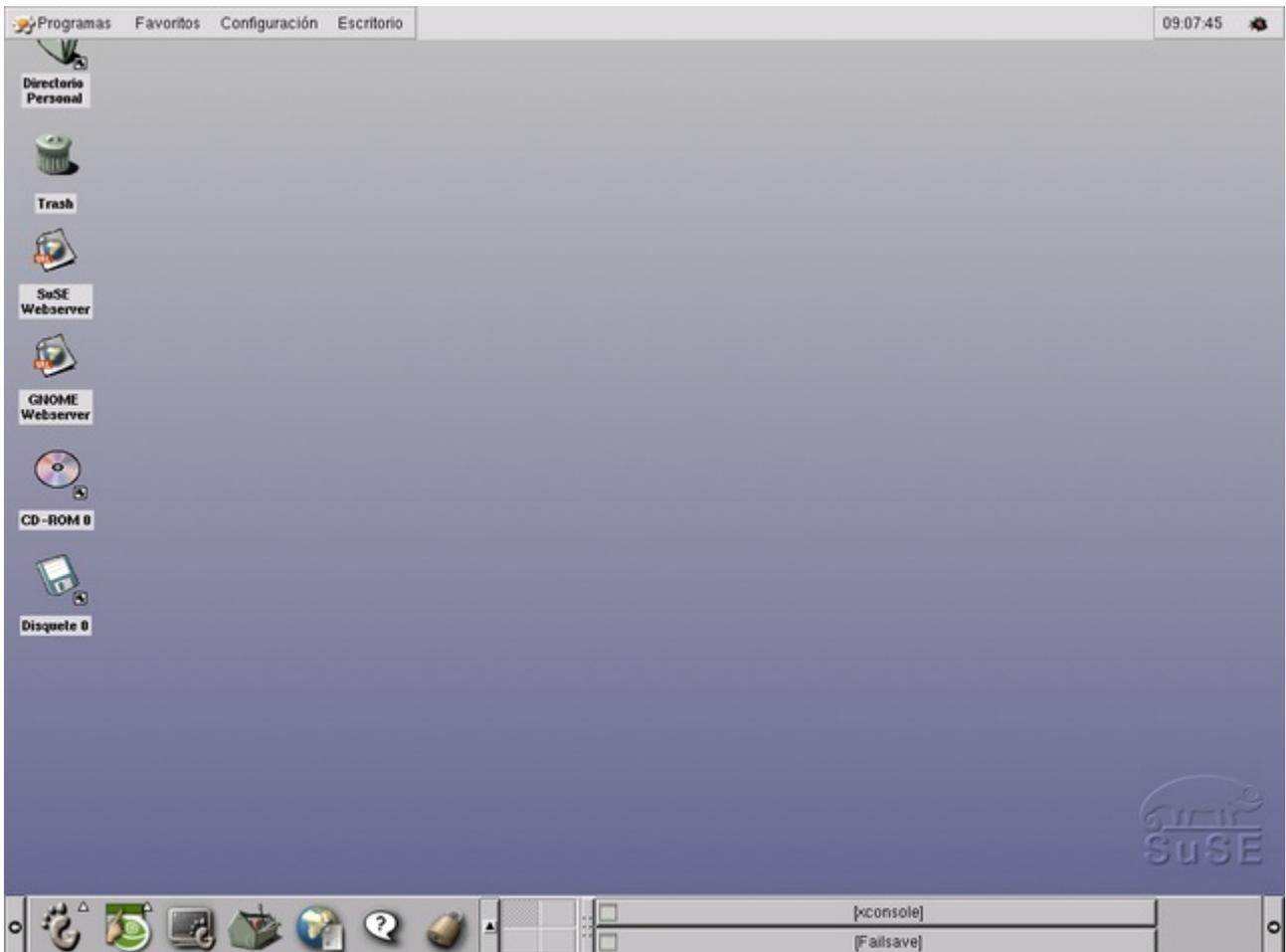
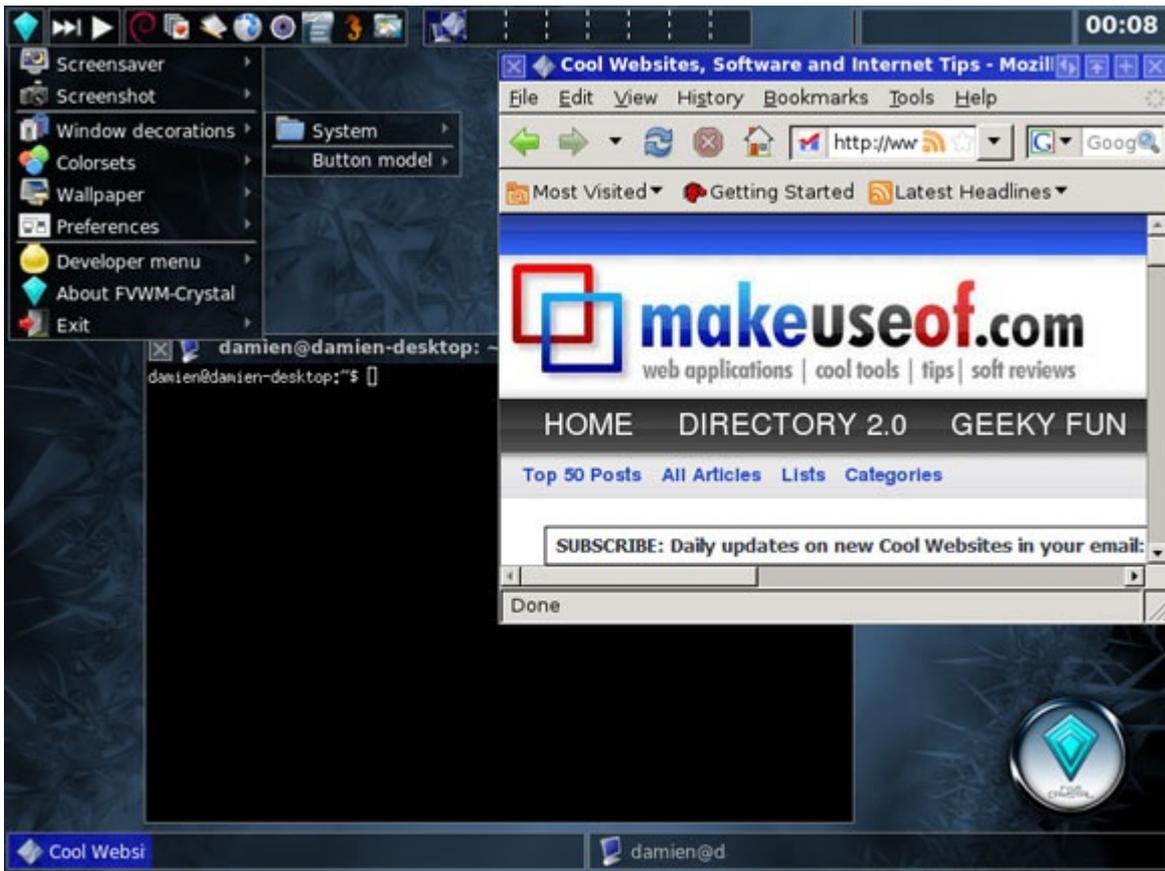


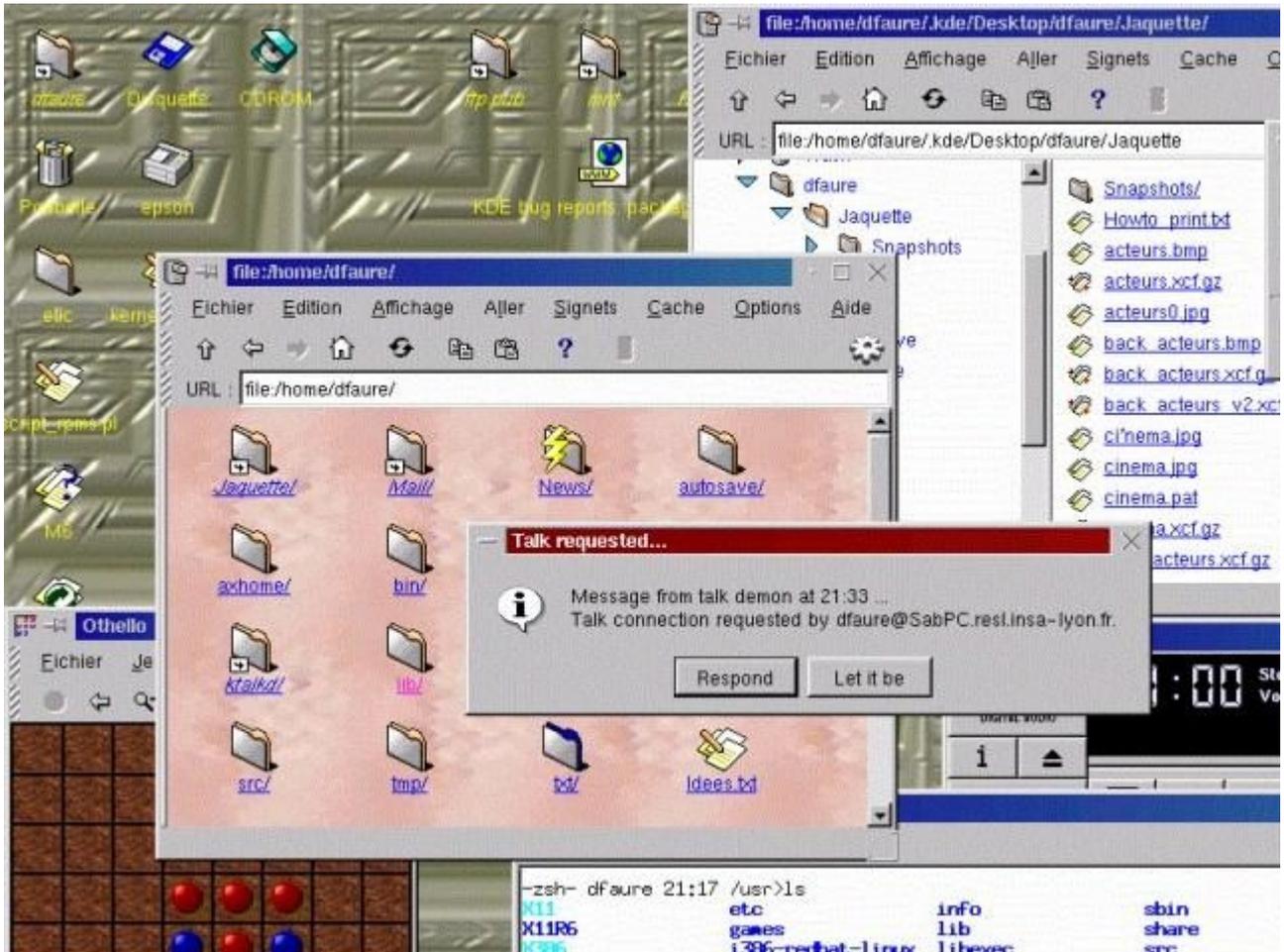


*Quelques captures d'écran de Windows 3.0 qui fût en son temps une révolution sur PC.*

GNU/Linux ont donc été créés à cette époque là. Et, malgré l'arrivée du serveur X Window / XFree86, et comme Unix/Linux étaient plutôt utilisés par des ingénieurs et des passionnés, l'interface graphique n'a pas été leur intérêt. Si bien qu'au milieu des années 90 peu d'interfaces graphiques existent : TWM, KDE, GNOME, sawfish, GNUStep, ICEWm, blackbox ... Le soucis a toujours été à l'époque comme celui que nous expérimentons en tant que libristes actuellement : que ce soient les SoC ou les cartes propriétaires wifi, il faut toujours trouver un driver quelque part, alors imaginez vous dans les années 90, avec une connexion à 56 kbps (7 ko/s) télécharger un fichier de quelques mégas pour un driver sur une ligne analogique qui est inutilisable tant qu'on reste connecté à internet...

L'intérêt progressif du public et de développeurs et développeuses pour les Linux s'est traduite progressivement par plus de développements et désormais une jungle d'interfaces. Au milieu des années 90 commencent à arriver les premières applications graphiques et interfaces graphiques dignes de ce nom, TWM, XFCE, FVWM, GNUStep, KDE, GNOME, metacity, ICEWm, blackbox ... mais hormis ces interfaces, peu de logiciels graphiques " grand public ". Le seul outil réellement développé est le terminal graphique où peuvent ainsi s'exécuter les logiciels dont les gens ont l'habitude.





Welcome to LWN.net [LWN.net] - SeaMonkey

File Edit View Go Bookmarks Tools Window Help

Back Forward Reload Stop <https://lwn.net/> Search Print

Home Bookmarks Most Visited SeaMonkey mozilla.org mozillaZine mozdev.org

 [Weekly edition](#) [Kernel](#) [Security](#) [Distributions](#) [Contact Us](#) [Search](#)  
[Archives](#) [Calendar](#) [Subscribe](#) [Write for LWN](#) [LWN.net FAQ](#) [Sponsor](#)

## Welcome to LWN.net

### Headlines for March 22, 2011

**Stable kernel 2.6.33.8 released**  
 [Kernel] Posted Mar 21, 2011 21:44 UTC (Mon) by ris

Logged in as zonker  
[My Account](#)  
[Unread comments](#)  
[Author Portal](#)  
[Log out](#)

**Recent Features**  
[Has Bionic stepped over the GPL line?](#)  
[LWN.net Weekly Edition for March 17, 2011](#)  
[Taking openSUSE 11.4 for a spin](#)  
[The Grumpy Editor's GNOME 3 experience](#)  
[LWN.net Weekly Edition for](#)

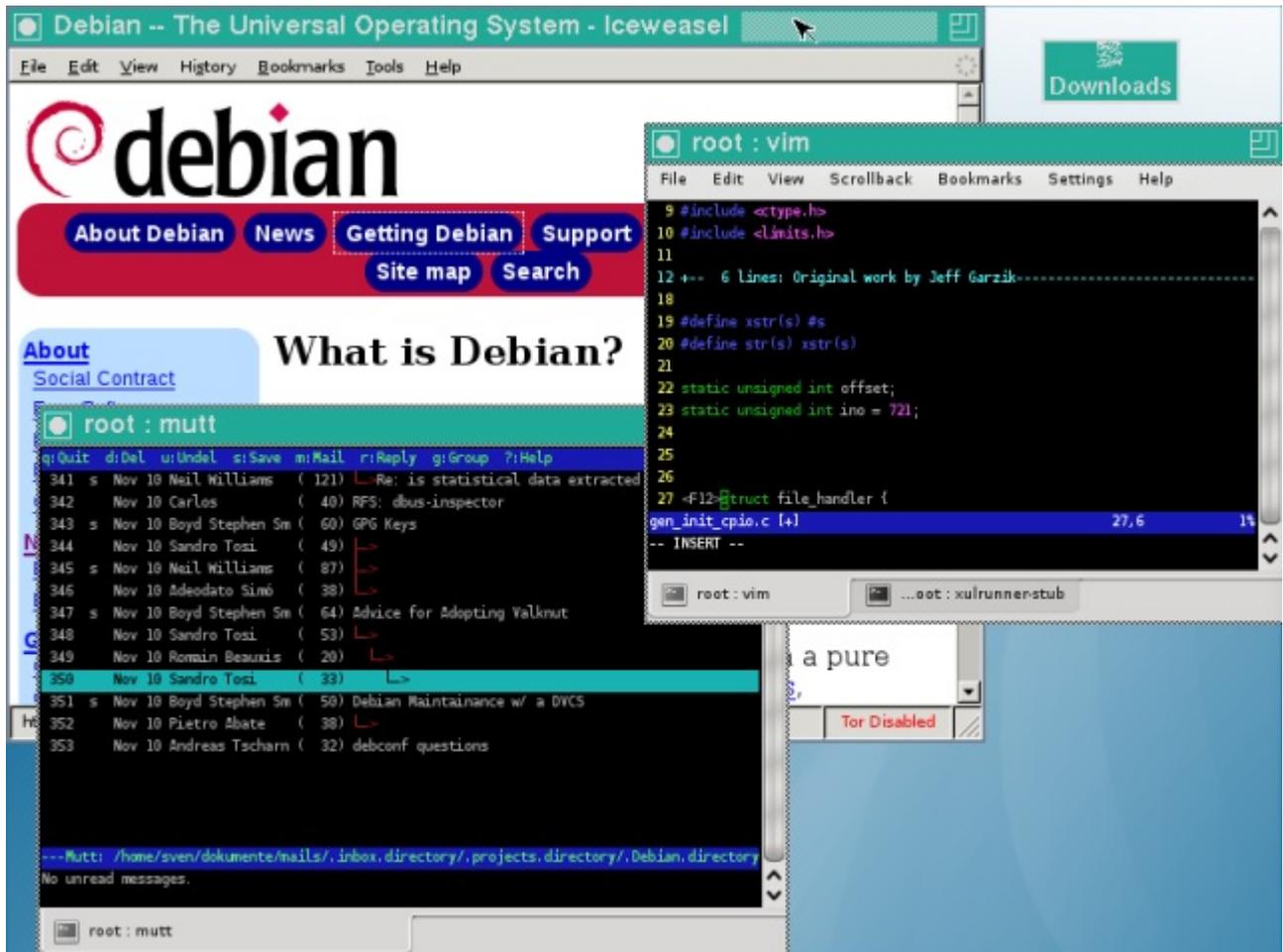
```

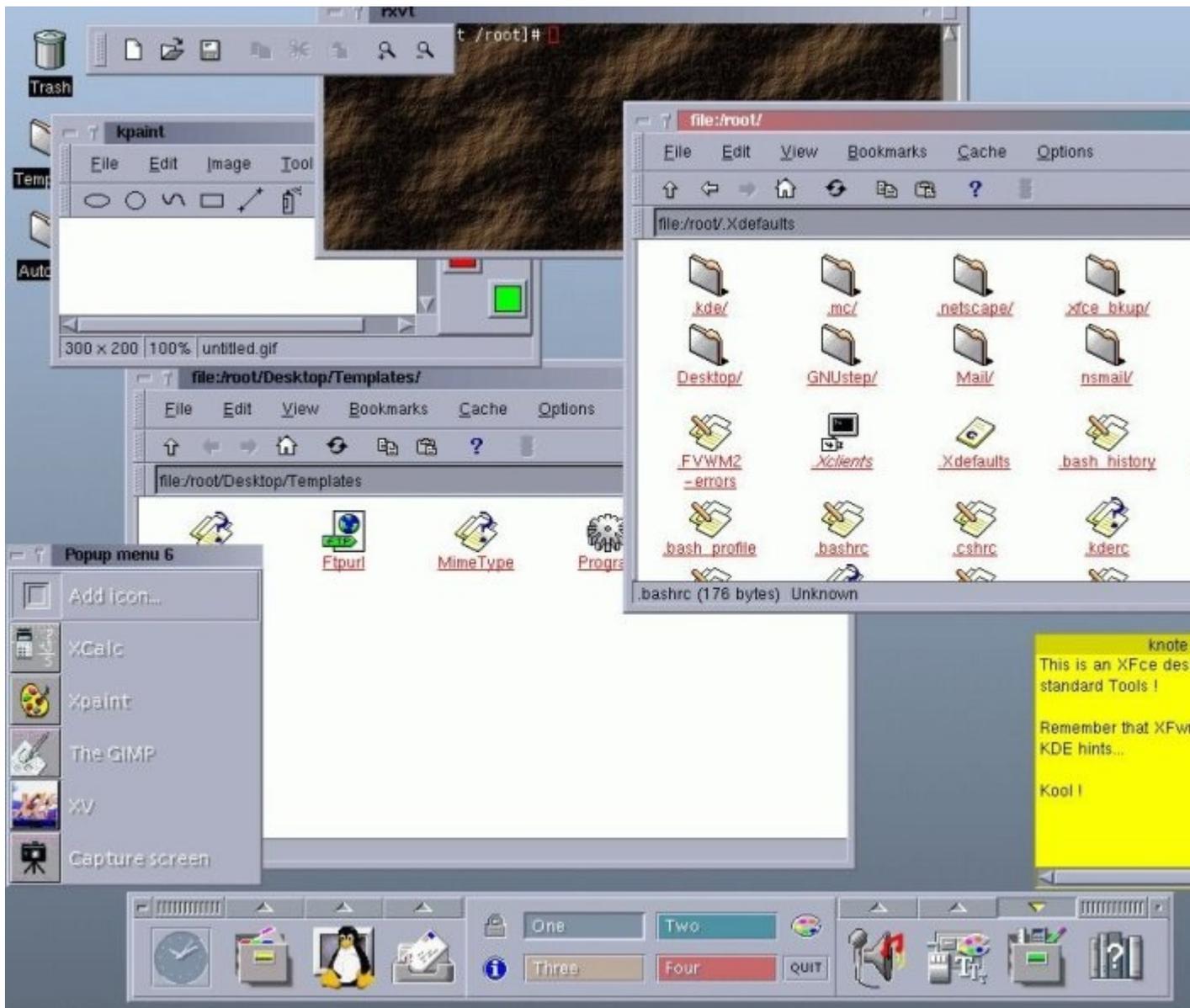
xterm
top - 23:40:47 up 1 day, 1:35, 2 users, load average: 0.01, 0.02, 0.05
Tasks: 105 total, 1 running, 104 sleeping, 0 stopped, 0 zombie
Cpu(s): 0.0%us, 0.0%sy, 0.0%ni,100.0%id, 0.0%wa, 0.0%hi, 0.0%si, 0.0%st
Mem: 1019476k total, 766728k used, 252748k free, 163648k buffers
Swap: 965992k total, 0k used, 965992k free, 424508k cached

  PID USER      PR  NI  VIRT  RES  SHR  S  %CPU  %MEM    TIME+  COMMAND
 5387 jzb       20   0 318M  98M  28M  S   0.8   1:22.37  seasonkey-bin
    1 root      20   0   824  304  260  S   0.0   0:02.24  init
    2 root      20   0     0     0     0  S   0.0   0:00.00  kthreadd
    3 root      20   0     0     0     0  S   0.0   0:00.09  ksoftirqd/0
    4 root      20   0     0     0     0  S   0.0   0:00.00  kworker/0:0
    6 root      RT   0     0     0     0  S   0.0   0:00.00  migration/0
    7 root      RT   0     0     0     0  S   0.0   0:00.00  migration/1
    8 root      20   0     0     0     0  S   0.0   0:00.00  kworker/1:0
    9 root      20   0     0     0     0  S   0.0   0:00.09  ksoftirqd/1
   10 root      20   0     0     0     0  S   0.0   0:20.35  kworker/0:1
   11 root      0  -20     0     0     0  S   0.0   0:00.00  cpuset
   12 root      0  -20     0     0     0  S   0.0   0:00.00  khelper
   13 root      20   0     0     0     0  S   0.0   0:02.05  kworker/u:1
   16 root      0  -20     0     0     0  S   0.0   0:00.00  netns
  279 root      20   0     0     0     0  S   0.0   0:00.28  sync_supers
  281 root      20   0     0     0     0  S   0.0   0:00.05  bdi-default
  283 root      0  -20     0     0     0  S   0.0   0:00.00  kblockd
  
```

Nook  
 ed in  
 ral  
 ad the

Desktop





Il est à noter que chacun a ses petits choux-choux en terme d'interfaces graphiques même si nombreux sont les gens qui restent sur l'interface "par défaut" ou choisissent leur distribution en fonction du *look* qu'elle montre par défaut (d'où à mon avis l'un des critères de la réussite d'ubuntu).

Cependant, il y a toujours eu des "pro" qui ne jurent que par la ligne de commande pour des raisons diverses et variées.

L'objet de la présentation actuelle est de vous donner quelques commandes de base et quelques exemples pratiques pour vous familiariser à la manipulation des dossiers et fichiers, quelques outils d'édition et de mises à jour. Pourquoi ?

Tout simplement car lorsqu'on est un-e habitué-e des interfaces graphiques les raisons d'utiliser une interface textuelle sont (entre autres) :

- une panne de l'interface graphique
- un goût pour certaines applications plus rapides, moins gourmandes en mode texte qu'en mode graphique (personnalisation à la mode h4ck3rzzz) et une

personnalisation à l'extrême du système avec uniquement les "paquets nécessaires".

- un pc avec des capacités faibles qui permet d'exécuter peu de logiciels graphiquement mais qui supporte aussi quelques applications textuelles complémentaires.

Historiquement, les années 80 et début 90 offrent aux utilisateurs de PC à but professionnel un logiciel de démarrage, un noyau, un shell et quelques applications. Cela suffit amplement pour beaucoup de gens.

Avant de commencer à faire mumuse dans le terminal, il est nécessaire de reposer quelques bases de l'organisation hiérarchique du système de fichier, car autant dans les interfaces graphiques en comparant celle des 3 systèmes qui en utilisent (je passerai outre les systèmes de téléphones portables ou des tablettes iOS où justement il n'y a plus que des icônes accédant directement aux données mais où le système de fichier sous-jacent est masqué à l'utilisateur basique.

Critère	Windows	Linux / Unix	Mac Operating System / MacPS
Le système (ordinateur)	"Ordinateur" ou "Bureau (XP)"	/	/
Désignation des systèmes attachés non-amovibles	C:\ D:\ ... Z:\ (sauf A:\ et B:\ réservés par défaut aux disquettes)	/dev/sda.. /dev/sdb... /dev/hd... (vieux linux) /dev/mmcblk0p... le premier de ces systèmes est accroché à "/"	/ (je n'ai jamais eu à tester des systèmes mac sur plusieurs DD)
Désignation des systèmes amovibles	A:\ B:\ D:\ E:\ ... Z:\ (toujours A et B réservés aux disquettes)	/mnt/device /media/device /run/media/device (vieux) /mnt/login/device /media/login/device /run/media/login/device ...	/Volumes/device
Localisation des dossiers des utilisateurs.	C:\Documents and Settings (XP)\login C:\Utilisateurs\login C:\Users\login	/home/login	/users/login
Les programmes	C:\Program Files C:\Program Files_(x86)	/bin /sbin /usr/bin /usr/sbin /usr/local/bin /usr/local/sbin	?????
	C:\Users C:\Utilisateurs C:\Documents and Settings Soit dans fichiers soit		

Les réglages "par défaut" des programmes

/etc/ ou /usr/lib ou /home/login/.config ou /home/login/.local

/Users/login/.Library  
?

## Où configurer le shell des utilisateurs ?

Ce paramètre se configure dans `/etc/passwd`

Note : Si on regarde les lignes `/usr/sbin/nologin`

Donc on commence à ouvrir un terminal pour nous entraîner.

Où démarre le shell du terminal ? Il démarre dans le dossier personnel de l'utilisateur donc `/home/login`. et si c'est l'utilisateur `root` à savoir l'administrateur c'est dans le dossier `/root`.

## Quelques commandes pour commencer :

Une majorité des commandes saisies par la suite aura entre 1 et 3 lettres, c'est un héritage d'UNIX.

### Où suis-je ? :

Par la commande (qui sera utilisée pas mal dans les essais suivants) `pwd` pour Print Working Directory (Imprimer à l'écran le dossier de travail actuel).

```
pwd
```

### Qui suis-je ? :

```
who et / ou whoami
```

### Qui est connecté au système ? :

C'est le rôle de la commande `w`

```
`w`
```

## je veux aller au dossier `/usr/share` (par exemple) ou revenir à la maison :

Pour effectuer le changement de répertoire (ou dossier) on utilise la commande `cd`. "Change Directory". Regardez les effets en tapant les commandes suivantes les unes après les autres et en saisissant la commande `pwd` entre 2 `cd`.

```
cd /usr/share
```

```
pwd
```

```
cd ..
```

```
pwd
```

cd .

pwd

cd

**je veux afficher l'organisation des disques :**

lsblk

**je veux afficher l'état de la mémoire :**

free

**Quel est l'état d'occupation du disque ? :**

df

**Quelles sont les tâches en cours d'exécution ? :**

(h) top et ps

**Quels sont les fichiers et "nodes" ouverts ? :**

lsof

Note : Il y a l'autocomplétion par la touche [ tab ] qui parcourt alphabétiquement les différents choix possible qui complètent le début saisi...

Deux liens physiques importants à connaître : ./ = dossier actuel, ../ = dossier parent.

**Maintenant entraînement à manipuler des dossiers :**

Je veux créer un dossier Mars comme sous-dossier du dossier actuel : mkdir Mars (attention à la casse).

Tel quelle la commande mkdir oblige à créer les sous-arborescence les unes après les autres mais en passant quelques options on peut créer tout en une seule fois.

```
mkdir -p Mars/Satellites/Phobos
```

Depuis le dossier actuel je veux aller dans le dossier Phobos.

```
cd Mars/Satellite/Phobos ou bien :
```

```
cd ../Mars/Satellite/Phobos et bien sûr pwd pour voir où vous êtes.
```

Je veux remonter de plusieurs dossiers en une fois :

1. retour dans Phobos : cd Phobos
2. remontée des deux dossiers : cd ../.. suivi de pwd.

Je veux remonter d'un dossier : cd .. la preuve par pwd

ET retourner maison

ex : `cd /usr/share/icons` puis vérification par `pwd` suivi de `cd` suivi de `pwd` pour vérifier à la fin.

Petit entraînement et préparation de la suite, saisie des caractères non reconnus nativement et des espaces :

```
cd
mkdir Mars/Satellites/Deimos
mkdir Mars/Satellites/Belle Mere
mkdir Mars/Satellites/Belle\ Soeur
mkdir 'Mars/Satellites/Beau Pere'
mkdir "Mars/Satellites/Beau Frere"
echo "Salut" > machin
```

Note : `cd` tout seul permet de revenir au dossier de base de l'utilisateur.

Note 2 : la dernière ligne commençant par `echo` sera expliquée plus tard.

### **Créer plusieurs sous-dossiers en une seule fois :**

```
mkdir {Jupiter,"Mini moi"}/Satellites/{Io,Europe,Ganymede,Callisto,Bidule
```

Attention quand on supprime c'est d é f i n i t i f !

### **Supprimer un dossier non vide et / ou une sous-arborescence**

```
rmdir Mini\ moi
```

cela crée une erreur, du coup :

```
rm -fR ./Mini\ moi
```

suivi de `ls -lh` pour vérifier.

Attention au gag : en tant que root, taper un `rm -fR /` efface la totalité du disque dur :p

### **La commande `ls`, `.` et `..`**

```
cd Mars
```

```
mkdir ../{Gaia,Venus,Mercure}
```

```
cd ..
```

```
ls
```

```
ls -l
```

```
ls -la
```

```
ls -lh
```

```
ls -lah
```

```
ls -rl
```

comment analyser la sortie d'un ls -l ?

```
ls -lh
```

```
...
```

```
drwxr-xr-x 3 anonymous anonymous 4,0K mai 4 23:47 Jupiter  
-rw-rw-r-- 1 anonymous anonymous 6 mai 4 23:56 machin
```

```
...
```

type d'entr ée	droits propri étaire	droits group e propri étaire	droits autre s	nombr e de liens (dossi ers + . + ..)	propri étaire	group e propri étaire	taille sur disqu e	date de derniè re modifi cation	heure de derniè re modifi cation	nom du fichier
d	rwX	r-x	r-x	3	anony mous	anony mous	4,0K	mai 4	23:47	Jupiter
-	rw-	rw-	r--	1	anony mous	anony mous	6	mai 4	23:56	machin

le - veut dire que le fichier n'a rien de spécial (c'est un simple fichier d'archives) d -> dossier, S -> Suid bit activé r -> lecture x -> exécution (ou parcours d'un dossier) w -> écriture

## Les informations de droits sont modifiables par la commande

chmod (change mode)

## Les informations de propriété sont modifiables par la commande

chown (change owner)

## rendre un fichier invisible :

```
ls -l
```

```
mv machin .machin
```

```
ls -l
```

à retenir : `mv` = move (déplacement du fichier).

à retenir 2 : un fichier "invisible" sous linux est un fichier (ou un dossier) dont le nom commence par ".".

## Créer un fichier ligne par ligne directement depuis le shell sans éditeur :

```
cd
cd Jupiter
echo Ceci est le dossier de Jupiter > Readme
less Readme
```

sortie de less par la touche "Q" (note : c'est la même chose quand on a une mise à jour critique ou grave pour raison de sécurité, un message de post-installation s'affiche, et pour quitter on appuie sur "Q").

```
cat Readme
```

vous voyez la différence entre less et cat.

```
echo ----- > Readme
```

regardons le résultat :

```
cat Readme
```

et oui ! Le début a été effacé. Utilisons la touche [flèche haut] pour revenir à la ligne `echo Ceci est le dossier de Jupiter > Readme` puis ensuite tapez `echo ----- >> Readme` et regardons par un `cat Readme`.

```
echo "Bonjour"
```

*A Retenir : less et cat permettent d'afficher des contenus de fichiers, echo permet soit d'afficher du texte, soit d'écrire un texte vers un périphérique ou un fichier en mode écrasement > ou ajout >> .*

## Edition par bloc

```
echo "
`> écrivez ici le texte que vous voulez mais sans pouvoir`
`> revenir à la ligne précédente. Chaque nouvelle ligne est`
`> précédée du signe > mais cela permet quand même d'éditer`
`> un fichier au fur et à mesure et quand c'est fini on a 3`
`> choix disponibles : annuler par [CTRL]+[Z]`
```

```
`> finir et afficher le texte à l'écran par [CTRL]+[D]`  
`> finir et enregistrer dans un fichier en une seule fois`  
`> par la redirection finale " >> Readme`
```

et le résultat obtenu par : `cat Readme`.

### **Copier un fichier (Readme) dans un autre dossier**

```
cp Readme ../Mars/
```

L'option "-v" rend la commande verbeuse.

### **Copier un dossier et son contenu en une fois**

```
cd ..  
pwd (vérification si on est au dessus de Jupiter)  
cp -r Jupiter Saturne  
ls -l Saturne (pour vérifier que tout à été copié).
```

### **Effacer un dossier et son contenu**

```
rm -fR Saturne
```

### **Filtrer une sortie d'après un mot clé :**

```
grep
```

ex : Afficher tous les paquets de debian contenant le mot clé "theme" puis filtrer les sorties pour n'afficher que celles contenant "gtk"

```
apt-cache search theme | grep gtk
```

Maintenant qu'on a fini de faire mumuse avec les sorties, les fichiers et les dossiers on va passer aux commandes dangereuses.

### **Les téléchargements :**

télécharger en mode texte :

```
wget -c ....
```

```
curl -L -C- -O ....
```

### **Passer en super utilisateur / administrateur**

si sudo disponible :

```
sudo su
```

si sudo pas disponibles :

su -

### **exécuter 1 commande unique en superutilisateur :**

sudo commande options

su -c "commande options"

ATTENTION : En tant que root vous avez le pouvoir de vie et de mort sur le système et il faut savoir que les applications unix ne disent rien si tout s'est passé comme prévu (si elles donnent des informations c'est qu'il y a une erreur). Si vous voulez forcer le bavardage des commandes alors il faut passer l'option "-v" la plupart du temps (v pour verbose c'est à dire verbeux).

### **Enchaînement des commandes : & et &&**

& : les commandes s'exécutent en parallèle dans des sous-shells

&& : la seconde commande s'exécute si la première renvoie un code d'erreur "0" (donc pas d'erreur).

### **Avoir de l'aide sur une commande**

man commande

### **Mettre à jour le système**

debian et clones (\*buntu ; lxde; !#; kali; tails; ...) :

apt update  
apt upgrade

redhat et fedora et clones

yum  
dnf

dnf system-upgrade clean  
dnf clean all  
dnf upgrade  
dnf system-upgrade download --releasever=28 --allowerasing --best  
dnf system-upgrade reboot

les distributions "snap"-isées

snap refresh

**archlinux, parabola gnu/linux-libre, manjaro (pour les paquets qui ne sont pas dans AUR)**

pacman -Syu

et pour les paquets dans AUR en plus des autres paquets des dépôts par défaut.

yaourt -Syua

**voidlinux**

xbps-install -Su

## **nettoyer le cache des paquets**

.deb : apt clean

arch & co : pacman -Sc / (yaourt -Sc ?)

void : xbps-remove -O (attention à la casse)

fedora, redhat :

snap-isées :

## **Vider les versions obsolètes et / ou le cache des paquets et orphelins**

deb : apt autoremove --purge (orphelins / dépendances inutiles) apt-get autoremove --purge (raspi, vieilles deb)

aptitude / dselect -> applications en curses comparables graphiquement à synaptic / ksynaptic

pacman -Scc

xbps-remove -Oo

fedora

snaps

## **supprimer un paquet unique (avec ou sans dépendances)**

deb apt remove --purge nom-du-paquet apt autoremove --purge (pour les dépendances)

pacman -Rs pacman -Rdd

xbps-remove paquet

snap remove paquet

federa : dnf / yum0 redhat : dnf / yum

## **monter / démonter un dossier**

## **Quelques utilitaires**

- mc

mc est un gestionnaire de fichiers à double panneau, pour les nostalgiques de Norton commander. Il est très pratique pour bon nombre d'actions et se montre plus rapide et plus bavard que certains outils graphiques.q

- nano
- moc / ncpdcpb
- alsamixer
- links / lynx / w3m
- dhcpcd / nm-cli / wicd
- sshfs
- ssh
- fusermount
- gpg
- systemctl / sv
- ln

## **Gérer les périphériques et les liens**

Formatage, partitionnement, vérification, gravure d'iso, création d'iso...

### **Partitionnement de périphériques (disques, clés, etc...)**

**Partitionner un disque (inférieur à 2,3 To)**

`fdisk /dev/...`

`cgdisk /dev/...`

**Partitionner un disque (supérieur à 2,3 To)**

### **Formater une partition**

**type ext**

**type swap**

**Formater une clé (par exemple) en FAT32 ou EFI**

```
mkfs.vfat -n NOM /dev/sd..
```

**Créer un lien symbolique**

```
ln -s cible alias
```

**Graver une image**

```
dd status=progress ... if=... of=....
```

**Monter un périphérique manuellement**

un périphérique réel

un ramdisk

une ISO en loop

**Éteindre et redémarrer**

suivant les distributions le droit d'éteindre ou de redémarrer le système n'est pas permis directement à l'utilisateur lambda. D'autres distributions permettent l'accès à cette commande. La commande est " shutdown " à la quelle peuvent être passés des arguments divers. Il y a bien sûr la commande "init" ou la commande "telinit" aussi.

Voici quelques options.

**Extinction directe du système :**

```
shutdown -h now ou /sbin/telinit 0 ou init 0
```

**Extinction différée du système dans un délais (en minutes) :**

```
shutdown -h +10
```

**Extinction différée du système à une heure spécifique :**

```
shutdown -h -t 23:35:00 (à vérifier)
```

**Redémarrage du système**

```
shutdown -r now ou /sbin/telinit 6 ou init 6
```

---

Sources : manpages <http://toastytech.com/guis/>