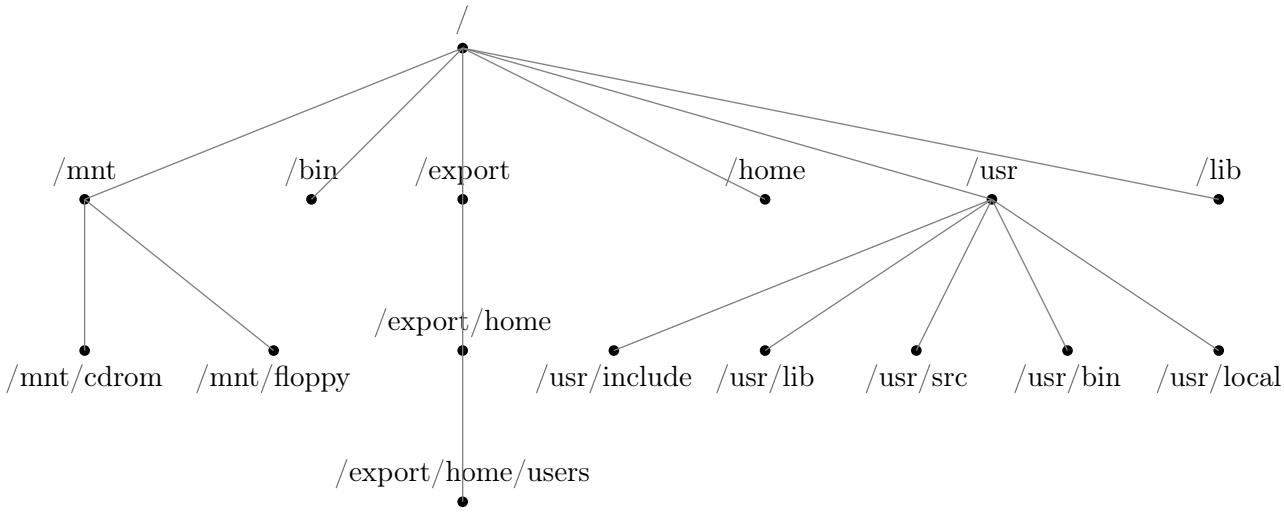
1 Guide de survie Unix

1.1 Des fichiers et des commandes

1.1.1 Arborescence Unix

Dans le monde Unix, tout est fichier (un fichier texte, un exécutable, un driver, un lien, un répertoire, sont tous des fichiers). Ces fichiers sont organisés dans le système de fichiers sous forme d'une arborescence dont la racine est notée "/". Les fichiers sont donc rangés dans des répertoires et ces répertoires sont organisés sous forme d'un arbre. Ainsi, le système de fichiers comporte une racine, tout répertoire (sauf la racine) a un répertoire père unique, et tout fichier appartient à un unique répertoire. Usuellement, on retrouve l'arborescence (partielle !) suivante :



1. /mnt : répertoire des lecteurs amovibles (disquette, clé usb, CD,...).
2. /bin : répertoire des commandes Unix.
3. /usr : répertoire des principaux utilitaires (notamment : gtk !).
4. /usr/local : répertoire des utilitaires installés *localement* à la machine.
5. sous-répertoires utiliaires bin, lib, src, include : binaires (exécutables), bibliothèque (ex, /usr/lib/libgtk-x11-2.0.la), source (code source des utilitaires), entête (interface des utilitaires, ex., /usr/include/gtk-2.0/gtk/gtk.h).
6. /export : machine serveur.

7. /export/home/users : répertoire des répertoires utilisateurs (on retrouve aussi souvent “users” dans /home).

1.1.2 Chemin absolu, chemin relatif

Pour décrire le chemin d'accès à un répertoire, on peut le faire :
- soit en donnant le cheminement depuis la racine (chemin absolu du répertoire depuis la racine) ;
- soit en donnant le cheminement depuis le répertoire courant (chemin relatif à la localisation actuelle de l'utilsateur).

Par exemple, si l'utilisateur courant est sur le répertoire “/users/toto”, alors pour atteindre le répertoire “/users/tata”, il peut écrire “cd /users/tata” (chemin absolu) ou “cd ../tata” (chemin relatif).

1.1.3 Répertoires particuliers

désignation	**code Unix**	**exemples**	**caractéristique**
racine	/	cd /	absolu
répertoire courant	.	cp gtk/makefile .	relatif
répertoire parent	..	cd ../	relatif
chez soi	∼	cd ∼ cd	absolu

1.1.4 Fenêtre de commande

Terminal sur lequel on tape les commandes en ligne (comme la fenêtre dos sous windows). Ce n'est rien de plus qu'une interface permettant de communiquer avec le système : par exemple, au lieu de lancer l'éditeur de texte “emacs” par sélection dans la barre de menus, vous pouvez le lancer à partir d'un terminal par la commande : emacs nom_fichier &

Attention, le monde Unix est un monde sensible à la casse ! Ainsi, toto.txt et Toto.txt désignent par exemple deux fichier distincts.

1.2 Commandes de base

1.2.1 Lister le contenu d'un répertoire : ls

La commande “ls” permet d'afficher le contenu d'un répertoire. On peut spécifier le répertoire visé, mais aussi restreindre l'affichage à certains noms de fichier :

ligne de commande	**résultat**
ls	contenu du répertoire courant
ls ../tata	contenu du répertoire /users/tata
ls ../tata/*.txt	fichiers du répertoire /users/tata d'extension “txt”

1.2.2 Savoir où l'on est : pwd

Synopsis : pwd
Indique le chemin absolu (*i.e.*, depuis la racine) du répertoire courant.

1.2.3 Se déplacer dans l'arborescence : cd

Synopsis : cd chemin_repertoire_cible
On peut donner le chemin relatif ou absolu du répertoire à atteindre.

1.2.4 Créer un répertoire : mkdir

Synopsis : mkdir chemin_repertoire_cible

Ex. 1 : mkdir test	Crée un répertoire "test" sur le répertoire courant.
Ex. 2 : mkdir /users/tata/test	Crée un répertoire "test" sur le répertoire "/users/tata".

1.2.5 Copier un fichier : cp

Synopsis : cp chemin_fichier_source chemin_fichier_cible.
Si le nom du fichier cible n'est pas spécifié, il sera identique à celui du fichier source.
Ex. 1 : cp test /users/tata
Copie le répertoire "test" situé dans "/users/toto" sur le répertoire "/users/tata".
Ex. 2 : cp ./* /users/tata
Copie tous les fichiers du répertoire courant sur le répertoire "/users/tata".
Ex. 3 : cp test /users/tata/test_tata
Copie le répertoire "test" (et tout ce qu'il contient) situé dans "/users/toto" sur le répertoire "/users/tata", sous le nom de "test_tata".

1.2.6 Déplacer ou renommer un fichier : mv

Synopsis : mv chemin_fichier_source chemin_fichier_cible.
Si le nom du fichier cible n'est pas spécifié, il sera identique à celui du fichier source.
Ex. 1 : mv test /users/tata
Supprime le répertoire "test" situé dans "/users/toto" pour le recréer dans le répertoire "/users/tata".
Ex. 2 : mv ./* /users/tata
Supprime tous les fichiers du répertoire courant pour les recréer dans le répertoire "/users/tata".
Ex. 3 : mv test /users/tata/test_tata
Supprime le répertoire "test" (et tout ce qu'il contient) situé dans "/users/toto" pour le recréer (ainsi que tout son contenu) dans le répertoire "/users/tata", sous le nom de "test_tata".

1.2.7 Détruire un fichier : rm et rmdir

Synopsis : rm chemin_fichier_cible rmdir chemin_repertoire_cible
Ex. 1 : rmdir test /users/tata
Supprime le répertoire "test" (et tout ce qu'il contient) situé dans "/users/toto".
Ex. 2 : rm ./*
Supprime tous les fichiers du répertoire courant.

Attention, le comportement des commandes dépend du paramétrage établi par l'administrateur système (ou par l'utilisateur dans ses fichiers de configuration). Ce paramétrage consiste notamment à affecter les options des commandes ou à surcharger les commandes par le biais d'alias. Aussi, ne pas hésiter à se référer à l'aide pour retrouver la liste des options des commandes.

1.3 Commandes essentielles

1.3.1 Recherche de fichier sur le nom : find

Synopsis : find chemin_repertoire_source -name nom_cible
Ex. : find . -name *.txt
Recherche à partir du répertoire courant tous les fichiers dont le nom se termine par ".txt" (recherche récursive).

1.3.2 Recherche de texte à l'intérieur des fichiers : grep

Synopsis : grep text_cible chemin_fichier_source
Ex. : grep "<table>" html/*.html
Recherche dans le répertoire /users/toto/html et dans tous les fichiers d'extension "html" de ce répertoire les lignes contenant le texte "<table>".

1.3.3 Demander de l'aide : man

Synopsis : man nomCommande
Ex. : man ls
Affiche l'aide (syntaxe, différentes options, exemples etc.) sur la commande ls.

1.3.4 Afficher un fichier : more et cat

Synopsis : more chemin_fichier_cible cat chemin_fichier_cible
Ex. 1 : more html/cv.html
Affiche pas-à-pas le contenu du fichier "cv.html" situé sur le répertoire "html".
Ex. 2 : cat html/cv.html
Affiche le contenu du fichier "cv.html" situé sur le répertoire "html".
Ex. 3 : more html/*.html
Affiche pas-à-pas le contenu de tous les fichiers d'extension "html" du répertoire "html".

1.3.5 Créer un fichier : touch et cat

Synopsis : cat > chemin_fichier_cible touch chemin_fichier_cible
Ex. 1 : touch > html/loisir.html
Crée un fichier vide du nom de "loisir.html" dans le répertoire "html".
Ex. 2 : cat > html/loisir.html
Crée un fichier "loisir.html" dans le répertoire "html" et propose d'en saisir le contenu ; pour sortir

du mode écriture dans le fichier, il faut taper “ctrl+D” après un retour chariot.

Plus généralement, pour conserver le résultat d’une commande dans un fichier, il suffit d‘écrire “> chemin_fichier_cible” à la suite de la commande. Par exemple, toto possède un répertoire mp3 et souhaite fournir à tata un fichier contenant la liste des mp3 dont il dispose : pour créer un tel fichier, il n’a qu’à taper : ls mp3 > listeMp3Toto.txt

1.3.6 Interrompre le processus premier plan en cours

Pour interrompre le procesus premier plan en cours, il faut taper “ctrl+C”.

1.3.7 Lancer un processus en tâche de fond

Pour lancer un processus en tâche de fond, il faut clore l’instruction de commande par “&” ; par exemple, pour ouvrir l’éditeur emacs en tâche de fond, taper : emacs html/loisir.html &

Remarque : Si vous oubliez le “&”, vous perdez la main sur la fenêtre de commande : celle-ci ne redeviendra accessible que lorsque le processus qui vient dêtre lancé sera terminé.

1.3.8 Compiler un programme C

Le compilateur C généralement utilisé sous Unix est “gcc”.
Ex. 1 : gcc nom_fichier.c
Compile le fichier source nom_fichier.c situé dans le répertoire courant et génère un fichier binaire “a.out” sur le répertoire courant.
Ex. 2 : gcc -o bin/monProg src/nom_fichier.c
Compile le fichier source nom_fichier.c situé dans le répertoire “src” et génère un fichier binaire “monProg” sur le répertoire “bin”.
Ex. 3 : gcc -o bin/monProg src/nom_fichier.c -Wall
Compile le fichier source nom_fichier.c en activant tous les warning (alertes sur le code, hors erreurs de compilation qui sont quant à elles toujours affichées).

1.3.9 Compiler une application gtk

Pour compiler uen application gtk, il faut ajouter des options à la commande de compilation :
gcc -o bin/mon_prog_gtk src/monProg.c -Wall ‘pkg-config --cflags --libs gtk+-2.0‘

1.3.10 Exécuter un fichier

Pour exécuter un fichier exécutable, il faut à chaque lancement de fichier taper :
./chemin_fichier (au lieu de “nom_fichier”)
Ex. : ./bin/mon_prog_gtk
Exécute l’application gtk que l’on vient de compiler.

Remarque

C'est pour des raisons de sécurité que le répertoire courant n'est pas mis dans le PATH : si auparavant le chemin "répertoire courant" était systématiquement ajouté à la variable PATH des systèmes Unix, il en a été récemment banni, du fait de la profusion des virus. En effet, imaginons que l'on reçoive un fichier virusé nommé **ls** sur son répertoire (pratique courante... sous Windows également, avec `notepad.exe` par exemple), alors l'utilsateur, en tapant **ls**, activera le virus !

1.3.11 Complétion automatique

Pourquoi taper le nom entier lorsque cela n'est pas nécessaire... La plupart des shell Unix&Linux disposent d'une fonction de complétion automatique : il vous suffit de taper le début du mot (qu'il s'agisse de l"un de vos fichiers ou d'une commande) puis de taper sur le caractère de tabulation pour que le nom apparaissent entier, dès lors qu'il n'y a pas d'ambiguïté.

1.3.12 Historique des commandes

Normalement, l'historique des commandes est préservé et pour rappeler vos commandes antérieures, il suffit de taper sur la touche "flêche vers le haut".

1.4 En savoir plus

1.4.1 Personnalisation

Les fichiers qui définissent l'environnement utilisateur se situent à la racine du répertoire utilisateur (`$HOME`) et ont un nom commençant par un **point**. Pour se rendre sur son répertoire, taper "cd" ou "cd ∼" ; pour lister ses fichiers d'environnement, taper "ls -a".

1.4.2 Variables d'environnment

Pour connaître les variables d'environnement qui sont définies, taper :
env
Pour afficher la valeur d'une variable d'environnment, taper :
echo $NOM_VARIABLE
Pour affecter la valeur "toto" à la variable d'environnement "TOTO", taper :
TOTO=toto

Remarques :

- les variables d'environnement n'ont pas besoin d'être déclarées pour être définies ;
- "echo $NOM_VARIABLE" renvoie le vide si la variable NOM_VARIABLE n'est pas définie ;
- "echo NOM_VARIABLE" renvoie "NOM_VARIABLE" (*i.e.*, si l'on omet de faire précéder par "$" le nom de la variable, c'est la valeur de la chaîne de caractères "NOM_VARIABLE" qui est renvoyée).

Exemple :

CC=gcc
#crée la variable d'environnement "CC" dont la valeur est "gcc"
CFLAGS=-Wall
#crée la variable d'environnement "CFLAGS" dont la valeur est "-Wall"
SRCDIR=$HOME/src
#crée la variable d'environnement "SRCDIR" dont la valeur est "$HOME/src", soit "/users/toto/src"
BINDIR=$HOME/bin
#crée la variable d'environnement "BINDIR" dont la valeur est "$HOME/bin", soit "/users/toto/bin"
$CC -o $BINDIR/monProg $SRCDIR/nom_fichier.c $CFLAGS
#Interprète la ligne de commande en remplaçant les variables par leur valeur, soit :
gcc -o /users/toto/bin/monProg /users/toto/src/nom_fichier.c -Wall

Variable PATH :

La variable PATH contient une liste de répertoires ; lorsque l'on écrit une commande, c'est dans ces répertoires que le système va la recherche. Par exemple, PATH contient toujours "/usr/bin" car celui-ci contient toutes les commandes Unix. Si vous avez besoin d'utiliser un outil particulier dont le chemin ne figure pas dans votre variable PATH, il faudra ajouter ce chemin à votre variable. Par exemple, pour utiliser un client mySql, il faudra peut-être ajouter le chemin "/usr/bin/local/mysql". Pour ne pas perdre les chemins déjà contenus dans votre variable, il faudra alors taper :
PATH=$PATH:/usr/bin/local/mysql (le caractère ":" est le caractère de séparation).

1.4.3 Gérer les processus : ps et kill

Pour lister les procesus en cours, il faut taper :
ps
Pour tuer un processus, il faut au préalable avoir pris connaissance de son identifiant (notons-le identifiant_processus) à l'aide de la commande "ps" (colonne "PID"), puis taper :
kill -9 identifiant_processus

1.4.4 Redirections

La sortie standard est généralement l'écran. Pour rediriger le résultat de n'importe quelle commande vers un fichier, il faut taper : "command > nom_fichier". Par exemple, si l'on souhaite conserver le manuel d'aide sur la commande "ls", on peut écrire :
man ls > manLs.txt.

Remarque : lorsque l'on utilise la redirection "> nom_fichier", si le fichier nom_fichier existe, il est effacé ; pour rediriger le résultat d'une commande *en fin de fichier* (le contenu initial du fichier n'est pas effacé), il faut écrire : nomCommande >> nom_fichier.

L'entrée standard est généralement le clavier. Pour rediriger l'entrée de n'importe quelle commande vers un fichier : "command < nom_fichier". Par exemple, "sort < nom_fichier" permet de trier le fichier désigné.

On peut enfin utiliser la sortie d'une commande comme entrée d'une seconde commande : pipe "|". Par exemple, pour trier l'aide de la commande "ls", on peut écrire : man ls |sort ; pour afficher les fichiers dont le nom contient la chaîne de caractères "radiohead", on peut écrire "ls mp3 |grep "radiohead"" de façon équivalente à " ls mp3/*radiohead*".

1.4.5 Permissions

Le système Unix distingue trois types d'utilisateurs : l'utilisateur propriétaire du compte, le groupe (par exemple : groupe licence1) et le reste de monde. L'utilisateur peut choisir de donner ou non accès à ses fichiers à ces trois types d'utilisateur. De plus, on distingue trois types de droits : droit de lecture, d'écriture et d'éxécution (pour les binaires et les répertoires).

Pour connaître les droits d'accès à vos fichiers :
ls -l
Ex. : ls -l /users
drwxr-xr– toto
drwxr-xr– tata
La première lettre indique le type de fichier (ici, "d" pour répertoire).
Les 3 suivantes décrivent les droits d'accès pour l'utilisateur propriétaire du répertoire (ici, "r" pour lecture, "w" pour écriture et "x" pour exécution).
Les 3 suivantes décrivent les droits d'accès pour tout utilisateur du même groupe (ici, "r" pour lecture et "x" pour exécution).
Les 3 dernières décrivent les droits d'accès pour tout l'utilisateur lambda (ici, "r" pour lecture).

Pour modifier les droits d'accès à vos fichiers :
chmod [ugo] [+-] [rwx]
Ex. 1 : chmod g+w /users/toto
Donne accès en écriture à tout uilisateur du groupe.
Ex. 2 : chmod o-r /users/toto
Retire l'accès en lecture à tout uilisateur extérieur au groupe et à l'utilisateur.

Remarque : pour être juste, il faut mentionner un quatrième type d'utilisateur : l'administrateur système, appelé "root" ; celui-ci a tous les droits !

2 TD

2.1 Fichiers, répertoires

Un terminal est ouvert sur lequel le répertoire courant est la racine de votre compte.

1. Créez un répertoire td (à la racine de votre compte) puis vous y rendre.
2. Créez un fichier fic-1.txt contenant le texte “Coucou ! ! !” sur ce répertoire.
3. Créez deux fichiers vides fic-2.txt et fic-3.txt sur ce répertoire.
4. Créez sous le répertoire td les sous-répertoires td-1, td-2 et td-3.
5. Déplacez le fichier fic-1.txt dans le répertoire td-1, fic-2.txt dans le td-2 et fic-3.txt dans td-3.
6. Copiez le contenu de fic-1.txt dans fic-3.txt.
7. Changez le nom du répertoire td-3 en td-2-1 et le nom du fichier fic-3.txt en fic-2-1.txt.
8. Déplacez le répertoire td-2-1 dans le répertoire td-2.
9. Copiez tous les fichiers de td-1 et td-2 dans td.

2.2 Prénoms

Un terminal est ouvert sur lequel le répertoire courant est la racine de votre compte.

1. Créez dans td un fichier “prenom” contentant les lignes suivantes :
   alain
   jerome
   xavier
   claire
   aurelie
2. Ajouter à “prenom” les lignes suivantes :
   noemie
   arthur
   alain
3. Affichez les prénoms contenus dans le fichier “prenom” par ordre alphabétique.
4. Créez un fichier ‘prenom-trie” qui contient les prénoms contenus dans le fichier “prenom”, mais triés par ordre alphabétique.

2.3 Recherche de/dans les fichiers

Un terminal est ouvert sur lequel le répertoire courant est la racine de votre compte.

1. Affichez tous les fichiers de votre répertoire racine dont le nom contient la lettre “p”.
2. Êtes-vous certain d’avoir affiché *tous* les fichiers contenant le lettre “p” ?
3. Affichez tous les fichiers de votre compte dont le nom commence par la lettre “f”.
4. Affichez les lignes des fichiers du répertoire “td” qui contiennent la chaîne de caractères “la”.

3 TP

3.1 Fichiers, répertoires

Un terminal est ouvert sur lequel le répertoire courant est la racine de votre compte.

1. Quel est le chemin absolu de votre répertoire courant ?
2. Rendez-vous sur le répertoire de votre voisin en décrivant le chemin absolu.
3. Retournez chez vous, en décrivant le chemin relatif.
4. Créez chez vous les répertoires “toto” et “tata”.
5. Créez chez votre voisin le répertoire “titi” ; que se passe-t-il ?
6. Détruisez votre répertoire “toto”.
7. Détruisez le répertoire “toto” de votre voisin ; que se passe-t-il ?

3.2 Organisez votre espace de travail

1. Créez un répertoire gtk à la racine de votre compte.
2. Dans votre répertoire gtk, créez les répertoires : bin, src et inc.

3.3 Ensuite, faites un peu d’espionnage industriel

Le chemin absolu de mon répertoire gtk est le suivant : /export/home/users/Enseignants/toulouse/gtk le contenu :

1. Copier tous les fichiers de mon répertoire gtk sur votre répertoire gtk en utilisant cette information.
2. Et vous, quel est le chemin absolu de votre répertoire gtk ?
3. Copier tous les fichiers de mon répertoire gtk/src sur votre répertoire gtk/src en utilisant les chemins relatifs.
4. Idem inc et bin.

3.3.1 Vérifions que tout s'est bien passé

Après vous être placé à la racine de votre compte :

1. Donnez le contenu de la racine de votre compte ; de votre répertoire gtk.
2. Placez-vous à présent dans votre répertoire gtk et :
   (a) Listez le contenu du répertoire bin.
   (b) De quel type les éléments qu'il contient sont-ils ?
   (c) Qui a quel type d'accès à ces éléments ?
3. Idem pour gtk ; idem pour src.

3.4 Quelques manip pour la forme

Placez-vous à la racine de votre répertoire.

1. Que vaut votre variable PATH ?
2. Tapez PATH= (enter) ; tapez ls (enter) ; que se passe-t-il ?
3. Que vaut votre variable PATH ?
4. Fermez votre terminal et ouvrez-en un nouveau.
5. Que vaut votre variable PATH (après j'arrête avec PATH, c'est promis) ?
6. Créez un fichier vide "toto.txt" puis tapez "emacs toto.txt".
7. Comment reprendre la main sur le terminal ?

3.5 Débuter avec gtk

3.5.1 Lancer une application gtk

Dans votre répertoire bin, vous devez avoir récupéré l'exécutable mon_prog_gtk : lancez-le !

3.5.2 Personnaliser l'application

Ouvrez le fichier source mon_prog_gtk.c situé dans le répertoire src avec l'éditeur de votre choix. On va vous demander de modifier ce fichier, de le compiler et en cas de réussite, d'exécuter le binaire résultant.

Pour compiler le fichier, il faut se placer sur la fenêtre de commandes et taper make. Pour information : les directives de compilation sont écrites dans le fichier "makefile" et lorsque l'on tape la commande "make", le système effectue la compilation en suivant les directives qui y sont décrites. ***Attention*** : pour que la commande make trouve le fichier makefile (en fait on aurait pu appeler autrement ce fichier, mais il aurait alors fallu spécifier le nom du fichier cible dans les options de la

ligne de commandes), il faut lancer la commande dans le même répertoire, i.e., dans le répertoire d'appartenance du fichier makefile visé. Pour information encore, le compilateur utilisé est gcc.

Ce makefile compile toujours le fichier mon_prog_gtk.c situé dans le répertoire src et écrit toujours le binaire résultant sous le nom de mon_prog_gtk dans le répertoire bin.

Placez-vous donc dans votre répertoire gtk (au cas où vous en seriez parti) et vérifiez qu'un fichier nommé makefile s'y situe bien. Pour chacune des questions qui suivent, il faut : 1) modifier le fichier source (et le sauvegarder bien sûr) ; 2) lancer la compilation ; 3) exécuter le binaire et observer le résultat !

1. Dans la barre de titre, écrivez plutôt "gtk - Premier TP".
2. Traduisez en français le texte affiché à l'intérieur de la fenêtre !
3. Faites en sorte que la fenêtre soit de largeur 1000 et de hauteur 500.

Récapitulatif des commandes Unix de base

commande	**vocation**	**exemples**
cd	atteindre un répertoire	cd igwHtml/doc/ cd / cd ../igwGtk
mkdir	créer un répertoire	mkdir old mkdir ../igwGtk/old
cp	copier un fichier	cp cv.html cv_old.html cp cv.html old/ cp cv.html old/cv.html
mv	déplacer/renommer un fichier/un répertoire	mv cv.html old/cv.html mv cv.html Cv.html mv igwHtml IgwHtml mv old ../htmlOld
ls	lister le contenu d'un répertoire	ls old ls *.html ls ../igwGtk/src/*.c
ls -l	... en détail	ls -l old ls -l *.html ls -l ../igwGtk/src/*.c
rm	supprimer un fichier	rm old/cv.html
rmdir	supprimer un répertoire	rmdir old
more	afficher pas-à-pas	more cv.html ls * | more
cat	afficher un fichier	cat cv.html
cat >	créer un fichier	cat > cvBis.html (saut ligne puis ctrl+D pour terminer)
cat >>	écrire en fin de fichier	cat >> cvBis.html (saut ligne puis ctrl+D pour terminer)
touch	créer un fichier vide	touch cvBis.html