
Chapitre 1 : Initiation aux commandes UNIX

1.1. UNE INTRODUCTION :

Pour utiliser un ordinateur, l'utilisateur dialogue avec un logiciel qui gère l'ensemble de ses ressources (clavier, écran, souris, mémoire centrale, disques, imprimante, ...). Ce logiciel porte le nom de SYSTÈME D'EXPLOITATION. C'est maintenant devenu un système multi-utilisateurs et multi-tâches.

Le système d'exploitation des ordinateurs PC peut être WINDOWS ou LINUX. Celui des ordinateurs APPLE s'appelle SYSTEM 10 avec un noyau UNIX. Les stations de travail de type HP ou IBM ou SGI ou SUN jusqu'aux ordinateurs vectoriels/parallèles peuvent employer le système UNIX.

Ce système UNIX a été conçu à l'origine par Dennis RITCHIE et Ken THOMPSON des laboratoires Bell au début des années 1970. Depuis, il a donné naissance à une famille de systèmes (AIX, BERKELEY, LINUX, SYSTEM V, ...) qui "réagissent" comme UNIX.

Chaque système UNIX permet le dialogue entre l'ordinateur et le ou les usagers selon un **langage de commandes** et la gestion des ressources informatiques grâce à

- un **gestionnaire de processus** pour exécuter une séquence de commandes ;
- un **système hiérarchisé de fichiers** ;
- des **fonctions spécialisées**.

Les commandes permettent

- toutes les manipulations de fichiers (ls, cp, mv, rm, ...) ;
- l'édition et le traitement de textes ; (ed, vi, emacs, nedit, xedit, ...) ;
- la compilation des différents langages informatiques (C, C++, Fortran, Pascal, Ada, Java, ...) ;
- l'édition de liens des modules issus des compilations et leur archivage ; (ld, ar, ...)
- l'exécution des modules exécutables (NomFichier, ...) ;
- l'échange de données sur les réseaux de télécommunications tel INTERNET (mail, news, telnet, ssh, scp, ftp, web, ...).

L'interpréteur de commandes, encore dit le SHELL, peut exécuter des commandes regroupées dans un fichier. L'ensemble de ces nouvelles commandes définissent alors une nouvelle commande (de nom le nom du fichier) et ainsi de suite , Chaque utilisateur construit progressivement son environnement personnel pour répondre à l'ensemble de ses besoins.

1.2. LES GRANDS PRINCIPES D'UNIX :

1.2.1. LE SYSTÈME de GESTION des FICHIERS :

Un fichier est formé d'informations (texte, instructions, commandes, images, sons, films, ...) regroupées sous un NOM et généralement à conserver entre 2 exécutions.

À ce fichier sont associés des DROITS D'ACCÈS qui permettent de l'utiliser en lecture et/ou en écriture et/ou en exécution.

Pour éviter que les fichiers apparaissent sur un seul niveau, la notion de RÉPERTOIRE ou CATALOGUE (directory en anglais) permet la hiérarchisation des fichiers en une ARBORESCENCE. Répertoires (noeuds communs à plusieurs branches de l'arbre) et fichiers (feuilles de l'arbre) constituent un ARBRE dont la racine (le répertoire initial) a pour nom /

Le caractère / en début d'un nom de fichier ou répertoire désigne le disque par défaut de l'ordinateur. Souvent, il s'agit de l'unique disque "dur" de la station de travail ou de l'ordinateur.

De même que pour les fichiers, les DROITS D'ACCÈS d'un répertoire permettent ou non la lecture de ses fichiers ou sous-répertoires, l'ajout ou non de nouveaux fichiers ou sous-répertoires aux utilisateurs habilités.

Les unités périphériques (clavier, souris, modems, ...) envoient en mémoire soit centrale notée MC, soit, secondaire (disque ou disquette) notée MS, des informations (caractères tapés au clavier, mouvement de la souris, ...) regroupées sous la forme d'un fichier dit d'ENTRÉE.

En sens inverse, des informations contenues en mémoire et regroupées sous la forme d'un fichier dit de SORTIE peuvent être transmises aux unités périphériques (affichage sur l'écran ou l'imprimante, transfert de la MC sur MS, envoi sur une ligne téléphonique, ...). De façon générale, les fichiers de SORTIES ne sont pas conservés localement après leur transfert.

1.2.2. Les PROCESSUS ou TACHES :

Tous les travaux de l'utilisateur sont réalisés par des processus. Un PROCESSUS est l'exécution d'un programme particulier appelé "SHELL" qui est en fait un interpréteur de commandes. Une commande frappée est interprétée par le processus qui réserve tout ce qui est nécessaire à son exécution (mémoire centrale, fichiers, ...) et l'exécute.

Au démarrage du système, un processus est créé. Un processus, dit alors père, peut demander la création d'un nouveau processus dit processus fils. Un processus fils est créé par copie du processus père. Le processus père a la possibilité d'attendre la fin de l'exécution du processus fils. Un processus peut aussi se remplacer lui-même par un autre code exécutable.

Un processus peut être exécuté en interactif ou en tâche de fond (& doit être ajouté à la fin de la commande).

Les commandes peuvent être tapées au clavier ou lues dans un fichier. Le nom d'un fichier contenant des commandes exécutables apparaît ainsi comme une nouvelle commande. Il est ainsi possible de créer un environnement adapté aux besoins. Il est aussi possible d'enchaîner des processus, les résultats du précédant servant de données au suivant grâce à la notion de TUBE ou CANAL (pipe en anglais).

1.3. COMMENT UTILISER UNIX EN INTERACTIF :

Un UTILISATEUR d'Unix est reconnu par son User-Id (par exemple son nom martin). De plus, il appartient à un ou plusieurs groupes repérés par un Group-ID (par exemple p6-edpan).

Ces 2 identificateurs ont dû être enregistrés par l'administrateur du système avant toute tentative de connexion.

La commande **id** ou **id UserId** retourne les 2 noms déclarés.

La commande **newgrp AutreGroupe** permet le changement de groupe d'un utilisateur.

1.3.1. LA CONNEXION :

Après avoir allumé le PC, le système vous demande votre nom et mot de passe dans des cases à gauche d'une ellipse contenant XFree86.

Dans la case supérieure, taper votre nom d'utilisateur, puis sur la ligne suivante, votre mot de passe (8 caractères sont souhaités).

L'affichage de ces caractères est annihilé pour des raisons évidentes de sécurité.

Valider par la frappe de la touche <**Entrée**>.

Votre nom doit être connu du système, c'est à dire avoir été enregistré par la procédure d'inscription.

Si le nom ou mot de passe est inconnu, vous n'aurez plus qu'à recommencer...

Sinon, le fond de l'écran devient gris-bleu et 2 rectangles apparaissent. Le gestionnaire de fenêtres est fvwm et le rectangle de droite montre les 9 fenêtres possibles. Cliquer sur une pour avoir son contenu sur l'écran.

Cliquer le bouton gauche de la souris pour faire apparaître le menu principal.

Pour ouvrir une fenêtre xterm, choisir par déplacement de la souris l'option

Root menu + Shells + XTerm

Une fenêtre s'ouvre alors et derrière le curseur, il est alors possible d'y taper toute commande UNIX, ...

Pour terminer la session, il faut cliquer la dernière option du menu principal

Quit FVWM et confirmer par Yes Really Quit.

1.3.2. CHANGER le MOT DE PASSE :

Il faut taper la commande

passwd

Il apparaîtra ensuite

(current) UNIX password : (donnez l'ancien mot de passe)

New UNIX password : (donnez le nouveau mot de passe)

Retype new password : (redonnez le même nouveau mot de passe !)

ATTENTION : N'oubliez pas ce nouveau mot de passe sous peine d'avoir à contacter le secrétariat du Master pour forcer un nouveau mot de passe ce qui vous fera perdre du temps.

Attention, la législation réprime tout accès ou tentative d'accès de personnes non autorisées à un ordinateur. Il est ABSOLUMENT NÉCESSAIRE de PROTÉGER votre MOT DE PASSE sous peine de risquer des poursuites dues à l'utilisation frauduleuse

de votre nom par une tierce personne. Votre mot de passe équivaut à votre signature. Il est donc recommandé de choisir un nom difficile à deviner et surtout de ne jamais stocker ce nom sur papier. A vous de le mémoriser !

1.3.3. Les FENÊTRES sur l'ÉCRAN et les TOUCHES du CLAVIER :

Dans tout ce qui suit, un terme entre <> désigne une touche du clavier.

Le caractère haut d'une touche du clavier s'obtient en gardant enfoncée la touche <Shift> et en frappant la touche.

Le caractère bas à droite d'une touche (par exemple : \ ...) s'obtient en gardant enfoncée la touche <AltGr> et en frappant la touche.

<Backspace> efface le caractère qui précède le curseur qui recule d'un caractère

<Char Del> efface le caractère du curseur sans bouger de place

Attention :

- Toute ligne est à terminer (valider) par la frappe de la touche <Entrée> ;
- Pour taper une commande sur plusieurs lignes, il faut terminer chaque ligne, exceptée la dernière, par un caractère \, SANS AUCUN AUTRE CARACTÈRE DERRIÈRE, et surtout pas un caractère "blanc". En effet, le caractère \ neutralise le caractère de fin de ligne <Entrée> placé derrière. S'il existe un caractère "blanc" invisible, \ le supprime et non pas celui de fin de ligne et cela entraîne une erreur car les arguments de la commande sur les lignes suivantes ne sont pas transmis.

. Une FENÊTRE ouverte sur l'écran comprend plusieurs parties :

+ Une barre d'en tête avec 3 boutons :

- le bouton - à gauche, fait apparaître un menu pour gérer la fenêtre, notamment, la déplacer ou la détruire.
- le bouton . à droite réduit la fenêtre à une icône. (Il suffit pour régénérer la fenêtre à partir de l'icône, de cliquer 2 fois rapidement l'icône).
- le bouton carré à l'extrême droite augmente la taille de la fenêtre jusqu'à lui faire occuper tout l'écran.

+ Une fenêtre avec 2 parties :

- la partie basse ou ligne courante permet l'entrée des COMMANDES UNIX
- la partie haute contient les RÉSULTATS des commandes déjà exécutées

La taille et la position de la fenêtre peuvent être modifiées comme suit :

- positionner avec la souris le curseur dans un des 4 coins du cadre de la fenêtre,
- appuyer sur le bouton gauche de la souris et déplacer la souris sans relâcher le bouton,
- relâcher le bouton à l'endroit choisi.

Pour faire apparaître une fenêtre cachée, il faut déplacer la fenêtre qui la masque.

<Ctrl> <D> sur la ligne courante d'une fenêtre, détruit la fenêtre.

Pour ouvrir une autre fenêtre, une méthode consiste à taper l'instruction

xterm -sb -sl 10000 -cr red -title NomTitre \

-bg black -fg white -fn 9x15 -fb 9x15bold -geometry 80x50+0+0 &

Alors une fenêtre, avec ascenseur, pouvant contenir au plus 10000 lignes, avec un curseur rouge, un titre NomTitre, un fond noir et le texte écrit en blanc selon la fonte normale ou grasse formée de 9 caractères de large sur 15 de haut avec 50 lignes de 80 caractères

s'ouvre en haut et à gauche sur l'écran. D'autres fontes sont très souvent disponibles (7x14, 8x13, 10x20, ...).

1.3.4. Les **NOMS** de **FICHIERS** et **RÉPERTOIRES** :

Le nom du disque serveur de référence est noté / et sert de racine de l'arbre.

Tout fichier (file) ou répertoire (directory) a un NOM dans l'arbre qui l'identifie.

Un nom de fichier peut comprendre un SUFFIXE séparé du nom par un .

exemples : sp.f projet1/prog.cpp /share/thot/users/2005/p6-edpan/martin/tri.f

Le suffixe suit des **conventions** : .f ou .F pour fortran, .cpp pour C++, .pas pour pascal, .tex pour T_EX, ...

/usr/local/mefisto est un répertoire c'est à dire en fait la liste des noms de ses sous-répertoires et de ses fichiers.

/bin contient la plupart des commandes UNIX ;

/dev contient les fichiers d'emploi des périphériques (lecteurs de bandes magnétiques, "drivers", ...);

/tmp contient les fichiers temporaires nécessités par le système ;

/etc contient des utilitaires d'administration du système ;

/usr contient des répertoires d'utilitaires pour les utilisateurs

/usr/man contient les 8 chapitres de la documentation en ligne (cf man) ;

/usr/lib contient les librairies X, Motif, ... ;

/home ou /users contiennent souvent les répertoires des Utilisateurs.

À l'initialisation de la session interactive, une position dans l'arbre des répertoires est fixée. Il s'agit du RÉPERTOIRE PERSONNEL (HOME directory) de l'usager.

En Master, si le nom de l'usager est martin du groupe p6-edpan alors son répertoire personnel a pour nom /share/thot/users/2005/p6-edpan/martin

Attention : ici, ce répertoire réside en fait sur l'ordinateur central d'accès isis et non pas le PC ou terminal X utilisé.

Ce répertoire est aussi, à cet instant, le RÉPERTOIRE COURANT, encore dit répertoire de travail (WORKING Directory).

De là, pour atteindre un fichier ou répertoire (navigation dans l'arbre), il faut taper

- soit, le NOM ou chemin ABSOLU qui débute par la racine ;

par exemple : /share/thot/users/dea/martin/.bashrc

- soit, le NOM ou chemin RELATIF au répertoire courant ;

par exemple : .bashrc

désigne le même fichier que /share/thot/users/2005/p6-edpan/martin/.bashrc si le répertoire courant est le HOME directory.

. désigne le nom du répertoire courant par exemple :

/share/thot/users/2005/p6-edpan/martin

.. désigne le nom du répertoire père du répertoire courant par exemple : le répertoire père de /share/thot/users/2005/p6-edpan/martin est /share/thot/users/p6-edpan

1.3.5. LES PRINCIPALES COMMANDES D'UNIX :

Une COMMANDE est constituée d'un NOM de COMMANDE suivi d'aucun ou plusieurs mots séparés par au moins un caractère blanc.

NomCommande {-option} {paramètre} {redirection} {&}

Les caractères { et } ne sont là que pour indiquer que le texte inclu peut être présent, répété ou absent.

(Dans les lignes qui suivent **fic** désigne un nom (absolu ou relatif) quelconque de fichier et **rep** un nom quelconque de répertoire)

ls rep liste les fichiers et répertoires du répertoire rep

ls -l rep liste les fichiers et répertoires du répertoire rep avec ses droits d'accès et d'autres informations (taille, date de dernière utilisation, ...)

ls -a rep liste les fichiers et notamment ceux débutant par le caractère . (à ne pas confondre avec l'abréviation pour le répertoire courant) et les répertoires du répertoire rep avec d'autres informations. Les fichiers dont le nom débute par un point . désignent des fichiers d'initialisation de commandes comme .bashrc pour /bin/bash, .ssh pour la commande ssh, ...

ls -R rep liste les fichiers et sous-répertoires du répertoire rep mais aussi des sous-sous-...-répertoires du répertoire rep

ls -rt rep liste les fichiers et sous-répertoires du répertoire rep dans l'ordre inverse de leur date de dernière modification.

Le caractère > redirige les **sorties**.

ls -l >fic

envoie les noms des fichiers et répertoires du répertoire courant, 1 par ligne, dans le fichier de nom fic et non pas sur l'écran ! Les sorties sont dites redirigées dans le fichier fic. Si le fichier fic existait son ancien contenu est perdu.

ls -l >>fic effectue la même opération mais en ajoutant les noms à la suite du contenu actuel du fichier fic s'il existe.

cat fic1 fic2 fic3 >fic concatène les fichiers fic1, fic2, fic3 dans le fichier fic.

En l'absence de >fic le fichier concaténé apparaît sur l'écran.

cat >fic permet l'entrée au clavier du texte d'un fichier, à terminer par la frappe des touches <Ctrl> <D>.

Mais attention, il ne faut pas se tromper car à part la touche <Backspace>, rien n'est prévu pour faire des modifications (Pour cela, employer un éditeur de textes comme ed, vi, nedit, xemacs, ...).

De même, le caractère < redirige les **entrées**. Par exemple

mail AdresseElectronique <lettre envoie à l'adresse électronique le contenu du fichier lettre (de dernière ligne contenant en première colonne le caractère . et rien d'autre) du répertoire courant.

Si le choix de l'une des options d'une commande vous pose des problèmes, il est possible de lire la documentation en ligne en tapant

man NomCommande {numéro 1 à 8 du chapitre}

Par exemple : man ls 1

Le texte de la documentation apparaît et il suffit de parcourir le texte en s'aidant des touches marquées d'une flèche (4 sens) pour obtenir les renseignements recherchés.

Il existe 8 chapitres dans cette documentation en ligne :

- 1 : Les commandes accessibles à l'utilisateur
- 2 : L'interface entre UNIX et le langage C
- 3 : La bibliothèque C et macros et fonctions mathématiques
- 4 : Les caractéristiques des fichiers système associés aux périphériques
- 5 : Les formats des fichiers système
- 6 : Les jeux
- 7 : Les bibliothèques de macros pour la manipulation des documents
- 8 : Les commandes de maintenance du système

date affiche l'heure et la date du jour.

cal 10 1999 affiche le calendrier du mois d'octobre 1999.

cd rep rep devient le répertoire courant.

cd le répertoire courant devient le répertoire personnel (équivalent à **cd \$HOME**).

mkdir rep (Make Directory) crée dans le répertoire courant le nouveau répertoire rep

pwd (Print Working Directory) affiche le nom du répertoire courant

rmdir rep (Remove Directory) détruit le répertoire rep sous la condition qu'il soit vide de tout fichier et sous-répertoire

cp fic1 fic2 ... ficn rep (Copy) fait une copie des fichiers fic1 fic2 ... ficn dans le répertoire rep sans détruire les fichiers fic1 fic2 ... ficn initiaux

mv rep1 rep2 (Move) copie tous les fichiers et répertoires du répertoire rep1 dans le répertoire rep2, puis détruit le répertoire rep1

mv fic1 fic2 ... ficn rep (Move) déplace les fichiers fic1 fic2 ... ficn dans le répertoire rep et détruit les fichiers fic1 fic2 ... ficn initiaux (GARE aux ERREURS si rep est oublié ! ficn contient fic n-1 et fic1 fic2 ... fic n-1 sont détruits !)

rm fic (Remove) détruit le fichier fic

rm -R rep (Remove) détruit toute la branche issue du répertoire rep et le répertoire rep lui-même. A utiliser avec discernement !

rm -Rf rep (Remove) détruit toute la branche issue du répertoire rep et le répertoire rep lui-même sans aucune demande de confirmation. A utiliser encore plus avec discernement !

lpr -Plw6c fic imprime le fichier fic sur l'imprimante de nom lw6c.

more fic affiche le contenu du fichier fic.

Pour passer à la page suivante taper un espace. Attention, pas de remontée possible.

Pour terminer, taper :q

Pour spécifier plusieurs noms de fichiers ou répertoires, il est possible d'employer la "STAR-CONVENTION" c'est à dire les caractères JOKER encore dits métacaractères (neutralisables par \) ou spéciaux :

***** remplace n'importe quel nom jusqu'au prochain blanc ou . ou / possible

exemple :

***.f** désigne tous les fichiers du répertoire courant suffixés par f

. désigne tous les fichiers ayant un suffixe

? remplace n'importe quel caractère

! indique NON de ce qui suit

exemple :

sp ?.f désigne tous les fichiers dont le nom commence par les 2 caractères sp, suivi d'un caractère et de suffixe .f .

[**a-z**] remplace n'importe quel caractère a ou b ou ... ou z

[**!a-w**] remplace n'importe quel caractère différent de a ou b ou ... ou w

[**0-9**] remplace n'importe quel caractère 0 ou 1 ou ... ou 9

Avant utilisation, il est prudent d'afficher le résultat de l'analyse des ces caractères JOKER à l'aide de la commande echo :

echo NomAvecCaractèresJOKER affiche tous les noms ainsi désignés

Une commande peut souvent être stoppée par <Ctrl> <C>

Les **droits d'accès** d'un fichier ou répertoire peuvent être listés par la commande **ls -l** et redéfinis par leur propriétaire à l'aide de la commande

chmod -R rwxrwxrwx rep ou fic

chmod mode rep ou fic

où r vaut 0 ou 4, w vaut 0 ou 2, x vaut 0 ou 1, sont à sommer pour donner l'autorisation en lecture (r=4), en écriture(w=2), en exécution (x=1), pour les 3 cas :

- le premier groupe rwx est pour le propriétaire ;
- le second groupe rwx est pour le groupe ;
- le troisième groupe rwx est pour tous les autres utilisateurs ;
- -R signifie récursivement c'est à dire dans les sous-répertoires aussi ;
- mode vaut

. en premier caractère **u** (utilisateur) ou **g** (groupe) ou **o** (autres) ou **a** (à la fois ugo)

. en second caractère + ou - ou = pour ajouter, supprimer ou définir une valeur ;

. en troisième caractère **r** pour lecture, **w** pour écriture et **x** pour exécution.

Exemples :

chmod 750 fic1

donne les droits de lecture-écriture-exécution au propriétaire, lecture-exécution (pas en écriture !) aux membres du même groupe et aucun accès aux autres utilisateurs du fichier fic1.

chmod o-w MonFichier

retire la possibilité d'écrire aux autres (excepté le propriétaire et les membres du même groupe) et donc de modifier le fichier MonFichier.

chmod a+rx MaProc

permet à tout le monde la lecture et l'exécution du fichier MaProc.

La recherche des occurrences d'un MOT dans le contenu des fichiers d'une liste de fichiers (générée à partir de caractères JOKER) s'obtient par la frappe de

grep MOT NomsDesFichiers affiche les lignes contenant au moins une fois MOT dans le **contenu** des fichiers d'une liste de fichiers

grep -n MOT NomsDesFichiers affiche les lignes et leur numéro contenant au moins une fois MOT dans le contenu des fichiers NomsDesFichiers

grep -i MOT NomsDesFichiers affiche les lignes contenant au moins une fois MOT sans se préoccuper de l'aspect minuscule ou majuscule dans le contenu des fichiers NomsDesFichiers.

find rep -name "MOT" -print affiche les noms des fichiers du répertoire rep dont chaque **nom** contient MOT .

Ici aussi, MOT peut être décrit avec des caractères JOKER.

wc fic donne le nombre de lignes, mots et caractères du fichier.

grep MOT *.f | **wc** équivaut à

grep MOT *.f >fic ; **wc** fic ; **rm** fic

Le caractère séparateur ; permet la frappe de plusieurs commandes sur une même ligne.

Le TUBE ou CANAL | permet d'employer les sorties de la première commande comme entrées de la seconde sans avoir à utiliser un fichier auxiliaire.

diff NomFichier1 NomFichier2 affiche toutes les différences entre les contenus des fichiers NomFichier1 et NomFichier2

La commande TAR :

La commande tar permet le transfert du contenu de fichiers ou de répertoires dans un fichier séquentiel (traditionnellement suffixé par .tar) lequel peut être ensuite copié sur une bande magnétique ou CD ou tout autre support stable.

tar -cvf NomFichier.tar Répertoires Fichiers

tar -cvf /dev/NomUnitéBande Répertoires Fichiers

transfère tous les Fichiers et Répertoires (et leurs fichiers et sous-répertoires) dans le fichier NomFichier.tar en ne mentionnant surtout pas dans le nom du chemin des Répertoires et Fichiers les caractères conduisant à Répertoires ou Fichiers.

ATTENTION :

- Le nom du ou des répertoires doit être un chemin RELATIF et NON PAS un chemin absolu, pour permettre une restauration dans n'importe quel répertoire après transfert.
- L'un des 2 termes, Répertoires ou (exclusif) Fichiers, peut ne pas exister.
- Le mot NomUnitéBande est dépendant du périphérique lecteur de bande et dépendant de la station de travail.

En sens inverse,

tar -xvf NomFichier.tar .

transfère dans le répertoire courant le contenu du fichier NomFichier.tar en restaurant tous les réertoires et fichiers.

tar -xvf /dev/NomUnitéBande .

transfère dans le répertoire courant le contenu de la bande.

tar -tvf /dev/NomUnitéBande

liste seulement le nom de tous les fichiers et répertoires stockés sur la bande.

Les "m"commandes du package MTOOLS (non disponibles en salle de TP) :

Elles permettent d' employer sous Unix les disquettes dans le format FAT32. Les commandes sont celles du DOS précédées du caractère "m".

mformat a: formate la disquette.

mcoppy a:fic.suf . copie le fichier fic.suf de la disquette vers le répertoire courant.

mcoppy fic.suf a: copie le fichier fic.suf du répertoire courant sur la disquette.

mdir a : affiche le contenu de la disquette.

Les commandes GZIP et GUNZIP :

Ces commandes compressent et décompressent le contenu du fichier donné. Le fichier compressé a pour nom son ancien nom suffixé par **.gz**

Par exemple :

gzip rep.tar

fournit le fichier compressé rep.tar.gz

Inversement,

gunzip rep.tar.gz

redonne le fichier rep.tar sous sa forme non compressée.

Au total, pour transporter un répertoire (et tous ses sous-répertoires), la commande tar est d'abord utilisée et immédiatement suivie de gzip.

Par exemple :

tar -cvf rep.tar rep

gzip rep.tar

tar -cvf /dev/NomUnitéBande rep.tar.gz

Après transport et positionnement dans le bon répertoire

tar -xvf /dev/NomUnitéBande rep.tar.gz

gunzip rep.tar.gz

tar -xvf rep.tar

Le taux de compression est parfois étonnant sur les fichiers textes, mais, moins important sur les fichiers binaires.

Il est souvent possible de coupler tar et gunzip ou l'opération inverse en ajoutant simplement l'option z à celles de la commande tar.

Par exemple sous LINUX :

tar -czvf rep.tar.gz rep1 rep2

regroupe tous les fichiers des répertoires rep1 et rep2 et leurs sous-répertoires, puis, compresse le tout dans le fichier rep.tar.gz

tar -xzvf rep.tar.gz

décompresse et copie dans les répertoires les différents fichiers.

Très souvent le suffixe .tar.gz est remplacé par le suffixe .tgz

Les LIENS SYMBOLIQUES :

La commande

ln -s AncienNom NomLié

permet de donner facilement à un fichier ou un répertoire un second nom ou alias.

Exemple :

ln -s /share/thot/users/2005/p6-edpan/martin/MonLogiciel /usr/local/MonLogiciel

permet l'écriture de procédures avec la variable d'environnement

MonLogiciel=/usr/local/MonLogiciel

et alors, de placer ce logiciel n'importe où dans l'arbre des répertoires. La maintenance de différentes versions d'un logiciel s'en trouve facilitée.

ls -l NomLié

affiche l' AncienNom véritable.

rm NomLié

ne détruit pas le fichier mais seulement le lien, c'est-à-dire que le NomLié n'existe plus.

1.4. UTILISATION DE L'ÉDITEUR vi (ou mieux vim) :

Cet éditeur est toujours présent sous Unix et très utile lorsque X n'est pas actif, ce qui est le cas lors de certaines installations de LINUX ou sans serveur X ou de recherche de bogues. De plus, certaines de ses commandes servent lors de la manipulation de diverses documentations.

vi NomFichier charge le fichier et l'affiche sur l'écran

2 modes : le mode commande (pour déplacer le curseur ou modifier le texte) et le mode texte (pour sa saisie)

1.4.1. Sorties et sauvegardes

:wq ou **:x** sauvegarde du fichier sur lui-même et sortie de vi

:q sortie de vi

:q ! sortie forcée de vi sans sauvegarde

:w sauvegarde du fichier sur lui-même

:w AutreFichier sauvegarde dans AutreFichier

1.4.2. Déplacements du curseur dans le texte (Pas de : devant)

Ctrl-F (Forward) déplacement d'un écran vers le bas

Ctrl-D (Down) déplacement d'une demi-fenêtre vers le bas

Ctrl-B (Backward) déplacement d'une fenêtre vers le haut

Ctrl-U (Up) déplacement d'une demi-fenêtre vers le haut

Ligne Courante c'est la ligne où se trouve le curseur

Espace déplace le curseur d'un caractère vers la droite

Entrée passage à la ligne suivante

+ passage à la ligne suivante

- passage au début de la ligne précédente

/texte/ déplace le curseur au premier caractère de la première occurrence de texte rencontrée (Attention : une minuscule et une majuscule sont différenciées)

?texte? de même mais en remontant le texte

G déplace le curseur à la fin du texte

w déplace le curseur au début du prochain mot

3w déplace le curseur au début du 3-ème prochain mot

b déplace le curseur au début du mot précédent

5b déplace le curseur au début du 5-ème mot précédent

3H déplace le curseur au début de la 3-ème ligne de la fenêtre (et non du texte)

4L déplace le curseur au début de la 4-ème ligne avant la fin de la fenêtre ou du texte situé sur la fenêtre

M déplace le curseur au début de la ligne située au milieu de la fenêtre

z. centre la fenêtre sur la ligne courante

~ **sur une lettre** transforme la lettre majuscule en minuscule ou le contraire

Selon le terminal utilisé (xterm, vt100,...), les touches “Flèches” peuvent ou non être employées pour déplacer le curseur d’un caractère dans les 4 directions ou d’une fenêtre vers le haut ou le bas.

1.4.3. Suppressions de caractères

x supprime le caractère où se trouve le curseur

4x supprime les 4 caractères à partir du curseur

dd supprime la ligne support du curseur

3dd supprime la ligne courante et les 2 lignes suivantes

D supprime le reste de la ligne courante à partir du curseur

dw supprime le mot courant

1.4.4. Saisie de caractères => Passage en MODE TEXTE

<Echappement> provoque la sortie du mode texte et l’entrée en mode commande

i permet ensuite l’insertion continue de caractères juste avant le curseur et cela jusqu’à la frappe du caractère **<Echappement>**

I idem mais au début de la ligne courante

a permet ensuite l’insertion continue de caractères juste après le curseur et cela jusqu’à la frappe du caractère **<Echappement>**

A idem mais à la fin de la ligne courante

C remplace le reste de la ligne à partir du curseur

1.4.5. Déplacement de textes

La destruction précède le déplacement puis la restitution

Exemple :

/texte/

5dd

10-

p

Les 5 lignes à partir de la prochaine occurrence de texte sont détruites

le curseur est remonté de 10 lignes

Le texte des 5 lignes est restitué à partir de la position du curseur

En fait, toute destruction est mémorisée, le curseur doit être déplacé et le texte mémorisé est restitué par la frappe de **p**

9p restitue les 9 dernières destructions (9 est le maximum permis)

De même, **yy** mémorise la ligne (**5yy** mémorise la ligne courante et les 4 lignes suivantes) et après déplacement **p** les restituent. Ainsi, il est possible de copier une partie du texte.

1.4.6. Exécuter des requêtes (précédées du caractère :)

:début,fin p affiche les lignes de numéro début à fin

début est un numéro de ligne ou . la ligne courante ou .+4 la 4-ème ligne après la ligne courante ou .-3 la 3-ème ligne avant la ligne courante

fin est un numéro de ligne ou . la ligne courante ou comme ci-dessus ou \$ la dernière ligne du texte

sans précision de ligne, la ligne courante est utilisée

:début,fin s/AncienneChaine/NouvelleChaine/

cette requête substitue pour les lignes début à fin l' AncienneChaine par la NouvelleChaine

u restitue l'état antérieur des lignes avant la dernière substitution

Dans la définition d'une chaîne :

. désigne n'importe quel caractère

***** désigne la répétition (voire 0 fois) du caractère précédent

/.*/ désigne n'importe quelle chaîne terminée par ;

(Dans le shell * suffit pour désigner ici .*)

^ désigne le caractère fictif qui précède le premier de la ligne

\$ désigne le caractère fictif qui suit le dernier de la ligne

[a-z] désigne n'importe quelle lettre minuscule (une seule)

[^ 0-9] désigne tout caractère différent d'un chiffre 0 à 9

^ désigne le complémentaire de l'ensemble des caractères qui suit

sauf s'il n'est pas le premier caractère et il désigne alors le caractère lui-même

**** neutralise le caractère spécial (**.** **\$** ***** **^** **[** **]** **)** qui le suit

:s/1\2\.*\3/ substitue 1.2. et le reste de la ligne par 3

:s/a+3/(&)/ encadre de parenthèses les occurrences de a+3

& désigne la chaîne de caractères qui précède

:/\ (chaîne1\)caractère ou non\ (chaîne2\) .../

définit 2 ou plusieurs sous-chaînes limitées par **\(** et **\)** et utilisables ensuite avec **\1** ou **\2**

Exemple : Le texte ci-dessous traité par la commande qui suit

abcdefgXhijklmnoYpqrstuvZwxyz

:s/\(.*\)X\(.*\)Y\(.*\)Z\(.*\))/\3 \1 . \4; \2/

donne

pqrstuv abcdefg . wxyz ; hijklmno

:début,fin g/chaîne/requête fait subir à toute ligne contenant chaîne la requête qui suit

.,.5g / ^ a/s/a/A/ substitue en A sur la ligne courante et ses 5 suivantes le caractère a s'il est le premier caractère de la ligne

Si plusieurs requêtes s'avèrent nécessaires, avant de frapper la touche entrée, il faut frapper le caractère **** pour annihiler la fin de ligne

v en lieu et place de **g** applique les requêtes sur les lignes ne contenant pas chaîne

:r fichier ajoute le texte du fichier à partir de la position du curseur

!:commande exécute la commande Unix puis revient dans vi

Exemple : **!:cd /MonCatalogue**

1.4.7. Les options de vi

:set all liste toutes les options possibles

Une option peut être activée par

:set NomOption ou

:set NomOption=valeur

et supprimée par

:set no NomOption

Les plus intéressantes sont :

autoindent ou en abrégé **ai** autoindentation

autowrite ou **aw** sauvegarde automatique avant :n et :!

ignorecase ou **ic** pas de distinction entre minuscule et majuscule

list affiche les caractères de tabulation selon ^ I

number ou **nu** affiche le numéro de chaque ligne

1.5. UTILISATION DE L'ÉDITEUR nedit :

Cet éditeur simple d'emploi est très utile lorsque X est actif. Il formate le texte du fichier selon le langage employé pour son écriture (C++, Fortran, TeX, ...) et met en évidence les mots clés, indente le texte, montre la parenthèse ouvrante dès la frappe d'une parenthèse fermante, ... De plus, comme il permet l'emploi de la souris pour rectifier le texte, la frappe du texte est beaucoup plus facile.

Il est exécutable soit en tapant la commande `nedit &` ou bien à partir des options

Programs + Editors + NEdit des menus

Ouvrir le fichier à partir de la définition de son nom

Ensuite, utiliser les menus pour éditer, sauvegarder, ouvrir les fichiers, ...

Son utilisation est très intuitive et ses menus sont facilement compréhensibles.

1.6. UTILISATION DE L'ÉDITEUR xemacs :

Cet éditeur est utile lorsque X est actif. Lui aussi, il formate le texte du fichier selon le langage employé pour son écriture. Il est très complet, peut être trop, ce qui au début peut rebuter. En fait, il faut commencer par utiliser les menus, puis retenir les raccourcis les plus utilisés et finalement trouver son propre mode d'emploi.

A partir d'une fenêtre XTerm taper

xemacs NomFichier &

Le fichier est alors chargé et affiché dans une nouvelle fenêtre.

Voici seulement quelques conseils pour faciliter l'initiation à son emploi :

Déplacer le pointeur de souris sur les boutons, sous la barre des menus.

- Le premier bouton (Define a file (affiché en bas à gauche de la fenêtre)) ouvre une boîte de dialogue pour faciliter la recherche du chemin vers le fichier à ouvrir.

- Le second bouton (Edit a directory) ouvre une boîte de dialogue pour lire les noms de fichiers et répertoires d'un répertoire.

- Le troisième bouton (Save buffer) sauvegarde dans le fichier lu le texte actuel. (A utiliser toutes les 5 minutes pour éviter le risque de la perte d'un texte en cours de frappe !)

- Le quatrième bouton (Print buffer) imprime le texte actuel sur l'imprimante par défaut.

- Le cinquième bouton (Kill region) détruit le texte surligné par un clic du bouton gauche de la souris, enfoncé au début du texte, puis relâché à la fin du texte sélectionné.

Attention, le texte n'est pas perdu, mais stocké en mémoire pour être éventuellement collé ensuite (cf le septième bouton).

- Le sixième bouton (Copy region) copie le texte surligné grâce à la souris, stocké pour être collé ensuite (cf le septième bouton).
- Le septième bouton (Paste from clipboard) permet, après positionnement dans le texte par un clic de son bouton gauche de coller ce qui a été détruit (cinquième bouton) ou copié (sixième bouton). En combinant l'emploi des 5, 6 et 7-ièmes boutons, du texte peut être détruit, déplacé ou copié.
- Le huitième bouton (Undo edit) annule la dernière commande en restituant le texte avant son exécution.
- Le neuvième bouton (Check spelling) vérifie l'orthographe du texte.
- Le dixième bouton (Search & replace) demande le texte à changer, validé par Entrée puis, le texte de remplacement. Le changement est proposé à partir de la position courante. Taper "y" pour le changement ou "n" sinon.
- Le onzième bouton (Info documentation) fournit de la documentation sur xemacs.
- Le douzième bouton (Start a compilation) démarre une compilation, le langage étant défini par le suffixe du fichier.
- Le treizième bouton (Start a debugger) démarre un logiciel de débogage si disponible.
- Le quatorzième bouton (Read news) permet la lecture des "news", messagerie autour d'un thème.

Quelques commandes utiles :

Sélectionner un texte + Touche Echappement + lettre l : transforme les lettres majuscules du texte en lettres minuscules.

Sélectionner un texte + Touche Echappement + lettre u : transforme les lettres minuscules du texte en lettres majuscules.

Touche fonction F5 + numéro de la ligne : centre le texte avec la ligne de ce numéro au milieu de la fenêtre

Touche Contrôle + Touche k : détruit le texte de la ligne au delà du curseur.

Touche Contrôle + Touche s : demande le texte à trouver à partir du curseur. Attention, xemacs positionne le curseur sur la première occurrence rencontrée. En cas d'erreur, la touche Backspace efface le dernier caractère du texte à rechercher et le curseur se repositionne sur l'occurrence précédente.

Touche Contrôle + Touche s + Touche s : répète la recherche du texte précédent.

Bien d'autres possibilités sont offertes. À découvrir au fur et à mesure.

1.7. LES VARIABLES d'ENVIRONNEMENT sous le shell BASH

Les variables locales à un processus n'ont pas à être déclarées. Il suffit de les initialiser. Par exemple

VarLocale1=valeur1 VarLocale2=valeur2 ...

La **valeur** de VarLocale1 est \$VarLocale1 ou \${VarLocale1} ce qui en permet la concaténation avec tout autre texte.

Pour rendre ces variables locales permanentes, il faut les "exporter" par la commande **export VarLocale1 VarLocale2**

Dès lors, ces variables deviennent des variables d'environnement utilisables dans n'importe quelle procédure de commandes.

La frappe de la commande **env** liste toutes les variables d'environnement actuelles.

Certaines de ces variables sont définies par le système ou par l'exécution de fichiers de démarrage. Par exemple :

- **HOME** contient le nom du répertoire personnel ;
- **PATH** contient la liste des répertoires où les commandes doivent être recherchées ;
- **CDPATH** contient la liste des répertoires atteignables par la commande **cd** sans avoir à préciser le chemin complet.

Exemples :

```
MEFISTO=/usr/local/mefisto
```

```
PATH=PATH : . :MEFISTO/bin
```

```
INITIER
```

évite d'avoir à taper \$MEFISTO/bin/INITIER

```
CDPATH=$MEFISTO
```

```
cd test
```

évite d'avoir à taper cd \$MEFISTO/test

Remarque : \$(NomCommande) récupère le résultat de l'exécution de la commande.

Par exemple : echo \$(hostname) affiche le nom de la machine de travail.

1.8. LES FICHIERS de DÉMARRAGE

Pour initialiser ou modifier ces variables, ou en créer d'autres, il peut être judicieux d'ajouter des commandes aux fichiers de démarrage. Selon le "shell" utilisé (bash, csh, ksh, zsh, ...) il existe un fichier caché d'initialisation, rangé dans le répertoire personnel, de nom le nom du shell débutant par un point et terminé par "rc" (.bashrc, .cshrc, .kshrc, ...). Leur liste s'obtient par la commande **ls -a \$HOME** et il apparaît au moins

.	.gtkrc-kde	.netscape	..
.acrorc	Desktop	.kde	
.autosave	.ee	.kderc	.qt
axhome	.enlightenment	.kxmlrpcd	.shosts
.bash_history	Mail	.ssh	
.bash_profile	.first_start_kde	.mc	.viminfo
.bash_profile~	.fontalias	.mcp	.wmrc
.bashrc	.forward	.MCOP-random-seed	.Xauthority
.bashrc~	.fullcircle	.mcpirc	.xfigrc
.bashrcold	.gimp	mefisto	.xftcache
capture1.png	.gimp-1.2	.xmaplev5rc	
capture2.png	.gnome	mefistox	.xsession-errors
.gnome-desktop	.mozilla	.xsession-local	.gnome-help-browser .ncftp
.gnome_private	.neditdb		

Le listage à l'aide de la commande **cat** du fichier .bash_profile

```
# .bash_profile
echo "Debut de l'execution du fichier $HOME/.bash_profile"
```

```
# User specific environment and startup programs
export MEFISTO=/usr/local/mefisto
export MEFISTOX=$HOME/mefistox
```



```

export CDPATH=. :$HOME :$MEFISTO
export PATH=$PATH :. :$HOME :$MEFISTO/bin

# User specific aliases and functions
alias la='ls -a'
alias ll='ls -l'
alias lla='ls -al'
alias rtl='ls -rtl'
alias lrt='ls -rtl'
alias rtl='ls -rtl'

#demande de commande avec confirmation
alias rm='rm -i'
alias cp='cp -i'
alias mv='mv -i'

#invite
export PS1='$(hostname) :${PWD} -bash-> '

echo "Fin    de l'execution du fichier $HOME/.bash_profile"

et le fichier .bashrc contient

# User specific aliases and functions
#echo "Entree dans .bashrc sur thot      A EVITER SINON scp PLANTE"

export MEFISTO=/usr/local/mefisto
export MEFISTOX=$HOME/mefistox

export CDPATH=. :$HOME :$MEFISTO
export PATH=$PATH :. :$HOME :$MEFISTO/bin

alias la='ls -a'
alias ll='ls -l'
alias lla='ls -al'
alias rtl='ls -rtl'
alias lrt='ls -rtl'
alias rtl='ls -rtl'

#demande de commande avec confirmation
alias rm='rm -i'
alias cp='cp -i'
alias mv='mv -i'

#invite
export PS1='$(hostname) :${PWD} -bash-> '

#echo "Sortie de .bashrc sur thot      A EVITER SINON scp PLANTE !"

```

Pour modifier son environnement initial, il suffit de modifier le fichier **.bash_profile** du répertoire personnel.

Pour modifier son environnement lors de l'ouverture d'un processus bash, il suffit de modifier le fichier **.bashrc** s'il existe dans le répertoire personnel et sinon de le créer.

Attention à la différence entre les 2 appels :

.bash_profile est exécuté une seule fois au login avant l'appel du premier bash ;

.bashrc est exécuté lors du démarrage de chaque appel de /bin/bash

Il convient donc de ne pas multiplier les concaténations dans les variables exportées du fichier `.bashrc` qui peut être appelé de très nombreuses fois.

1.9. La touche Tabulation (touche du clavier en haut à gauche avec 2 flèches)

Lors d'une frappe de commande, un gain de temps important est obtenu grâce à la touche de tabulation. Très souvent, un nom de répertoire ou de fichier doit être tapé. L'idée est de taper suffisamment de caractères pour que la dénomination soit unique dans le répertoire. La frappe de la touche de tabulation complète alors automatiquement le nom.

Si le choix est multiple, la double frappe de la touche tabulation affiche toutes les possibilités.

1.10. L'historique des commandes frappées

Avant de frapper le premier caractère Sur le clavier, les 4 touches des flèches horizontales ou verticales permettent de se déplacer dans les commandes précédemment tapées :

- < – fait régresser le curseur d'un caractère dans le texte de la commande actuelle
- > fait progresser le curseur d'un caractère dans le texte de la commande actuelle

La flèche vers le haut affiche la commande précédente.

La flèche vers le bas affiche la commande suivante.

1.11. Quelques logiciels pour lire des fichiers spécialisés puis les imprimer

acroread NomFichier.pdf permet la lecture d'un fichier `.pdf` puis de l'imprimer, ...

gv NomFichier.ps permet la lecture d'un fichiers `.ps` ou `.eps` puis de l'imprimer, ...

tex NomFichier.tex permet la compilation du fichier `NomFichier.tex` en le fichier `NomFichier.dvi`

latex NomFichier.tex permet la compilation du fichier `NomFichier.tex` (écrit en latex) en le fichier `NomFichier.dvi`

dvips NomFichier -o permet la traduction du fichier `NomFichier.dvi` en le fichier `NomFichier.ps`

lpr -Plw1c NomFichier.ps permet l'impression sur l'imprimante PostScript de la salle du 1-er étage du fichier. Dans la salle du 6-ème étage, il faut remplacer `lw1c` par `lw6c` pour obtenir le même résultat.