

B.A.-ba pour l'utilisation d'Unix au CTIG

Rédacteur :	Sylvie Nugier
Liste de diffusion :	
Date de dernière mise à jour :	25 novembre 2009
Vérificateur :	
Date de vérification	

PLAN DU DOCUMENT.

Introduction	4
Le contexte historique.....	4
Le système d'exploitation Unix	5
Les interfaces utilisateurs	6
En mode caractère - Telnet et ssh (Putty).....	6
En mode graphique – X-Window et ICA (citrix).....	6
Les commandes de base.....	8
cd - change directory	8
ls - Liste les fichiers du répertoire	8
Bash – utilisation de la complétion	9
Conseils pour nommer les objets	10
Quelques caractères spéciaux	10
Affichage des fichiers	10
Manipulation des fichiers mkdir, rm, cp, mv	11
Les droits avec ls et chmod	11
grep - recherche une chaîne de caractères dans les fichiers.....	13
Autres commandes – ps, vjob, kill, wc, sort, diff, du.....	13
Logiciels et utilitaires proposés	15
Dfile – le gestionnaire de fichiers et nedit.....	15
SAS.....	16
Loadleveler	17
sauvegardes (TSM)	19
FTP – File transfert protocol	19
Annexes.....	20
Exo 1 : Utilisation des commandes cd et ls.....	20

Exo 2 : Manipulation ligne de commande et complétion	20
Exo 3 : Affichage du contenu des fichiers	21
Exo 4 : Manipulation des fichiers et répertoires.....	21
Exo 5 : utilisation des droits	22
Exo 6 : rechercher dans un fichier.....	22
Les principales commandes UNIX (Formation bioinformatique MIG 2004/2005)	24
Un peu d'histoire	26
Logiciels libres et GNU/Linux.....	27

Introduction

Le contexte historique

En 1969, Ken Thompson qui travaille pour les laboratoires Bell développe la première version d'un système d'exploitation mono-utilisateur. Dès 1973, presque tout Unix est réécrit dans le langage C (Dennis Ritchie) ce qui le rend portable.

Les différents UNIX :

- BSD
- System V : AIX, Solaris, HP/UX ...
- XENIX, Free BSD, Linux, Mac OS X

On peut regrouper les premiers utilisateurs d'Unix en deux groupes :

- les universités et centres de recherches américains d'une part. L'université de Californie à Berkeley notamment est à l'origine de l'une des plus anciennes branches d'Unix, avec la *Berkeley Software Distribution* (BSD).
- les grands constructeurs informatiques d'autre part. Chacun de ces grands groupes a développé Unix selon ses propres besoins et intérêts, créant ainsi son propre Unix comme Solaris pour Sun Microsystems ou AIX pour IBM (Unix sur dga2). C'est l'autre grande branche d'Unix : *Unix System V* (lire « cinq »).

Ouvert -> Portable -> succès -> -> ! Contre-productivité

De nos jours, les systèmes UNIX propriétaires, longtemps majoritaires dans l'industrie et l'éducation, sont de moins en moins utilisés. En revanche, trois systèmes de type UNIX basés sur BSD d'une part, et le système GNU/Linux, compatible UNIX, d'autre part, ainsi que Mac OS X, occupent une part de marché de plus en plus importante.

Logiciels libres

- La liberté d'exécuter le programme, pour tous les usages
- La liberté d'étudier le fonctionnement du programme, et de l'adapter à vos besoins : l'accès au code source
- La liberté de redistribuer des copies, donc d'aider votre voisin
- La liberté d'améliorer le programme et de publier vos améliorations, pour en faire profiter toute la communauté

Un logiciel libre est un logiciel dont la licence dite libre donne à chacun le droit de l'utiliser pour tous les usages, d'étudier son fonctionnement voire de l'adapter à ses besoins ou de l'améliorer (l'accès au code source est nécessaire), de le diffuser (copier, donner et vendre). Richard Stallman a formalisé la notion de logiciel libre dans la première moitié des années 1980 en créant la Free Software Foundation afin de promouvoir le projet GNU.

En 1991, l'étudiant finlandais Linus Torvalds entreprend d'écrire un noyau de système d'exploitation qu'on appellera plus tard « noyau **Linux** ».

L'association du projet GNU avec le noyau Linux se produit naturellement, l'arrivée du noyau Linux rend disponibles les logiciels du projet GNU sur les ordinateurs animés par des microprocesseurs de la famille Intel x86. Cela leur assura une large diffusion car ces projets étaient complémentaires.

Le système d'exploitation Unix

Un système d'exploitation (ou « OS » pour operating system) est un ensemble de programmes capable de faire fonctionner un ordinateur et ses périphériques pour que les utilisateurs puissent créer, exécuter, manipuler et archiver des programmes ou des données.

Un système multi-utilisateur est conçu pour que plusieurs utilisateurs puissent profiter des mêmes ressources simultanément.

Un système multi-tâches est capable de mener de front plusieurs tâches à la fois, c'est-à-dire d'exécuter plusieurs programmes en quasi-simultanéité pour les utilisateurs (qui n'ont du temps qu'une perception imprécise) mais séquentiellement, pour lui, pendant de courts laps de temps de l'ordre d'une fraction de seconde.

Unix est un système d'exploitation multi-utilisateurs et multi-tâches.

Unix se compose :

- d'un noyau (cœur du système chargé en mémoire vive s'occupant de la gestion des processus, de la mémoire, des entrées/sorties, implémentation des systèmes de fichiers, des protocoles réseau ...),
- d'un système de gestion de fichiers hiérarchisé,
- d'interpréteurs de commandes (exemple : Korn shell ou Bourne-Again shell).
Bash est l'interprète par défaut sur de nombreux Unix libres, notamment sur les systèmes GNU/Linux (porté sous Windows par le projet Cygwin). Dans les différents systèmes d'exploitation Microsoft Windows, le programme analogue est command.com, ou cmd.exe. (une coquille est un récipient qui permet de manipuler toutes sortes de contenus), il est utile pour un usage interactif ou de la conception de script.
- de commandes, (internes (ex: cd) / externes -script de commande ou /usr/bin/ls - commande type)
- d'utilitaires (programmes comme les scripts),
- de services de communication (basés sur le protocole Internet TCP/IP),
- d'un environnement graphique, (X-Window, bureau comme CDE, KDE ou gnome)
- d'applications pour les utilisateurs.



Unix distingue les majuscules des minuscules (tant au niveau des commandes/options que des noms de fichiers).

Les interfaces utilisateurs

En mode caractère - Telnet et ssh (Putty)

Exemple avec telnet :

→ démarrer → Exécuter → cmd	telnet dgaprod.jouy.inra.fr login : ugenxxx ugenxxx's Password : ***** ls (commande Unix) exit
-----------------------------	--

- 1- Installer l'utilitaire Putty (trouvé sur Internet).
- 2- Exécuter putty.exe
- 3- Utilisation de **SSH** (secure shell) protocole de communication sécurisé avec **clé de chiffrement**.
Utilisation sur le serveur de communication **dgasp4**.

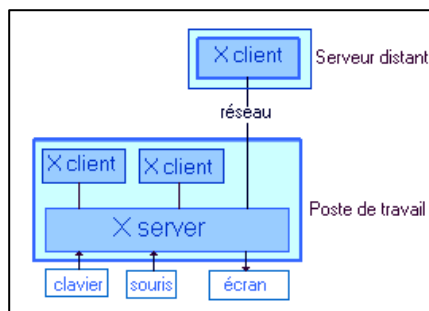
→ « Session »	champ « Host Name » : 193.54.97.163 Connection type : cocher « SSH » (Port=22) « Saved Sessions » : ctigdgasp4 → Save → Open
---------------	--



Ces connexions se font en mode caractères, elles peuvent être utiles pour changer le mot de passe de dgasp4 (commande : passwd).

En mode graphique – X-Window et ICA (citrix)

Avec un émulateur X-Window comme Xming ou celui de Cygwin, on affiche le bureau du serveur.



X-Window (ou X11) fonctionne suivant le modèle client/serveur :

- le logiciel serveur X tourne sur une machine qui est dotée d'un écran, d'un clavier et d'une souris ; il reçoit et sert des requêtes d'affichage, d'entrées de texte et de déplacement de souris sur un port logiciel,
- un logiciel client X (logiciel graphique) se connecte au serveur X et lui envoie ses requêtes d'affichages en utilisant le protocole X au travers de la bibliothèque X (Xlib).

Parmi les clients X, on distingue le gestionnaire de fenêtres dont le rôle est de gérer l'affichage, la sélection, le déplacement, le re-dimensionnement et les décorations des fenêtres.

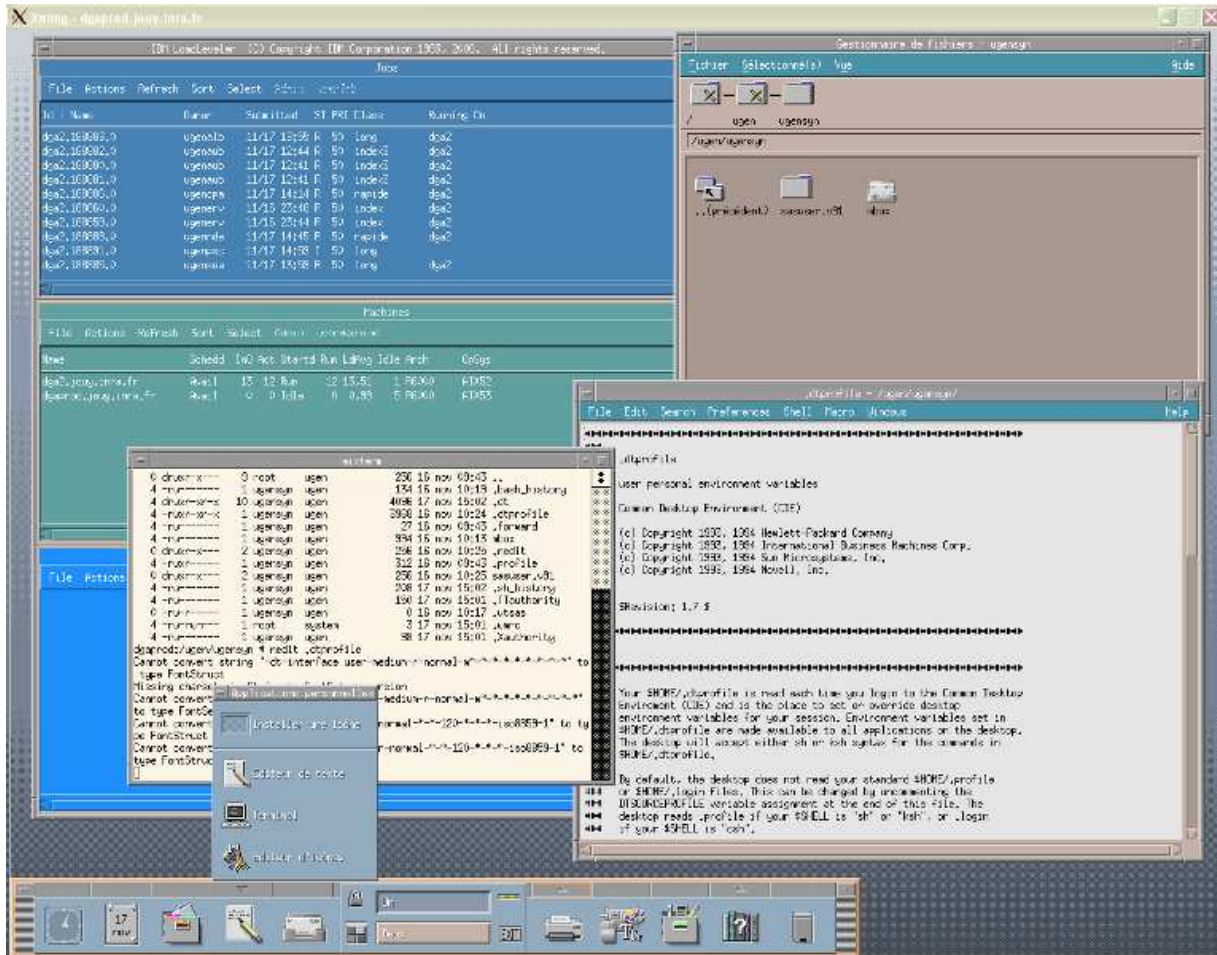


Figure 1- Le bureau CDE

Le bureau CDE (Common Desktop Environment) est utilisé par la plupart des fabricants de stations UNIX. Pour vous permettre d'accéder à ses fonctions, le bureau CDE place le panneau de contrôle au bas de votre écran. Il comporte une série d'icônes qui lancent divers programmes ou services. La flèche vers le haut au-dessus de chaque icône permet d'ouvrir le menu de l'icône qui présente d'autres applications.

Il existe d'autres bureaux comme KDE et Gnome plus répandus sous Linux.

Le client Citrix

Le client de la société CITRIX utilise le protocole ICA (Independent Computing Architecture) proposé par le serveur metaframe. C'est un protocole moins lourd que le protocole X11 (X-Window) utilisé habituellement. Il fournit un environnement graphique.

Il est utilisé au CTIG au travers d'un tunnel crypté via le client putty et le serveur dgas4.

Voir le document détaillé sur http://dgas1.jouy.inra.fr:49213/doc_ctig/ :

→ « accès via Metaframe » → « Accès ICA par tunnel SSH (téléchargement version PDF) »

Le bureau CDE simplifié apparaît sur dgas4, il permet de se connecter à dga2 en ouvrant une fenêtre aixterm ou xterm.

Les commandes de base

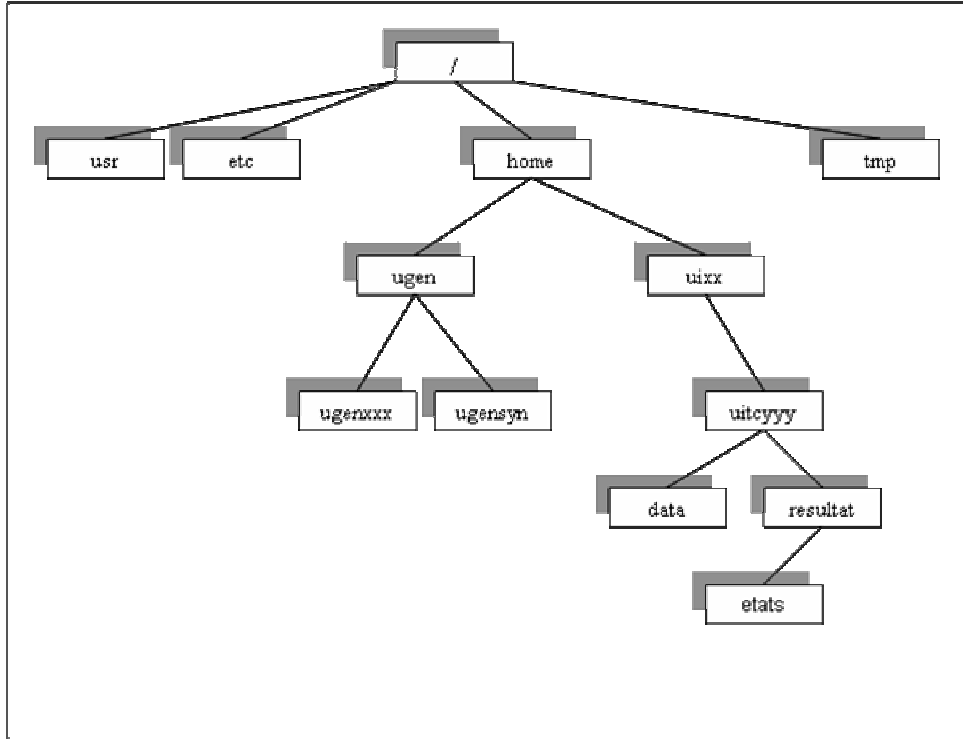


Figure 2 - Système de fichiers – Arborescence

cd - change directory

Permet de naviguer dans l'arborescence des répertoires du système de fichiers.

cd /ugen/ugenxxx/dossier1	on indique ici le chemin absolu du répertoire dossier1 car on part de la racine (root) /.
cd dossier1	on indique le chemin relatif du répertoire (le nom ne commence pas par /). C'est à dire à partir de là où l'on se trouve, ici : /ugen/ugenxxx .
cd	cd sans nom de répertoire permet de se positionner sur le répertoire de base (home directory). Ce répertoire peut être aussi représenté par le caractère ~ .
cd ..	permet de remonter sur le répertoire supérieur dans l'arborescence.

ls - Liste les fichiers du répertoire

ls -ls

Permet d'afficher entre autres les droits par rapport aux fichiers et répertoires.

dga2:/ugen/ugenxxx/répertoire # **ls -ls** (voir aussi **ls -ls *** ou **ls -Rls ***)

```
4 -rw-r----- 1 ugenxxx ugen 698 26 Mar 16:05 fichier1
```



```
4 -rw-r----- 1 ugenxxx ugen      698 26 Mar 16:05 fichier2
4 -rw-r----- 1 ugenxxx ugen      698 26 Mar 16:05 fichier3
4 drwxr-s---  2 ugenxxx ugen      512 26 Mar 16:17 sous_rep1
```

Pour obtenir des informations sur le répertoire courant :

dga2:/ugen/ugenxxx/répertoire # **ls -lsd .**

```
4 drwxr-s---  2 ugenxxx ugen      512 26 Mar 16:05 .
```

Pour afficher les fichiers par ordre de date de création croissante : **ls -ltr**

Voir dans l'annexe l'exercice « EXO 1 ».



La plupart du temps l'ordre des options dans la commande n'a pas d'importance, la commande « **ls -ls** » est équivalent à « **ls -sl** ». Les options sont cumulatives.



La commande **tree** affiche l'arborescence du répertoire (cette commande n'existe pas toujours sous tous les Unix).



La commande **man** est l'aide de base pour les commandes Unix, **man ls** donne le détail des options de la commande **ls**.

Bash – utilisation de la complétion

Shell bash, raccourci clavier (emacs) :

- Les flèches (<- et ->) : déplacement du curseur dans le texte de la ligne de commande
- Ctrl d : efface le caractère courant
- Ctrl a : saut en début de ligne
- Ctrl e : saut en fin de ligne
- Backspace : effacer le caractère précédent

Le shell permet à l'utilisateur de dialoguer avec le système. Il interprète les commandes, les exécute, gère l'environnement (variables, alias) et les ouvertures/fermetures de connexion.

Il existe plusieurs types de shell. Sous Linux le shell **bash** est utilisé par défaut, sous AIX c'est le Korn shell qui est proposé. La syntaxe diffère quelque peu entre ces deux shells, notamment pour le positionnement des variables d'environnements et la manipulation de la ligne de commande. L'utilisation du shell

bash (Linux) est plus confortable notamment avec la complétion qui s'obtient avec la touche **Tab** (->).

On peut changer à tout moment de shell, et prendre un shell différent du shell par défaut. Il suffit de taper le nom du shell pour passer sous son environnement (exemple : **ksh** ou **bash**).

La commande **history** affiche l'ensemble des commandes en mémoire liées au shell en cours d'exécution.

Voir dans l'annexe l'exercice « EXO 2 ».



Pour interrompre une commande en cours d'exécution on envoie un signal via l'appui simultané sur les touches Contrôle et C.

Conseils pour nommer les objets

Quelques conseils pour se simplifier la vie notamment en cas d'échange de fichiers avec d'autres systèmes d'exploitation :

- Ne pas utiliser de caractères accentués.
- Ne pas utiliser le caractère espace car il s'agit du caractère séparateur. Sinon mettre le nom comprenant un blanc entre le caractère double-quote (").
- Utiliser uniquement les caractères alphanumériques (N'insérez pas les caractères comme / et \ ou !)
- Unix fait la différence entre les lettres minuscules et majuscules. Ainsi, les noms « Resultat », « RESULTAT » et « resultat » sont tous les trois différents.
- La taille d'un nom de fichier est de 256 caractères maximum.
- utiliser le point « . », le tiret « - » ou le souligné « _ » ainsi que la majuscule comme délimiteur dans un nom de fichier
- Ne pas hésiter à utiliser des extensions (suffixes) logiques en fonction du type de fichier. Par exemple projet.out pour le résultat.

Quelques caractères spéciaux

..	pour le répertoire supérieur
~	pour le répertoire de base (home directory)
.	pour le répertoire courant (celui où l'on se trouve)
*	permet de remplacer des caractères dans les noms des fichiers
>	dirige le résultat de la commande dans un fichier
>>	ajoute le résultat de la commande à la fin d'un fichier
2>	dirige les messages d'erreurs de la commande dans un fichier
	<p>passse le résultat de la commande à la commande suivante.</p> <p>ls ar* qprt -Pprt7s imprime sur la prt7s le résultat de la liste des fichiers commençant par ar.</p>
&	Ce caractère ajouté à la fin d'une commande permet de la lancer en arrière plan, la ligne de commande n'est pas bloquée.

Affichage des fichiers

more NomFichier	<p>Affichage paginé du fichier.</p> <p>q : pour sortir (ou CTRL+C)</p> <p>/ « chaine » : recherche d'une chaîne de caractère</p> <p>Barre espace : page suivante</p> <p>b : remonter d'une page</p> <p>Retour chariot : ligne suivante</p>
cat NomFichier	<p>Affichage non paginé du fichier.</p> <p>Utile avec une redirection vers un autre fichier ou vers une commande :</p>

	cat nom1 nom2 > nom3 : Concatène deux fichiers dans un 3ème.
head -n NomFichier	Affiche les n premières lignes du fichier.
tail -n NomFichier	Affiche les n dernières lignes du fichier.

Voir dans l'annexe l'exercice « EXO 3 ».

La plupart des résultats des commandes peuvent être obtenus avec des outils graphiques mais leur intérêt apparaît évident lors de l'automatisation des tâches avec l'écriture de script.

Manipulation des fichiers mkdir, rm, cp, mv

mkdir NomRepertoire	make directory, création du répertoire NomRepertoire, à partir de la position où l'on se trouve.
rm NomFichier	remove directory, pour effacer un ou plusieurs fichiers
rm -R NomRepertoire	pour effacer un répertoire et toute l'arborescence en dessous.
rm -i Nomfichier*	Pour effacer les fichiers commençant par Nomfichier avec demande de confirmation.
cp nom-fic-depart nom-fic-arrive	copie un fichier
cp -R nom_rep nom-rep2	copie d'un répertoire et de son arborescence.
mv ancien-NomFic nouveau-NomFic	permet de déplacer des fichiers et de changer leur nom.

Attention pas de corbeille avec la ligne de commande sous Unix !!

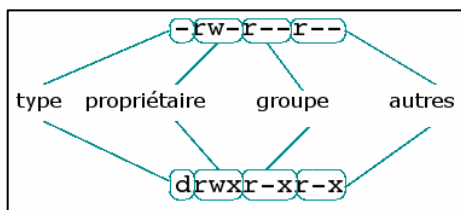
On peut avec la commande **alias** ajouter l'option **-i** aux commandes **rm**, **mv** et **cp** pour demander une confirmation avant remplacement d'un fichier existant. Exemple : `alias rm='rm -i'`

La commande **alias** dans le fichier `.env` sur AIX est valable pour la session.

Pour effacer un répertoire contenant un nombre de fichiers importants sans demande de confirmation utiliser l'option **-f** ou enlever l'alias par `unalias rm`.

Voir dans l'annexe l'exercice « EXO 4 ».

Les droits avec ls et chmod



```
dga2:/ugen/ugenxxx # ls -lsd *
```

```
4 -rw-r--r-- 1 ugenxxx ugen 698 26 Mar 16:05 fic1
```

```
4 -rw-r--r-- 1 ugenxxx ugen 698 26 Mar 16:05 fic2
```

```
4 -rw-r--r-- 1 ugenxxx ugen 698 26 Mar 16:05 fic3
```

```
4 drwxr-sr-x 2 ugenxxx ugen 512 26 Mar 16:17 rep1
```

Le deuxième champ `"- rw- r-- r--"` se lit ainsi:

- Lorsque le premier caractère est `"-"`, on a affaire à un fichier.

Les 3 caractères suivants `"rw-"` expliquent vos droits sur le fichier.

`r` autorise la lecture, `-` ne l'autorise pas

w autorise l'écriture, - ne l'autorise pas

x ou s autorise l'exécution du fichier (n'a de sens que s'il est conçu pour être une commande).

Les 3 autres caractères r-- expliquent les droits des utilisateurs de votre groupe (ugen) sur le fichier et enfin les 3 derniers caractères expliquent les droits des autres c'est à dire les utilisateurs ne faisant pas partie de votre groupe.

- Lorsque le premier caractère est "d", on a affaire à un répertoire.

La suite du champ s'interprète de la même façon que pour un fichier : vos droits pour les 3 caractères suivants, les droits du groupe pour les 3 caractères suivants et les droits des autres pour les 3 derniers caractères.

Pour un répertoire, la signification des caractères est:

r autorise la lecture du contenu du répertoire,

w autorise l'ajout ou la suppression d'un fichier dans le répertoire (quelque soit les droits du fichier),

x ou s autorise le passage par ce répertoire pour accéder aux fichiers ou sous-répertoires.

```
chmod [-R] [u][g][o] {-/+=} [r][w][x] {nom_fichier/nom_répertoire}
```

La commande **chmod** permet de changer les valeurs des autorisations.

-R permet de changer les droits de manière récursive dans l'arborescence des répertoires.

u pour user (vos droits), **g** pour le groupe, **o** pour others (les autres).

- retire le droit suivant (**r** ou **w** ou **x**), + ajoute le droit suivant, = impose le droit suivant.

chmod g+w repertoire1	Autorise les utilisateurs de votre groupe à ajouter et à supprimer des fichiers dans repertoire1.
chmod go-w+x monrep	Empêche les utilisateurs de votre groupe et les autres de créer ou de supprimer des fichiers dans monrep, les autorise à parcourir monrep avec une commande ou dans un chemin.
chmod u=rwx cmd1 cmd2	Donne les droits de lecture, écriture et exécution au fichier cmd1 et cmd2 pour le propriétaire seulement, empêche les autres utilisateurs d'accéder à ces fichiers (go=).
chmod -R g+w fic*	Donne les droits d'écriture aux utilisateurs du groupe pour tous les fichiers et répertoires commençant par fic dans toute l'arborescence.

Pour travailler à plusieurs dans un répertoire, 2 solutions :

- Demander la création d'un nouveau groupe puis utiliser la commande chmod pour gérer les droits des fichiers et sous répertoires, au besoin utiliser des scripts pour affecter les droits avec chmod automatiquement.
- Utiliser les ACL (Access Control Lists), cette fonctionnalité permet de gérer plus finement les droits d'accès en ajoutant des permissions étendues (supplémentaires) aux fichiers et répertoires.

La commande `ls -lsde` affiche un + en onzième caractère (ex : `-rw-r--r--+`) qui permet de savoir si le fichier possède des accès étendus.

Voir dans l'annexe l'exercice « EXO 5 ».



La commande `lsgroup` permet de connaître les membres d'un groupe (ex : `lsgroup uixx`).

La commande `chgrp` permet de changer le groupe d'un fichier et d'un répertoire.

Il peut exister des procédures d'ACL qui tourne toutes les nuits pour mettre en place la politique d'autorisation.

Si une modification des droits doit perdurer, le signaler à `ctig.system@dga.jouy.inra.fr`.

grep - recherche une chaîne de caractères dans les fichiers.

La commande `grep` (Get Regular Expression Print) extrait d'un ou plusieurs fichiers les lignes de texte dont tout ou partie correspond à un motif. Elle permet donc de retrouver dans un ensemble de fichiers, les lignes qui contiennent une chaîne de caractère répondant à des critères donnés.

grep 'motif' {fichier(s)}

-i : ne distingue pas les majuscules et les minuscules

-n : affiche aussi le numéro de ligne

-l : n'affiche que le nom des fichiers où chaîne a été trouvée

-c : affiche uniquement nombre de lignes contenant chaîne

-h : n'affiche pas le nom de fichier, normalement placé devant chaque ligne

-v = : affiche toutes les lignes ne satisfaisant pas la condition

Exemples :

`grep lundi semaine_log` -> cherche le mot lundi dans le fichier `semaine_log`.

`grep -n lundi sem*` -> cherche lundi dans tous les fichiers commençant par `sem` ; -n affiche le numéro de la ligne dans le fichier.

Il existe aussi la commande `egrep` (extended grep) plus puissante :

`egrep '(Jules|Jim)' fichier` : affiche toutes les lignes de *fichier* contenant "Jules" ou/et "Jim"

Voir dans l'annexe l'exercice « EXO 6 ».

Autres commandes – ps, vjob, kill, wc, sort, diff, du

`ps -eaf` Affiche les processus (commandes et leurs environnements) en cours.

`ps -eaf | grep ugenxxx` -> sélectionne dans la liste des processus ceux qui contiennent le mot `ugenxxx`.

`vjob ugenxxx script` (procédure) sélectionnant les processus contenant la chaîne de caractères `ugenxxx`.

`kill pid` permet d'arrêter un processus.

`wc` retourne le nombre de ligne d'un fichier ex : `ls | wc -l`

`sort` permet de trier un fichier

`diff` compare des fichiers deux à deux

du -k nomrep affiche la taille en ko d'un répertoire

Voir l'annexe « les principales commandes Unix »

Logiciels et utilitaires proposés

Dtfile – le gestionnaire de fichiers et nedit

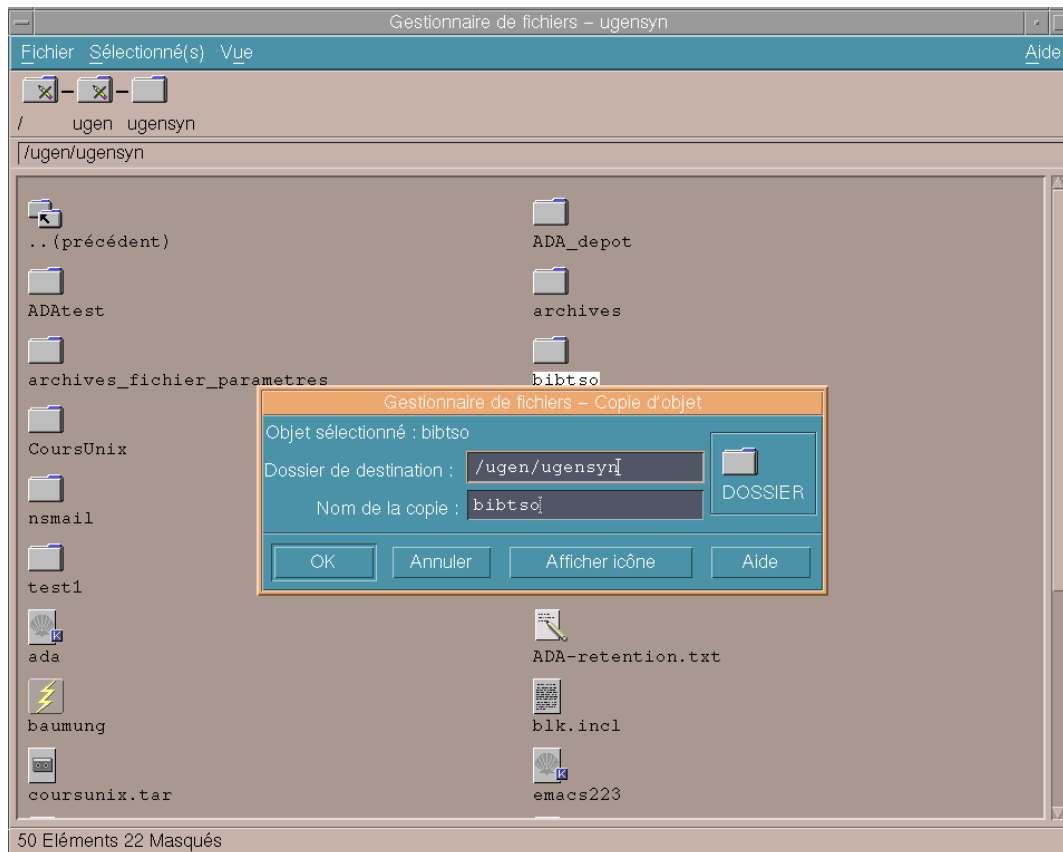


Figure 3 – dtfile

Le gestionnaire du bureau dtfile permet d'effectuer la plupart des commandes de base sur les fichiers et répertoires sans connaître les commandes.

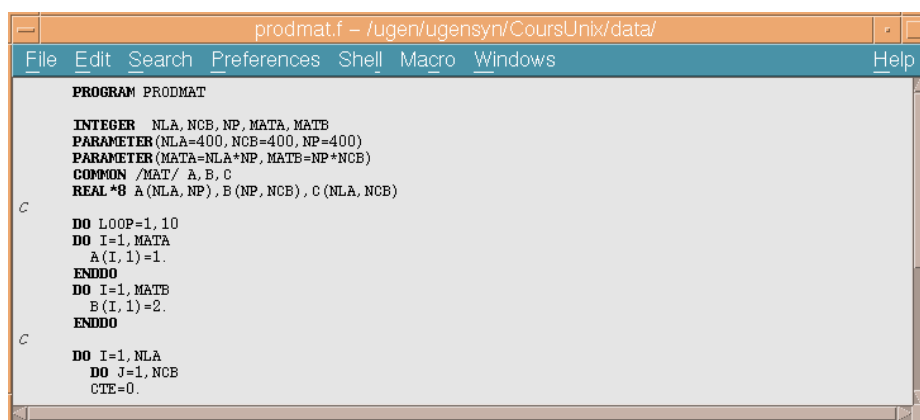
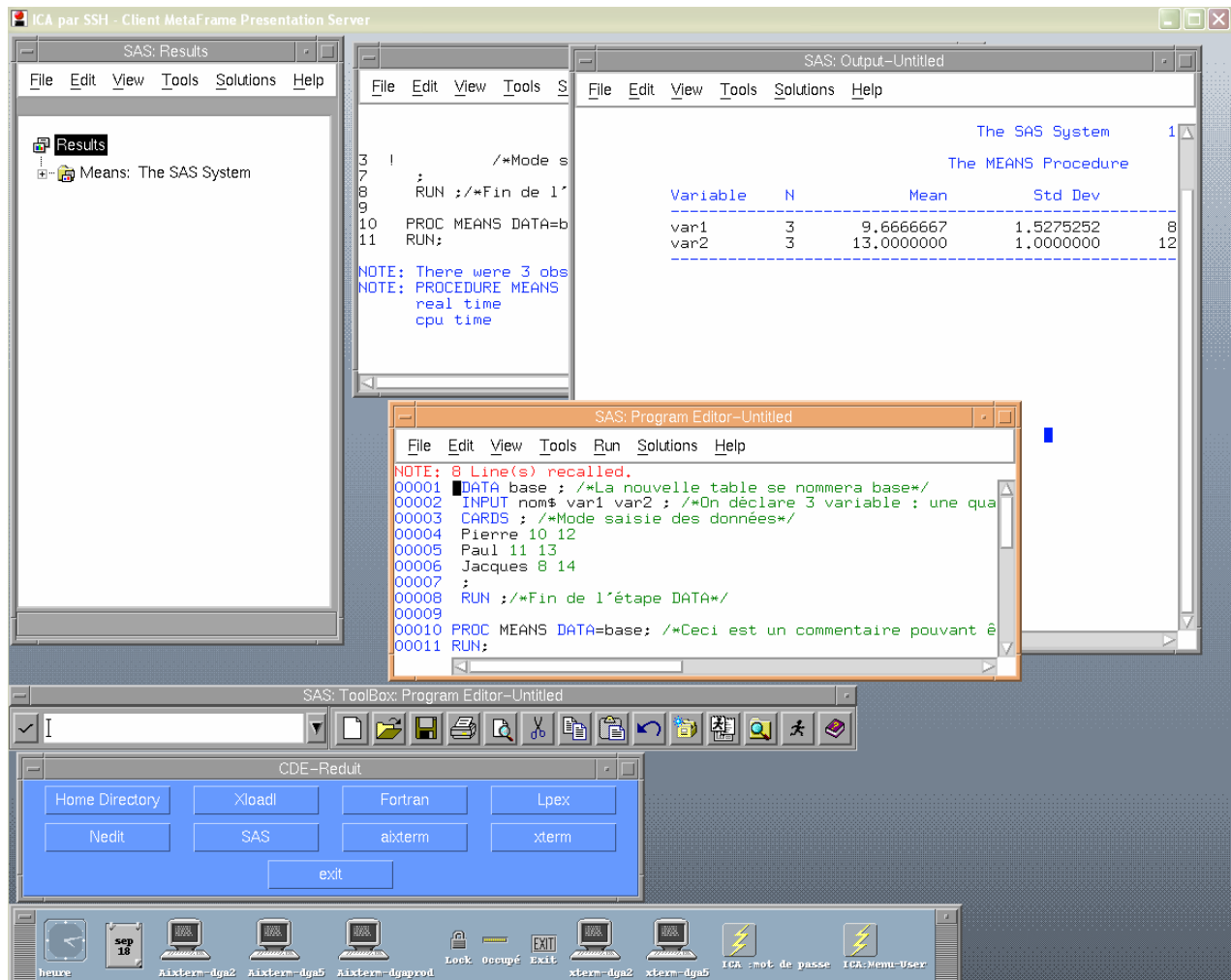


Figure 4 - nedit

nedit est un éditeur simple et intuitif mis à disposition.

SAS



Le langage de commande de SAS, pour Statistical Analysis System, est un langage de programmation de quatrième génération (L4G) édité par « SAS Institute ». Il existe depuis plus de trente ans.

Le logiciel traditionnel se compose d'un ensemble de modules permettant de répondre aux besoins suivants par la programmation :

1. Création et gestion de bases de données
2. Traitements analytiques des bases de données
3. Création et diffusion de rapports de synthèse et de listing.

Quelques modules :

- Base SAS
- SAS/ACCESS (un module spécifique par Base de Données)
- SAS/FSP (pour la création d'interface)
- SAS/GRAPH (pour créer des graphiques)

- SAS/STAT (permettant l'accès à toutes les procédures statistiques)

Le langage SAS est basé sur trois parties :

- Les étapes DATA. Elles permettent de répondre aux besoins correspondant à la création et à la gestion de bases de données. Une étape crée ou modifie une table d'observations. En colonne se trouvent le plus souvent un identifiant et des indicateurs et en ligne, les sujets observés.

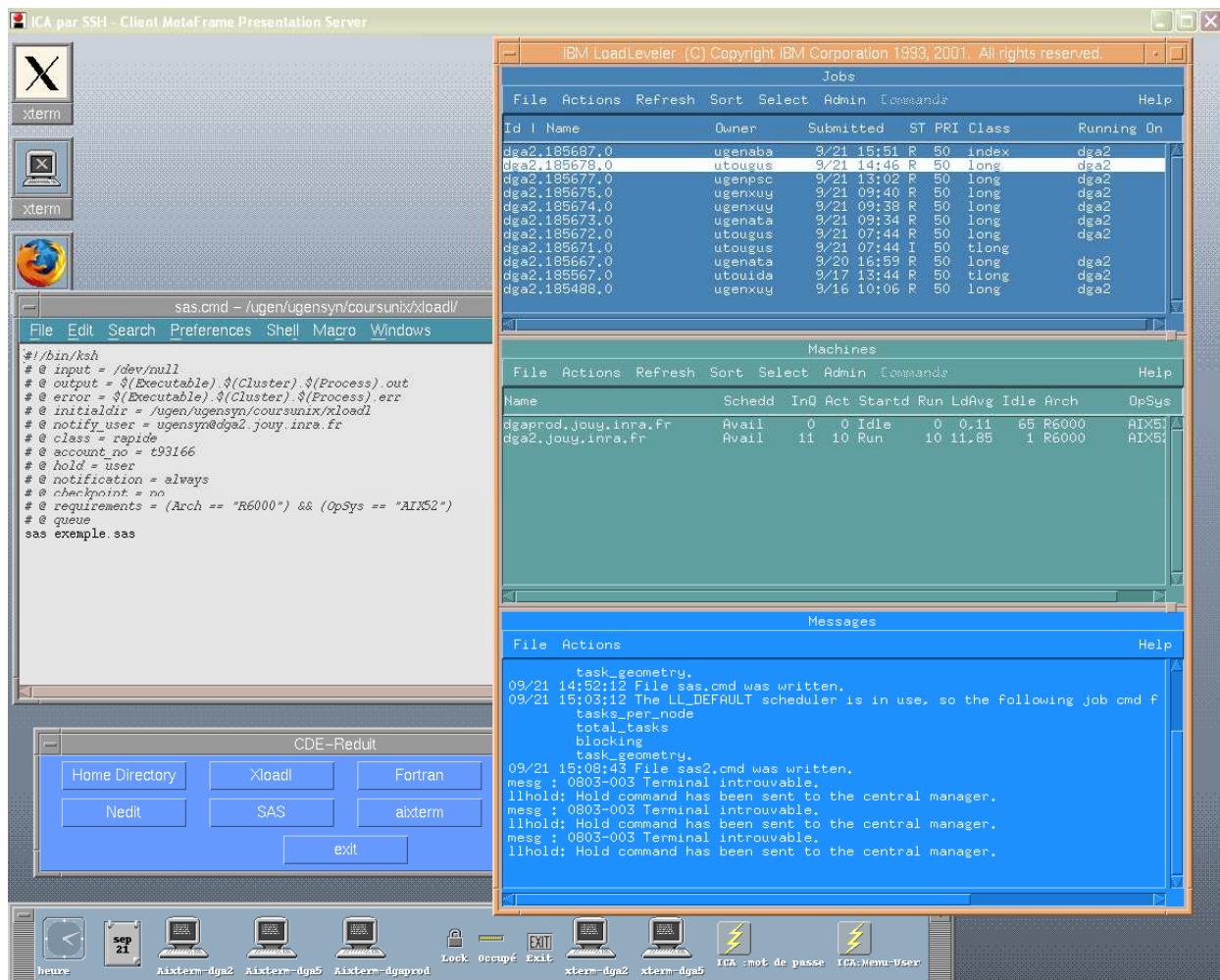
- Les procédures ont chacune leur objectif : l'une permet de trier une table, l'autre de compter les lignes ou de synthétiser les indicateurs... ou simplement d'imprimer la table ou un échantillon.

- Le langage MACRO est un langage permettant d'automatiser les traitements.

Pour en savoir plus : <http://fabrice-morlais.developpez.com/tutoriels/sas/initiation-sas/>

« sas NomProgramme.sas » permet d'exécuter un programme SAS sans interface graphique, un fichier log (comment s'est déroulé l'exécution) et lst (listing contenant les sorties) sont créés pendant l'exécution du programme.

Loadleveler



The screenshot shows a terminal window with the following content:

```

sas.cmd - /ugen/ugensyn/coursunix/xload/
File Edit Search Preferences Shell Macro Windows
#!/bin/ksh
#@ input = /dev/null
#@ output = $(Executable) $(Cluster) $(Process).out
#@ error = $(Executable) $(Cluster) $(Process).err
#@ initialdir = /ugen/ugensyn/coursunix/xload
#@ notify user = ugensyn@dga2.jouy.inra.fr
#@ class = rapide
#@ account_no = t93166
#@ hold = user
#@ notification = always
#@ checkpoint = no
#@ requirements = (Arch == "R6000") && (OpSys == "AIX52")
#@ queue
sas exemple.sas
  
```

The LoadLeveler Jobs window displays the following table:

Id	Name	Owner	Submitted	ST	PRI	Class	Running	On
dga2.185687.0		ugenaba	9/21 15:51	R	50	index	dga2	
dga2.185678.0		utougus	9/21 14:46	R	50	long	dga2	
dga2.185677.0		ugenpsc	9/21 13:02	R	50	long	dga2	
dga2.185675.0		ugenxuy	9/21 09:40	R	50	long	dga2	
dga2.185674.0		ugenxuy	9/21 09:38	R	50	long	dga2	
dga2.185673.0		ugenata	9/21 09:34	R	50	long	dga2	
dga2.185672.0		utougus	9/21 07:44	R	50	long	dga2	
dga2.185671.0		utougus	9/21 07:44	I	50	tlong	dga2	
dga2.185667.0		ugenata	9/20 16:59	R	50	long	dga2	
dga2.185667.0		utouida	9/17 13:44	R	50	tlong	dga2	
dga2.185488.0		ugenxuy	9/16 10:06	R	50	long	dga2	

The Machines window displays the following table:

Name	Schedd	InQ	Act	Startd	Run	LdAvg	Idle	Arch	OpSys
dgaprod.jouy.inra.fr	Avail	0	0	Idle	0	0.11	65	R6000	AIX52
dga2.jouy.inra.fr	Avail	11	10	Run	10	11.85	1	R6000	AIX52

The Messages window displays the following text:

```

task_geometry.
09/21 14:52:12 File sas.cmd was written.
09/21 15:03:12 The LL_DEFAULT scheduler is in use, so the following job cmd f
tasks_per_node
total_tasks
blocking
task_geometry.
09/21 15:09:43 File sas2.cmd was written.
msg : 0803-003 Terminal introuvable.
llhold: Hold command has been sent to the central manager.
msg : 0803-003 Terminal introuvable.
llhold: Hold command has been sent to the central manager.
msg : 0803-003 Terminal introuvable.
llhold: Hold command has been sent to the central manager.
  
```

Loadleveler est un gestionnaire de travaux en batch (traitement par lots).

Pour soumettre un job à Loadleveler, il faut préparer un fichier de commandes, puis le soumettre en lançant la commande suivante : **lsubmit xloadl_sample**

Exemple de fichier de commandes :

```
#!/bin/ksh
# @ input = /dev/null
# @ output = $(Executable).$(Cluster).$(Process).out
# @ error = $(Executable).$(Cluster).$(Process).err
# @ initialdir = /ugen/ugenxxx/coursunix
# @ notify_user = ugenxxx@dga2.jouy.inra.fr
# @ class = rapide
# @ account_no = tyxxx
# @ hold = user
# @ notification = always
# @ checkpoint = no
# @ requirements = (Arch == "R6000") && (OpSys == "AIX52")
# @ queue
sas exemple.sas
```

account_no : numéro de travail, il doit être obtenu auprès de vos informaticiens.

Initialdir : répertoire de travail où seront créés ou lus les fichiers qui ne sont pas référencés par leur chemin absolu.

hold : Le job ne démarrera que lorsque son propriétaire l'aura dé-heldé (XLOADL -> Sélection du job puis -> Actions -> Release From Hold)

startdate : Le job ne démarrera pas avant l'heure et le jour spécifié.

Les lignes commençant par #@ sont prises en compte, celles commençant par # ... sans le caractère @ sont considérées comme des commentaires.

Etats d'un job:

I Idle (n'a pas encore tourné)
R Running
H Job holdé par un user
S Job holdé par le système
D Deferred (traitement différé)

Lorsque l'on soumet plusieurs jobs de quelques heures de CPU les uns derrière les autres, il faut éviter de réquisitionner les machines et de créer ainsi une file d'attente de jobs se désespérant de tourner.

Classes d'exécution

rapide	long	mlong	tlong	index	index2	nombre max de travaux actifs
5	10	1	1	2	1	18

classe	temps CPU maximum (hh:mm:ss)
rapide	01:00:00
long	48:00:00
mlong	192:00:00 (réservée aux utilisateurs du groupe gena (programme BLAST))
tlong	240:00:00
index	illimitée (réservée aux utilisateurs du groupe ibl)
index2	illimitée (réservée aux utilisateurs du groupe ibl)

Plus de détails dans http://dgasp1.jouy.inra.fr:49213/doc_ctig/

Cette base d'information est accessible à partir d'un navigateur.

sauvegardes (TSM)

Les espaces utilisateurs sont sauvegardés toutes les nuits.

Un fichier est sauvegardé s'il a été modifié depuis la dernière sauvegarde. 3 versions par fichier sont gardées. Au bout de 90 jours de non modification du fichier sur disque, les 2 sauvegardes les plus vieilles sont supprimées.

Lorsque le fichier est effacé du disque, la dernière version de sauvegarde est gardée 40 jours.

Pour demander une restauration de sauvegarde, envoyez un mail à ctig.system@dga.jouy.inra.fr

FTP – File transfert protocol

Protocole utilisé pour copier des fichiers et des répertoires entre différents systèmes.

Des clients comme Filezilla ou Ws_ftp propose une interface graphique.

Après authentification sur la machine cible (exemple : **ftp dga2.jouy.inra.fr**), on peut entrer les commandes suivantes :

Help	liste les commandes ftp
quit	quitte ftp
CD dir	change le directory en cours sur le système cible
LS	liste les fichiers du directory en cours sur le système cible
PWD	affiche le nom du directory en cours sur le système cible
BINARY	transfert sans conversion
ASCII	transfert (conversion si système hôte EBCDIC)
GET fcible fhôte	transfert du fichier fcible vers le fichier fhôte
PUT fhôte fcible	transfert du fichier fhôte vers le fichier fcible

Annexes

Exo 1 : Utilisation des commandes cd et ls

Se positionner sur le répertoire scratch à la racine du serveur	cd /scratch
Lister le contenu de scratch.	ls
Se positionner sur son répertoire à partir du répertoire scratch.	cd ctigxxx
Lister le contenu de votre répertoire.	ls
Remonter d'un niveau	cd ..
Lister le contenu de scratch. avec les informations principales concernant son contenu.	ls -ls
Se positionner à la racine du serveur.	cd /
Aller dans le répertoire bin se trouvant sous ssys à partir de la racine, lister son contenu.	cd ssys/bin ls
Aller sur CoursUnix se trouvant sous ugensyn.	cd /ugen/ugensyn/CoursUnix
Lister le contenu avec les détails	ls -ls
Lister le contenu avec le 1 ^{er} niveau des sous-répertoires	ls -ls *
Lister le contenu avec toute l'arborescence	ls -Rls * more
revenir sur son répertoire de base	cd
lister les fichiers cachés	ls -als

Exo 2 : Manipulation ligne de commande et complétion

Aller sur CoursUnix se trouvant sous ugensyn.	cd C-> % C+touche tab %
Lister le contenu.	ls
Aller sur le répertoire data	cd d->

Lister le contenu en affichant le type des objets.	ls -F
Lister les détails du fichier Lot22perso.csv sans taper son nom.	ls Lot22 -> -> p ->
Afficher l'historique des dernières commandes	history

Exo 3 : Affichage du contenu des fichiers

Visualiser le contenu du fichier Lot22perso.csv.	cat Lot22perso.csv
Paginer le contenu du fichier Lot22perso.csv (page suivante, page précédente, ligne suivant, sortir)	more Lot22perso.csv (barre, b, retour, q)
Afficher les 5 premières lignes de tous les fichiers commençant par Lot22.	head -5 Lot22* more
créer un nouveau fichier sur votre répertoire de base contenant les 5 premières lignes de Lot11.csv et les 3 dernières lignes de Lot33.csv.	head -5 Lot11.csv > ~/temp1 tail -3 Lot33.csv > ~/temp2 cat ~/temp1 ~/temp2 > ~/result1 ou head -5 Lot11.csv > ~/temp1 tail -3 Lot33.csv >> ~/temp1
Stocker dans un fichier la liste des fichiers de votre répertoire de base.	cd ls -lsR * > listeFichier.txt

Exo 4 : Manipulation des fichiers et répertoires

Créer un répertoire temp sous votre répertoire de base.	(cd) mkdir temp
Dupliquer le fichier listeFichier.txt en listeFichier.txt_SV .	cp listeFichier.txt listeFichier.txt_SV
Renommer le fichier listeFichier.txt en tempfic.txt .	mv listeFichier.txt tempfic.txt

Effacer les fichiers temp que vous venez de créer.	ls -ls temp* rm temp *
Dupliquer le fichier listeFichier.txt_SV en tempfic.txt	cp listeFichier.txt_SV tempfic.txt
Effacer les fichiers temp que vous venez de créer.	rm -i temp* rm -Ri temp* % R pour le répertoire %
Copier CoursUnix sous votre répertoire de base en le renommant coursunix	(cd) cp -R /ugen/ugensyn/CoursUnix coursunix

Exo 5 : utilisation des droits

Voir les différents propriétaires et groupes dans /scratch	cd /scratch ls -lsF
Créer un répertoire prive contenant un fichier lisible que par vous	mkdir perso ls -lsd perso chmod g=--- perso
créer un script qui comprend les 2 lignes suivantes -> date -> banner salut	ncedit bonjour
exécuter le script bonjour	./bonjour
rendre le fichier exécutable	chmod u+x bonjour
exécuter le script bonjour à nouveau	./bonjour

Exo 6 : rechercher dans un fichier

Dans le répertoire coursunix/data, trouver les	grep -l « FR182 » *
--	----------------------------

fichiers contenant la chaîne « FR182 »	
Nombre d'occurrence de la chaîne FR182 dans les fichiers du répertoire.	grep -c « FR182 » *
afficher le nom du fichier + N° ligne et la ligne des fichiers contenant FR1823064011	grep -n « FR1823064011 » * more
Trouver les processus de ugensyn dans le résultat de la commande ps	ps -eaf grep ugenxxx

Les principales commandes UNIX (Formation bioinformatique MIG 2004/2005)

Gestion des répertoires

cd	Changer de répertoire courant
pwd	Afficher le répertoire courant (de travail)
ls	Lister le contenu d'un répertoire
mkdir	Créer un répertoire
rmdir	Détruire un répertoire vide

Gestion et manipulation des fichiers

cat	Afficher et concaténer des fichiers
touch	Créer un fichier vide
find	Rechercher un fichier
grep	Rechercher une chaîne de caractère
head	Afficher les n premières lignes/caractères
tail	Afficher les n dernières lignes/caractères
sort	Trier un fichier
diff	Comparer des fichiers ou des répertoires
tr	Transformer des chaînes de caractères
wc	Compter les lignes, mots ou caractères
file	Afficher le type du fichier (texte, postscript, binaire ...)
more	Afficher le contenu d'un fichier page par page

Gestion des fichiers et des répertoires

cp	Copier des fichiers ou répertoires
mv	Déplacer ou renommer des fichiers/répertoires
rsync -avz	Transférer des fichiers ou répertoires d'une machine à une autre
scp	Transférer des fichiers d'une machine à une autre
rm	Effacer des fichiers ou répertoires

Gestion des impressions

lpr	Imprimer
lpq	Afficher le statut des files d'attente
lprm	Supprimer une impression

Gestion des droits d'accès

chmod	Modifier les droits d'accès
ls	Lister les droits d'accès

Gestion des processus

ps	Afficher les processus
kill	Tuer un processus
nohup	Rendre indépendant un processus du terminal
Ctrl-Z	Suspendre un processus
bg	Relancer le processus stoppé et le placer en arrière plan.
fg	Ramène le job de l'arrière plan au premier plan
Ctrl-C	Terminer le processus en cours
&	Lancement en arrière plan
jobs	Afficher la liste des processus s'exécutant en arrière plan

Gestion de l'espace disque

df	Espace disque de chaque partition
du	Espace disque par répertoire
compress	Compresser un fichier en .Z
uncompress	Décompresser un fichier .Z
gzip	Compresser un fichier en .gz
gunzip	Décompresser un fichier .gz
unzip	Décompresser un fichier .zip

Un peu d'histoire

<http://fr.wikipedia.org/wiki/UNIX>

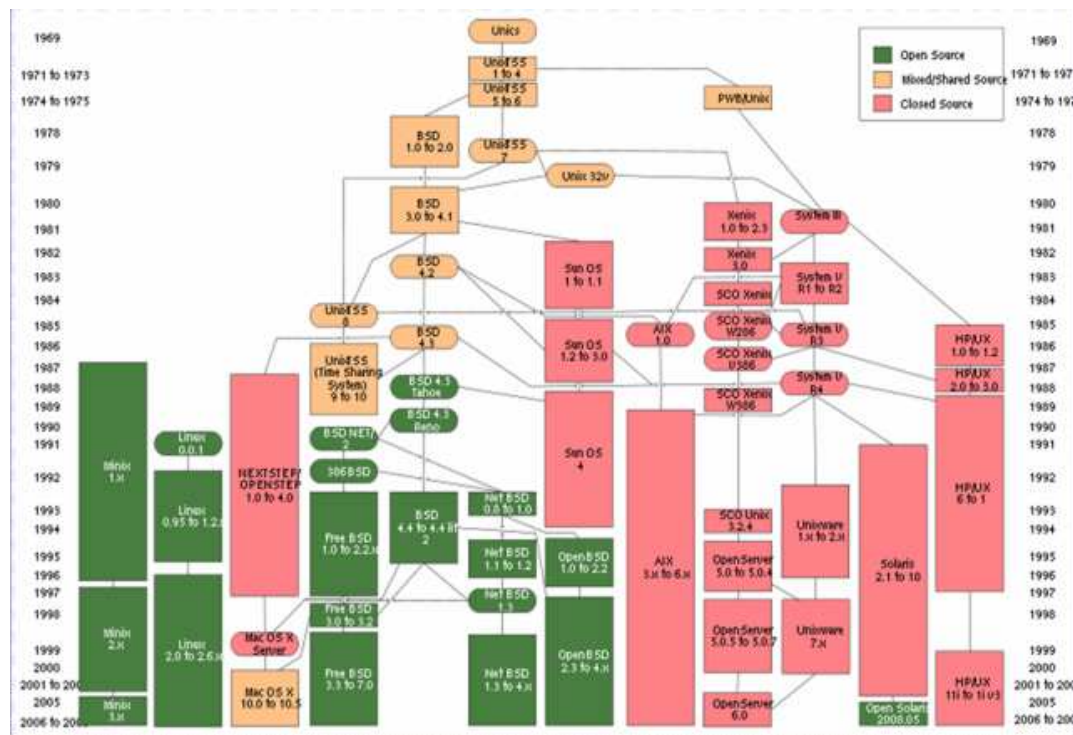
Les origines d'Unix remontent à la fin des années 1960 et à un laboratoire « Bell Labs » d'AT&T, la compagnie nationale de télécommunications américaine. Ce labo, en collaboration avec le MIT (Massachusetts Institute of Technology), développait un OS appelé « Multics ».

En 1969, Ken Thompson qui travaille alors pour les laboratoires Bell développa la première version d'un système d'exploitation mono-utilisateur. Le nom Unix dérive de Unics, un jeu de mot avec Multics, car contrairement à ce dernier qui visait à offrir simultanément plusieurs services à un ensemble d'utilisateurs, le système initial de Thompson se voulait moins ambitieux et utilisable par une seule personne à la fois avec des outils réalisant une seule tâche.

Un décret empêchait l'entreprise AT&T de commercialiser autre chose que des équipements téléphoniques ou télégraphiques. Bell Labs distribue donc son nouveau système d'exploitation sous forme de code source donnant à tout un chacun la possibilité de l'adapter et de développer librement des nouveautés, très vite apparaissent des familles différentes d'Unix.

On peut regrouper les premiers utilisateurs d'Unix en deux groupes :

- les universités et centres de recherches américains d'une part. L'université de Californie à Berkeley notamment est à l'origine de l'une des plus anciennes branches d'Unix, avec la *Berkeley Software Distribution* (BSD).
- les grands constructeurs informatiques d'autre part. Chacun de ces grands groupes a développé Unix selon ses propres besoins et intérêts, créant ainsi son propre Unix comme Solaris pour Sun Microsystems ou AIX pour IBM (Unix sur dga2). C'est l'autre grande branche d'Unix : *Unix System V* (lire « cinq »).



5-Principaux systèmes de type Unix

Dès 1973, presque tout Unix est réécrit dans le langage C (Dennis Ritchie). Ceci fait probablement d'Unix le premier système écrit dans un langage portable, c'est-à-dire autre chose que de l'assembleur qui est un langage très proche de la machine, compris directement par le processeur, et donc particulier à chaque type de machine. Il est ainsi possible de mettre en œuvre un système Unix sur la quasi-totalité des plates-formes matérielles.

Les constructeurs de stations et serveurs développent ce système d'exploitation pour pouvoir vendre leurs machines, en y ajoutant si possible un petit « plus » pour se démarquer de la concurrence. A la longue, les parcs Unix deviennent hétérogènes. Toute différence d'une machine à l'autre, menaçant l'interopérabilité et introduisant à terme une contre-productivité.

Plusieurs tentatives de normalisation ont été réalisées mais aucune n'a vraiment pris le dessus.

L'incompatibilité grandissante entre les nombreuses variantes d'Unix a fini par porter atteinte à sa popularité. C'est une des raisons pour lesquelles nombre de ces constructeurs proposent désormais le système GNU/Linux avec leurs serveurs. Toutefois, si le noyau Linux est bien défini, le système Linux change sensiblement d'une distribution à l'autre, ce qui conduit à des dissemblances causant parfois des pertes de temps.

De nos jours, les systèmes UNIX propriétaires, longtemps majoritaires dans l'industrie et l'éducation, sont de moins en moins utilisés. En revanche, trois systèmes de type UNIX basés sur BSD (FreeBSD, NetBSD et OpenBSD) d'une part, et le système GNU/Linux, compatible UNIX, d'autre part, ainsi que Mac OS X (basé sur Darwin), occupent une part de marché de plus en plus importante.

Logiciels libres et GNU/Linux

L'expression « Logiciel libre » fait référence à la liberté et non pas au prix. Pour comprendre le concept, vous devez penser à la « liberté d'expression », pas à « l'entrée libre ».

<http://www.gnu.org/gnu/gnu.fr.html> <http://fr.wikipedia.org/wiki/GNU>

Un logiciel libre est un logiciel dont la licence dite libre donne à chacun le droit de l'utiliser pour tous les usages, d'étudier son fonctionnement voire de l'adapter à ses besoins ou de l'améliorer (l'accès au code source est nécessaire), de le diffuser (copier, donner et vendre). Richard Stallman a formalisé la notion de logiciel libre dans la première moitié des années 1980 en créant la Free Software Foundation afin de promouvoir le projet GNU (GNU's Not Unix – Ensemble d'outils pour Unix : shell (Bash), éditeurs de texte (Emacs), compilateurs (GCC), système de fenêtrage, logiciel typographique (TeX) ...). Les logiciels libres constituent une alternative à ceux qui ne le sont pas, qualifiés de « propriétaires ».

Au début de l'informatique, le matériel de l'ordinateur constituait la source de revenus, le logiciel n'étant qu'un moyen d'en faciliter la vente, l'accès au code source était normal. Les constructeurs et les universitaires s'échangeaient volontiers logiciels et codes sources jusqu'à ce que les lois antitrust le leur interdisent afin de permettre l'exercice d'une concurrence dans le domaine du logiciel.

Parallèlement, les constructeurs ont restreint l'accès au code source des programmes, car les modifications souvent effectuées par les équipes des clients, rendaient problématique le soutien technique à distance.

Ainsi Bill Gates, en obtenant qu'IBM lui laisse les droits du DOS — puis de Windows — et ne commercialise qu'une licence d'usage, deviendra deux décennies plus tard l'homme le plus riche du monde

C'est pour lutter contre cette évolution allant à l'encontre de la pérennité du droit au partage que Richard Stallman a utilisé le droit d'auteur pour publier en 1989 La licence publique générale (GPL) notamment pour protéger GNU.

En 1991, l'étudiant finlandais Linus Torvalds, que la faible disponibilité du serveur Unix de l'université de Helsinki indispose et qui désire aussi comprendre le fonctionnement de son ordinateur basé sur un microprocesseur Intel 80386, entreprend d'écrire un noyau de système d'exploitation qu'on appellera plus tard « noyau Linux ».

L'association du projet GNU avec le noyau Linux se produit naturellement, l'arrivée du noyau Linux rend disponibles les logiciels du projet GNU sur les ordinateurs animés par des microprocesseurs de la famille Intel x86. Cela leur assura une large diffusion car ces projets étaient complémentaires. Depuis, des centaines de passionnés et des entreprises, petites ou géantes, sont venus participer au projet, dont Linus Torvalds est toujours le coordinateur.