

M2

« Astrophysique et instrumentations associées »

Cours UNIX

2006

Benoît Semelin

Les OS

OS = Operating System : Programme de base qui assure la gestion du matériel (clavier, écran, disque dur...), du système de fichier et des applications des utilisateurs.

Les différents OS:

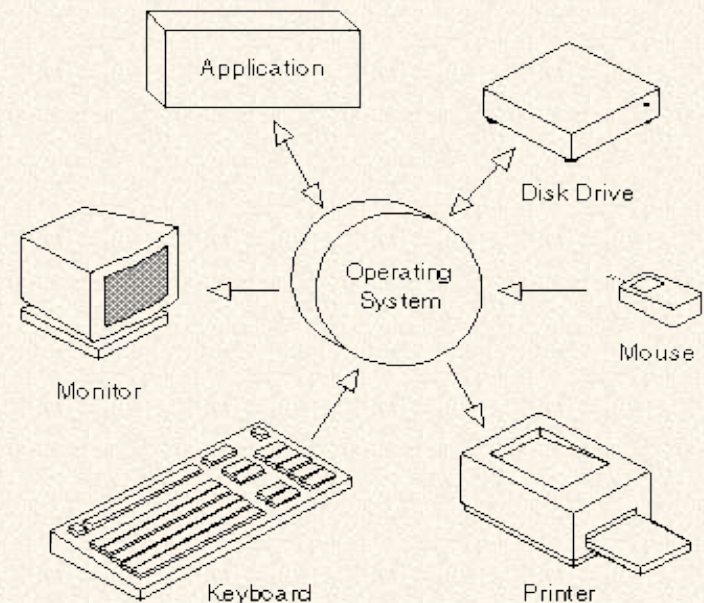
- Unix : Linux, BSD, Solaris, MacOS 10...
- Windows : NT, 2000, XP...
- MacOS < 10

Avantages d'Unix:

- Stable, sûr (peu de virus)
- Langage de commandes

Pourquoi ce cours?

Toutes les machines de calcul scientifique sont sous Unix.



Login



Il faut s'identifier pour avoir accès à un ordinateur sous Unix:

Login: dupont

Password: rx.73%\$ (changement avec la commande **passwd**)

Pourquoi?

- Système multi-utilisateurs: accéder à son environnement
- Sécurité: pas d'utilisateur non autorisé.
- Confidentialité: dupont ne lit le fichier de dubreuil que si dubreuil le décide (système de droits).

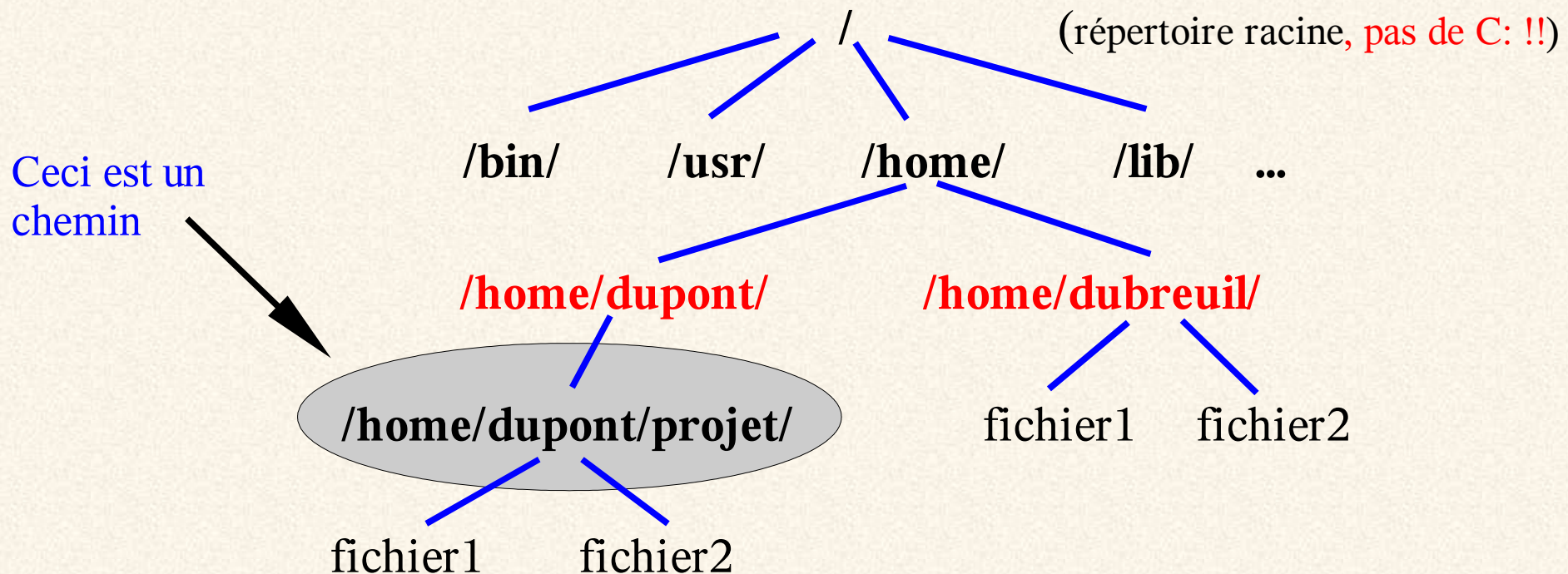
L'utilisateur « **root** »:

C'est l'administrateur. Certaines opérations ne peuvent être faites que pas lui. Il peut tout faire (danger !!).

Systeme de fichiers



Systeme de fichier de type arborescence:



Attention : /home/dupont/fichier \neq /home/dupont/Fichier \neq home/dupont/Fichier

A proscrire : Les espaces et caractères accentués dans les noms de fichiers et répertoires.

Montage NFS



NFS: Network File System.

Sur le domaine de l'IAP, votre **répertoire personnel** est disponible et **identique** sur **tous les ordinateurs**. On dit qu'ils sont montés par NFS.

Donc le /home/dupont est identiques sur toute les machines.

Par contre, le reste de l'arborescence (/bin, /usr) n'est pas forcément le même. Certaines commandes (les compilateurs) ne sont disponibles que sur certaines machines.

Avantages de NFS: Souplesse, commodité

Inconvénients de NFS: Instabilité en cas de lourde charge ?

L'interpréteur de commandes (ou Shell)



Le shell:

Après le login, dans une console, vous voyez (par exemple):

```
[dupont@machine]$ _
```

L'interpréteur de commande (ou Shell) attend vos instructions.

Le shell est un programme qui interprète vos commande et les transmet au noyau unix (OS). Ils permettent en plus de définir un environnement. Il existe plusieurs shells, les plus communs sont: **bash** et **tcsh**.

Définir/modifier un environnement de shell:

Votre environnement de shell est défini par des **variables**, par exemple: **PATH**

Cette variable définit la liste des répertoires où le Shell cherche le binaire correspondant à la commande que vous tapez (p.e.: **passwd**).

Pour connaître le contenu d'une variable:

```
[dupont@machine]$ echo $PATH
```

Modifier l'environnement du Shell



Modifier le contenu d'une variable:

Exemple, un nouveau compilateur a été installé dans `/opt/nanosoft/`. Votre shell ne le trouvera pas. Il faut modifier la variable **PATH**.

- Avec **bash** (catégorie des **sh**):

```
[dupont@machine]$ export PATH=$PATH:/opt/nanosoft
```

- Avec **tsh** (catégorie des **cs**):

```
[dupont@machine]$ setenv PATH $PATH:/opt/nanosoft
```

Ceci modifie **PATH** **uniquement pour cette console**.

Le shell, qui se lance automatiquement dès que vous ouvrez une console, lit, entre autres, les fichiers suivants: `/etc/profile`, `/etc/bashrc`, `/home/dupont/.bashrc` et `/home/dupont/.profile` (ou `.bash_profile`). En insérant les commandes ci dessus dans `.profile` ou `.bashrc`, la variable **PATH** sera **modifiée automatiquement à chaque fois** que vous ouvrirez une console.

La commande **env** donne la liste et le contenu de toutes les variables définies. Variables importantes: **SHELL, PATH, MANPATH, LD_LIBRARY_PATH**.

La commande man



La commande **man xxx** (pour manuel) permet d'afficher une documentation souvent très complète sur la commande **xxx**.

Essayez donc: **man man**

- La commande **man** est utile, en particulier, pour explorer les options possibles d'une commande.
- Flèches ↑ et ↓, barre d'espace et touche b pour faire défiler le manuel.
- En fin de manuel (parfois) une liste de commandes sur le même thème.
- La variable **MANPATH** définit dans quels répertoires **man** cherche les manuels. Il peut être nécessaire de la redéfinir, ou d'utiliser l'option -M de **man**.

La commande **info**, quand elle fonctionne, peut fournir une documentation encore plus précise.

Manipulations de répertoires



Parcourir l'arborescence des répertoires:

- Où suis-je?
pwd : affiche le chemin absolu pour le répertoire courant.
- Changer de répertoire:
cd rep1 : rentre dans le sous-répertoire rep1 du répertoire courant.
cd /rep1: tente d'entrer dans le répertoire de chemin absolu /rep1.
- Créer un répertoire:
mkdir rep1 : crée le répertoire rep1 comme sous répertoire du répertoire courant.

Symboles spéciaux dans les chemins:

- .. : désigne le répertoire parent du répertoire courant.
- ~ : désigne votre répertoire personnel, /home/dupont par exemple.
- . : désigne le répertoire courant.

Utilisation: **cd ..** , **cd ~** (identique à cd sans argument), **cd ~/rep1**, **cd -**

Manipulation de fichiers.



Commandes de base:

- Afficher la liste des fichiers: **ls**

ls : affiche la liste des fichiers et sous-répertoire du répertoire courant

ls rep1/toto : affiche la liste des fichiers et sous-répertoires du répertoire rep1/toto

ls -l : affiche une liste détaillée (droits, propriétaire, taille, etc...).

ls -a : affiche également les fichiers cachés.

ls -t : affiche par ordre de date de dernière modification

- Copier un fichier: **cp**

cp fich1 fich2 : copie le fichier fich1 dans le fichier fich2 du répertoire courant.

cp fich1 rep1/fich2: copie du répertoire courant vers un sous-répertoire.

cp -R rep1 rep2 : copie toute la arborescence de rep1 dans rep2.

-Déplacer, renommer un fichier: **mv**

mv fich1 fich2 : renomme fich1 en fich2.

mv fich1 ../fich2 : déplace en le renommant le fichier fich1 vers le répertoire parent.

Manipulation de fichiers (suite)



- Détruire un fichier: **rm**

rm fich1 : détruire le fichier fich1

rm repl/* : détruit tout les fichiers dans repl (voir sens de * ci-dessous)

rm -f repl/* : même chose, sans demande de confirmation.

rm -rf repl : détruit récursivement repl et ses sous répertoires.

Si root tape « **rm -rf /** » il détruit **tout** ce qu'il y a sur le disque....

Caractères spéciaux utiles:

? : remplace un caractère quelconque.

***** : remplace une chaîne de caractères quelconque.

Exemples d'utilisation:

rm repl/*.dat : détruit tout les fichiers du répertoire repl qui finissent par .dat

mv ../data/out0?.dat ~/poub/

Fichiers spéciaux: les liens symboliques



Les liens symboliques sont des fichiers spéciaux qui ne font que pointer vers un autre fichier ou répertoire. C'est l'équivalent des raccourcis sous windows.

Repérer un lien symbolique:

```
ls -l /usr/local/bin
```

```
-rwxr-xr-x 1 root root 189701 fév 12 2005 lamwipe*  
lrwxrwxrwx 1 root root 7 fév 12 2005 wipe -> lamwipe*
```

Ici, les commandes **wipe** et **lamwipe** sont identiques.

Créer un lien symbolique:

```
ln -s cible nom_du_lien
```

Cible peut être, par exemple, un chemin long à taper, dont on a souvent besoin.

Occupation du disque dur



Espace disque occupé par un répertoire:

du -ks rep1 : espace disque occupé par l'arborescence du répertoire rep1.

du -ks * : espace disque occupé par chaque sous répertoire du répertoire courant.

Liste des partitions et espace occupé:

df : liste des partitions montées, dont /home, avec espace occupé/libre.

Limite d'espace disque par utilisateur:

quota : *sur certains unix*, affiche le quota disque alloué à l'utilisateur (man quota).

Systeme de droits



Une situation courante:

```
[dupont@machine]$ cd /home/dubreuil
bash: cd: /home/dubreuil/: Permission denied
[dupont@machine]$ ls /home/dubreuil
ls: /home/dubreuil/: Permission denied
[dupont@machine]$ ls -l /home
drwx----- 10 dubreuil  groupe1 4096 mai 10 15:21 dubreuil/
drwxr-xr-x  8 dupont    groupe2 4096 mai 8  20:49 dupont/
```

Dupont **n'a pas les droits** pour faire ce qu'il veut, comme il s'en aperçoit grâce à ls -l (série de lettres et de - en début de ligne).

Les droits:

Il existe 3 types de droits, applicables à 3 classes d'utilisateurs.

- **r** : droit de **lire** dans un fichier/répertoire.
- **w** : droit d'**écrire** dans un fichier/répertoire.
- **x** : droit d'exécuter un fichier/répertoire.

Les caractères **2,3 et 4** s'appliquent au **propriétaire** du fichier/répertoire, les **5,6 et 7** s'appliquent au **groupe du propriétaire**, les **8,9 et 10** s'appliquent à **tout les autres utilisateurs**.

Changer les droits d'un fichier/répertoire.



Pour faire `cd /home/dubreuil`, dupont doit avoir le droit d'exécution (x) sur le répertoire.

Pour faire `ls /home/dubreuil`, il doit avoir le droit de lecture (r).

Comment changer les droits ?

Seul le propriétaire (et root) d'un fichier/répertoire peut en changer les droits.

La commande `chmod` permet de changer les droits:

`chmod o+x /home/dubreuil` : exécuté par dubreuil, donne le droit à tout le monde les droits d'exécution de son répertoire principal.

Trois symboles suivent **`chmod`**:

- 1^{er} symbole: **u** , change les droits du propriétaire
- g** , change les droits du groupe du propriétaire
- o** , change de tout les autres
- 2nd symbole: **+** , ajoute un droit
- , enlève un droit
- 3^{ième} symbole: **r,w** ou **x...**

Rechercher un fichier / dans un fichier



Localiser un fichier, une commande:

(s)locate toto.gif : recherche toto.gif sur toute l'arborescence.

find rep1 -name toto.gif : recherche, récursivement, les fichiers nommés toto.gif dans le repertoire et ses sous-répertoires. (man find)

which f90 : recherche l'exécutable f90 dans les répertoires de **PATH**

Recherche dans un (des) fichier(s):

grep non_de_variable func_*.c : Recherche la chaîne de caractère non_de_variable dans tout les fichiers « func_ *blabla.c* ».

diff fich1 fich2: Recherche, ligne à ligne les différences entre fich1 et fich2.

Éditer un fichier ASCII.



Examiner (rapidement) sans modifier:

more **fich1** : édite le contenu de fich1 (lecture uniquement). Parcours du fichier, recherche de chaîne

less **fich1** : comme **more**, en plus puissant.

tail -n 100 **fich1** : Affiche les 100 dernières lignes de fich1.

Éditer et modifier un fichier, éditeurs standards:

(x)emacs [**fich1**] : lance l'éditeur emacs. Le plus utilisé. Voir memo emacs.

vi(m) [**fich1**] : lance l'éditeur vi (ou vim). Présent sur tout les unix.

Compression de fichier.



Commandes de compression/décompression:

gzip (gunzip) fich1: compresse/décompresse **fich1**, et le renomme **fich1.gz**

bzip2 (bunzip2) fich1: compresse/décompresse **fich1**, et le renomme **fich1.bz2**

- * Un fichier compressé doit être décompressé avant de pouvoir être utilisé.
- * Le gain de taille peut varier de **90%** à **0%** de la taille initiale:
 - Types de fichiers déjà compressés: **.pdf**, **.jpeg**, **.mpeg**.
 - Types de fichiers non compressés: **.ps**, **.tiff**, **ASCII**.
- * Compresser un gros fichier (> 100 Mo) prend du temps, il ne faut pas le faire si le fichier est souvent utilisé.
- * Comprimez avant d'envoyer en pièce attachée ou de transférer par le réseau!

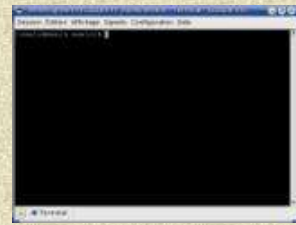
Faire une archive:

Il est utile, en vue de le compresser et de le transférer, de transformer toute l'arborescence d'un répertoire en un seul fichier.

tar cvf rep1.tar rep1 : crée l'archive **rep1.tar** à partir du répertoire **rep1**.

tar x(z)vf rep1.tar(.gz) : reconstitue le répertoire **rep1**.

Contrôle des « process »



Un process est un programme en cours d'exécution. Il lui est attribué un PID (process ID), un nombre qui le caractérise de manière unique.

Informations sur les process en cours:

ps -u : Affiche la liste de tout les process dont l'utilisateur est propriétaire, et des informations comme le PID, occupation mémoire, conso CPU, etc...

top : Liste en temps réel les process sur l'ordinateur, et ressources utilisées.

Modifier un process en cours:

Ctrl-C: interrompt le process en mode interactif.

kill -9 8564 : Tue le process de PID 8564.

renice 20 -p 8564 : Fixe à 20 la priorité du process de PID 8564.
(20 = priorité la plus faible)

Process en tâche de fond



Lancer un process en tâche de fond (et faire autre chose dans le shell):

gzip grosfichier & : & permet de lancer gzip en tâche de fond.

at 23:42 -f moncode.exe : lancera moncode.exe à 23h42 (en tâche de fond)

batch -f moncode.exe : lancera moncode.exe quand la charge de l'ordinateur le permettra (en tâche de fond).

at et **batch** fonctionnent même si l'utilisateur est délogué.

Mettre un process en suspend, puis le continuer en tâche de fond:

Contrôle-Z : Suspend l'exécution d'un process lancé en mode interactif (code, éditeur, navigateur, etc...)

jobs : Donne la liste des process suspendus.

fg %1: Reprend, en interactif, l'exécution du process suspendu numéro 1 (liste de jobs).

bg %1: Reprend, en tâche de fond, l'exécution du process suspendu numéro 1.

Opérateurs de redirection



Rediriger la sortie standard (sortie écran) dans un fichier: >

ls -l > list : écrit la liste des fichiers du répertoire courant dans list.

monprog.out > résultat : écrit la sortie standard de **monprog.out** dans le fichier résultat, au lieu de l'afficher à l'écran.

Utiliser un fichier comme entrée standard: <

monprog.out < data : Au lieu d'interroger l'utilisateur pour des valeurs de départ, monprog.out les lit dans le fichier data.

Transmettre la sortie d'une commande comme entrée à une autre: |

ls -l | grep jui : Afficher les lignes du résultat de ls -l, contenant « jui ».

ps -aux | more : Utilise more pour consulter la sortie de ps -aux.

ls | tee result.dat : la commande **tee** écrit l'entrée standard dans la sortie standard et dans un fichier...

Communiquer par le réseau



Se connecter à un autre ordinateur:

rlogin toto.iap.fr : se connecter à toto à l'IAP. (pas besoin de taper login et password)

telnet titi.obspm.fr : se connecter à titi.obspm.fr. (login + password)

ssh -l dupont tutu.parano.fr : se connecter sous dupont à tutu.parano.fr.
Connexion sécurisée.

Certains ordinateurs acceptent les connexions **telnet**, d'autres les connexions **ssh**, d'autres les deux. **ssh** est une commande puissante, beaucoup de fonctionnalités (man **ssh**).

Transférer un fichier par **ftp** (voir aussi **rcp**, **scp** et **rsync**):

ftp est un programme interactif basique de transfert de fichier.

ftp titi.obspm.fr : appelle la machine titi.obspm.fr (login + password)

- ◆ **cd**: change de répertoire sur titi
- ◆ **lcd**: change de répertoire sur la machine locale
- ◆ **put fich** : transfère fich de la machine locale vers titi (répertoires locaux)
- ◆ **get fich** : transfère fich de titi vers la machine locale.
- ◆ **bin** : transfère en mode binaire.
- ◆ **mget/mput fich***: transfère tout les fichiers commençant par fich..
- ◆ **prompt**: désactive/active la demande de confirmation.

Astuces pour aller plus vite !!



Complétion automatique:

- La touche **Tab** permet de compléter automatiquement une commande ou un nom de fichier.
- Si il y a plusieurs possibilités la liste des choix apparaît

Rappel de commande:

- Les touches **↑** et **↓** permettent de faire défiler la liste des commandes précédemment tapées (liste stockée dans `~/.history` ou `~/.bash_history`).

Répéter une ligne de commande:

- **!la répète** la dernière ligne de commande commençant par « la ».

Automatiser le travail: les scripts



sh et **csh** sont aussi des langages (basiques) de programmation. Les **scripts** sont de petits programmes contenus dans un fichier ascii exécutable, qui permettent d'automatiser de longues séries de commandes répétitives.

Exemple de script **sh**:

```
#!/bin/sh
for f in *45; do
if [ ! -f "$f/sft25.dat" ]
then
    rm $f/sft.lis
    rm $f/sft*.dat
fi
done
```

```
#!/bin/sh
for f in im*.ppm; do
cp $f im.temp
cat header.dat im.temp > $f
done
ppmtompeg pfile
```


Applications diverses

pine: Pour lire son e-mail dans une console texte.

mozilla: Pour naviger sur internet.

gnuplot: Tracer des courbes, des nuages de points, etc..

gimp: Manipulation d'images/film.

latex: Traitement de texte scientifique. (De nombreuses revues demandent un fichier latex)

gv/acroread: Visualiser des fichiers ps/pdf