Le b.a.ba d'UNIX

Nicolas Delestre, Michel Mainguenaud

23 septembre 2004

Qu'est ce qu'un SE...

- Un SE est le programme qui fait la liaison entre Le BIOS et les programmes utilisateurs
- Il s'occupe de :
  - la gestion des processus (programmes)
  - la gestion de la mémoire
  - la gestion des fichiers

Gestion des processus...

- Un SE est soit :
  - Mono-tâche : il ne peut faire tourner qu'un seul programme utilisateur à la fois
  - Multi-tâches : il peut faire tourner plusieurs programmes en même temps, mais comme il n' y a souvent qu'un seul micro-processeur, les programmes doivent se partager des temps d'exécution :
    - Multi-tâches coopératifs : chaque programme redonne régulièrement la main au SE
    - Multi-tâches préemptifs : le SE reprend régulièrement la main aux programmes

Gestion de la mémoire...

- Le SE :
  - vérifie que le programme en cours d'exécution :
    - accède à un espace mémoire existant
    - accède à un espace mémoire qui lui appartient
  - peut utiliser une partie de la mémoire physique pour enregistrer de la mémoire des programmes inactifs, et donc libérer la mémoire vive pour les programmes actifs : principe du `swap`

Gestion des fichiers...

- Le SE sait gérer un ou plusieurs formats d'enregistrement de données sur le disque (FAT, NTFS, Ext2, Ext3, Reiser-FS, .)
- Mais le SE présente toujours une structure logique des données sur le disque en présentant :
  - des volumes ou unités (par exemple A: et C: pour windows)
  - des répertoires (nommés aussi catalogue ou *directory*) qui contiennent :
    - des répertoires (on parle d'arborescence de répertoires)
    - des fichiers
  - des fichiers :
    - exécutables : ce sont des programmes, ils contiennent des instructions pour le microprocesseur
    - de données : des informations utilisées par le programme ou résultats de son exécution

Historique...

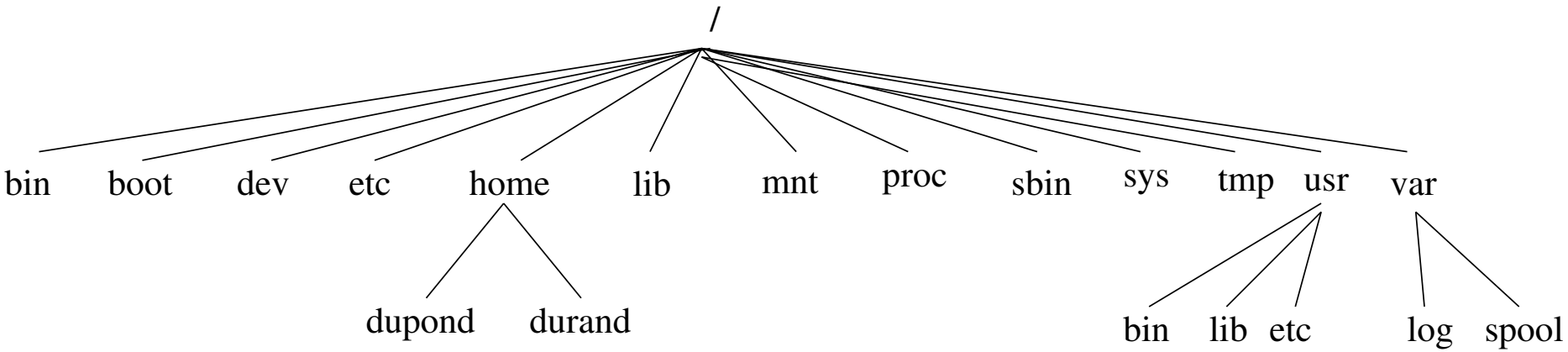
- Ken Thompson et Dennis Ritchie développent au début des années 70 un SE dans un langage évolué (le C) et non pas en assembleur
- De sa portablilité, plusieurs versions d'UNIX ont été développées
- Aujourd'hui il existe trois grandes familles d'UNIX :
  - systèmes BSD : NetBSD, FreeBSD, MacOS X
  - systèmes System V : HP/UX, AIX, IRIX
  - mixte : Sun Solaris, Linux
- Caractéristiques :
  - Multi-utilisateurs (droits d'accès aux fichiers)
  - Multi-tâches préemptifs
  - Tout est fichier (composants, processus, etc.)

Utilisateurs et groupe...

- Chaque utilisateur d'un système UNIX s'identifie à l'aide d'un identifiant (login) et un mot de passe (password)
- Chaque utilisateur appartient à un moins un groupe d'utilisateurs
  - Par exemple l'utilisateur "dupond" appartient au groupe "dupond" et peut appartenir aux groupes : *users*, *cdrom*, *floppy*, *audio*, *video*, etc.
- Il existe un compte particulier pour l'administrateur, le compte *root*.

Organisation du système de fichiers...

Vision extrêmement simplifiée :



Les fichiers...

- Chaque fichier appartient a un utilisateur
- À chaque fichier on définit des droits d'accès :
  - Pour l'utilisateur propriétaire
  - Pour un groupe d'utilisateurs
  - Pour les autres utilisateurs
- Ces droits d'accès sont les droits en :
  - lecture
  - écriture
  - exécution (pour les répertoires c'est le fait de pouvoir l'ouvrir)
- Les noms de fichier peuvent contenir pratiquement tous les caractères, mais il vaut mieux éviter les lettres accentuées et les espaces,
  - Attention : UNIX est sensible à la case

Les fichiers...

- Si le nom d'un fichier commence par "." alors c'est un fichier caché
- On référence un fichier soit de manière :
  - absolue en donnant le chemin intégrale d'accès à ce fichier depuis la racine (notée "/")
  - relative en donnant le chemin d'accès à ce fichier depuis l'endroit ou l'on se trouve. Les ".." permettent de référencer le répertoire supérieur
- On sépare le nom des répertoires utilisés à l'aide du caractère "/")
- Par exemple si on veut désigner le fichier cv.txt se trouvant dans le répertoire personnel de "dupont" en se trouvant dans le repertoire personnel de "durant"
  - Chemin absolu : `/home/dupont/cv.txt`
  - Chemin relatif : `../dupont/cv.txt`

Le shell...

- Par défaut un UNIX n'a pas d'interface graphique :
  - l'interface graphique est un programmme utilisateur comme un autre
  - on peut donc avoir plusieurs types d'interface graphique
- On interagie avec le système à l'aide d'un *shell* en tapant des commandes
- On peut souvent passer des paramètres à ces commandes (en séparant la commande et les paramètres par des espaces)
- On peut passer des options à une commande (ce sont des paramètres commençant par le caractère "-")
- Un shell est moins intuitif qu'une interface graphique mais beaucoup plus puissant

Les commandes de bases...

- `pwd`
  - permet de savoir où l'on se trouve dans l'arborecence
- `ls`
  - permet d'afficher la liste des fichiers présents dans un répertoire (si pas de répertoire de spécifié alors on affiche les fichiers du répertoire courant)
  - Options principales :
    - `-l` permet d'afficher les propriétés d'un fichier
    - `-a` permet d'afficher tous les fichiers (même les fichiers cachés)
- `cd repertoire`
  - permet de se positionner dans un répertoire
- `cp fichier1 fichier2`
  - permet de copier les données du fichier fichier1 dans le fichier fichier2

Les commandes de bases...

- `mkdir repertoire`
  - permet de créer un répertoire
- `rm fichier`
  - permet de supprimer un fichier
  - Options principales :
    - -f évite de confirmer la suppression
    - -r détruit un répertoire avec tout ce qu'il contient (répertoires et fichiers)
- `rmdir repertoire`
  - permet de supprimer un répertoire vide

Les commandes de bases...

- `mv fichierOuRepertoire destination`
  - déplace un fichier ou un répertoire dans un nouveau répertoire
- `chmod mode fichier`
  - permet de modifier les droits d'accès à un fichier ou un répertoire
  - `mode` est une succession de groupe de caractères séparés par une virgule :
    - `u`, `g` ou `o` pour indiquer que l'on va modifier le droit d'accès pour l'utilisateur, le groupe ou les autres
    - `+` ou `-` pour indiquer si on va ajouter un droit ou le supprimer
    - `r`, `w` ou `x` pour indiquer que l'on modifie le droit en lecture, écriture ou exécution
- `cat fichier`
  UNIXpermet d'afficher le contenu d'un fichier texte (ASCII)

Architecture des Systèmes d'Information

Les commandes de bases...

- `more fichier`
  - permet d'afficher le contenu d'un fichier texte avec demande de saut de page
- `chown utilisateur.groupe fichierOuRepertoire`
  - permet de changer le propriétaire et le groupe d'un fichier ou d'un répertoire
- `man commande`
  - Permet d'obtenir de l'aide sur une commande
- `passwd`
  - permet de changer le mot de passe
- `su - login`
  - permet de se connecter sous un nouvel utilisateur. Un nouveau shell est alors ouvert
- `exit`
  - permet de quitter le shell

Architecture des Systèmes d'Information

Expension des noms de fichiers...

- Lorsque l'on veut appliquer une commande à un ensemble de fichiers, on peut désigner cet ensemble de fichiers à l'aide d'une suite de caractères, que le shell va interpréter
- Par exemple :
  - imaginons que dans un répertoire nous ayons les fichiers suivants : fichier1.doc Fichier1.txt Fichier2.txt unFichier1.txt
  - comment afficher les propriétés de fichier1.doc et Fichier1.txt en une seule commande ?
  - il suffit de taper `ls -a [fF]*1*`, qui signifie afficher tous les fichiers fichiers qui commencent par un f ou un F, suivis de n'importe quels caractères, suivi du caractère 1, suivi de n'importe quels caractères

Expension des noms de fichiers...

Exemple[1]

```
bash$ ls -l
total 2
-rw-rw-r--    1 bozo   bozo          0 Aug  6 18:42 a.1
-rw-rw-r--    1 bozo   bozo          0 Aug  6 18:42 b.1
-rw-rw-r--    1 bozo   bozo          0 Aug  6 18:42 c.1
-rw-rw-r--    1 bozo   bozo        466 Aug  6 17:48 t2.sh
-rw-rw-r--    1 bozo   bozo        758 Jul 30 09:02 test1.txt

bash$ ls -l t?.sh
-rw-rw-r--    1 bozo   bozo        466 Aug  6 17:48 t2.sh

bash$ ls -l [ab]*
-rw-rw-r--    1 bozo   bozo          0 Aug  6 18:42 a.1
-rw-rw-r--    1 bozo   bozo          0 Aug  6 18:42 b.1

bash$ ls -l [a-c]*
-rw-rw-r--    1 bozo   bozo          0 Aug  6 18:42 a.1
-rw-rw-r--    1 bozo   bozo          0 Aug  6 18:42 b.1
-rw-rw-r--    1 bozo   bozo          0 Aug  6 18:42 c.1
```

[1] http://abs.ptithibou.org/abs-2.7-fr/globbingref.html

Expension des noms de fichiers...

```
bash$ ls -l [^ab]*
-rw-rw-r--    1 bozo   bozo          0 Aug  6 18:42 c.1
-rw-rw-r--    1 bozo   bozo        466 Aug  6 17:48 t2.sh
-rw-rw-r--    1 bozo   bozo        758 Jul 30 09:02 test1.txt

bash$ ls -l {b*,c*,*est*}
-rw-rw-r--    1 bozo   bozo          0 Aug  6 18:42 b.1
-rw-rw-r--    1 bozo   bozo          0 Aug  6 18:42 c.1
-rw-rw-r--    1 bozo   bozo        758 Jul 30 09:02 test1.txt
```

Lancer des programmes...

- On lance un programme depuis un shell comme une commande (certaines commandes du shell sont en fait des programmes, par exemple `ls`) :
  - on tape son nom et on appuie sur la touche Entrée (*Enter* ou *Return*)
- Attention :
  - Il peut y avoir plusieurs programmes qui portent le même nom, si on ne précise pas le chemin, le shell recherche le programme dans les chemins spécifiés dans la variable d'environnement PATH
- Lorsqu'on lance une commande, le shell devient inactif. Pour qu'il reste actif, il faut lancer le programme en "tâche de fond" (*background*) en suivant la commande du caractère "&"

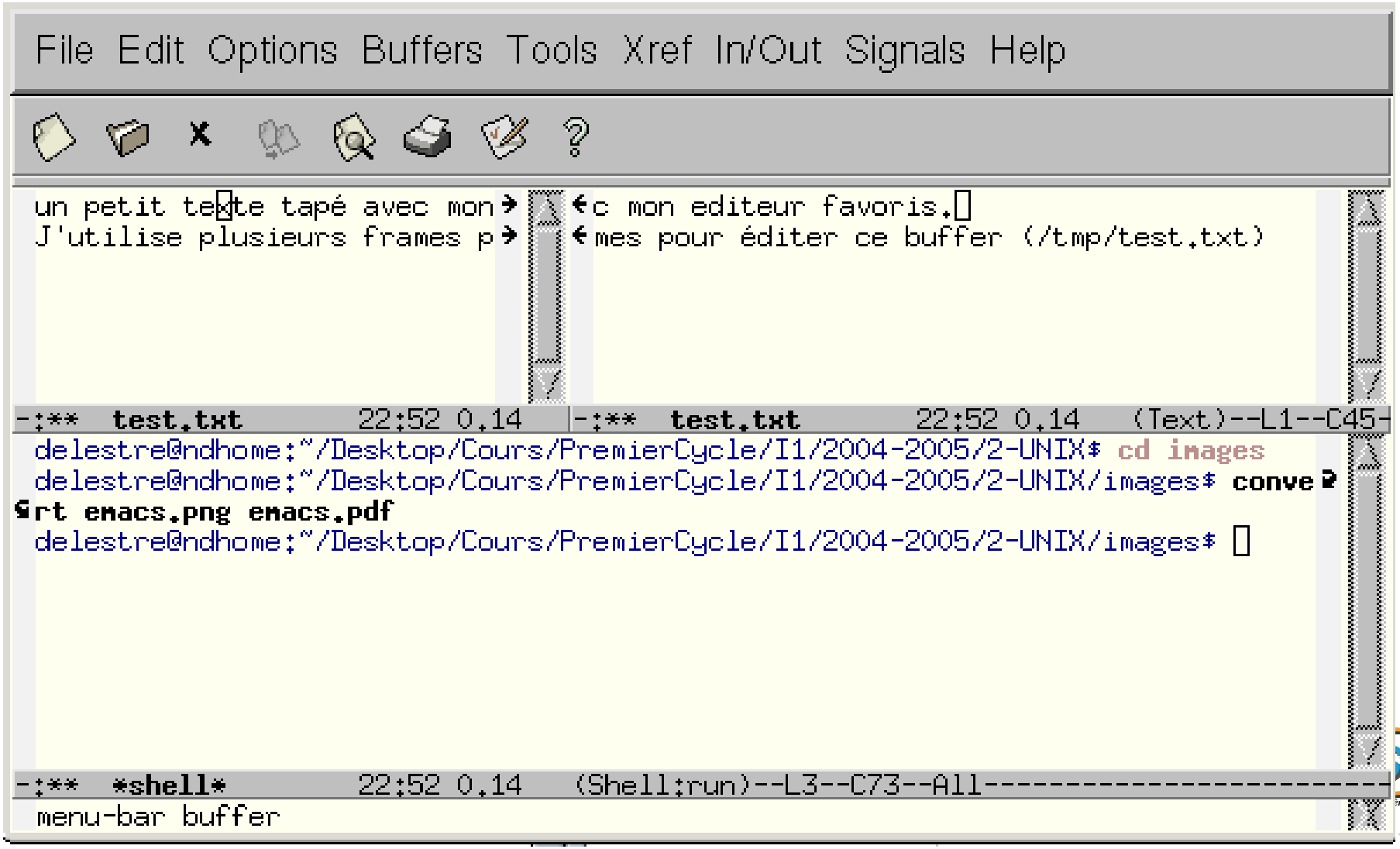
Emacs : Un éditeur pas comme les autres...

- Lorsque l'on veut éditer un texte, il faut utiliser un éditeur
- Emacs est un éditeur de texte "couteau suisse" :
  - très performant, ils s'adapte au type de document texte que vous tapez (une même action peut avoir des effets différents, par exemple mettre en commentaire une région)
  - nombreuses fonctionnalités
  - multi-plateforme
  - un peu compliqué à utiliser au début
- Il fonctionne en mode texte et en mode graphique
- Son fonctionnement est basé sur le concept de :
  - window
  - frame
  - buffer

Buffer, frame et window...

- Un `buffer` est l'entité que vous êtes en train d'éditer, d'utiliser :
  - Très souvent c'est un fichier, mais ca peut être aussi un shell, une page de documentation, une aide temporaire de l'éditeur, etc.
- Une `frame` est le visualisation de tout ou partie d'un buffer
  - On peut utiliser simultanément plusieurs frames d'un buffer
- Une `window` est une fenêtre de votre interface graphique qui peut contenir plusieurs frames

Buffer, frame et window...



Raccourcis de base...

- Pour les fichiers
  - C-x C-f : Ouvrir/Créer un fichier
  - C-x C-s : Sauvegarder un fichier
  - C-x C-c : Quitter Emacs
- Pour les déplacements
  - C-v : Pagedown
  - M-v : Pageup
  - C-a : Début de ligne
  - C-e : Fin de ligne
  - M-a : Début de phrase
  - M-e : Fin de phrase
  - M-< : Fin de fichier
  - M-> : Début de fichier

Raccourcis de base...

- Pour l'édition :
  - C-x u : Annulation (C-_)
  - C-espace: Marquer (début d'un copier/couper)
  - C-y : Coller
  - C-w : Couper
  - M-w : Coller
  - C-s : rechercher
  - M-% : remplacer
- Divers (mais TRÈS important) :
  - C-g : Annulation de l'action courante
  - M-x nomCommande : permet d'exécuter une commande