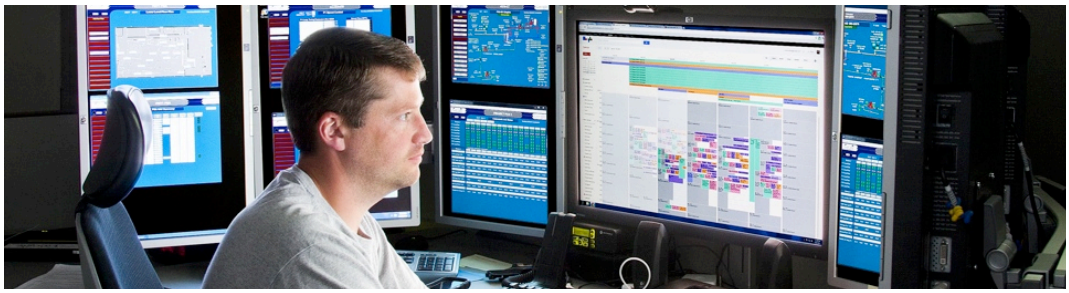
Administration Système — Les bases de la ligne de commande Unix



Année académique 2014/15

(C) 2015 Marcel Graf

Les bases d'Unix

- L'histoire d'Unix
- Le shell
- La ligne de commande
- Obtenir de l'aide
- Le système de fichiers

(C) 2015 Marcel Graf

Le système d'exploitation Unix
Origines

- Le système d'exploitation Unix a une longue histoire mais reste moderne
  - 1964 : Le MIT, AT&T Bell Labs et General Electric commencent à développer un nouveau système d'exploitation ambitieux, MULTICS, pour le mainframe GE-645.
  - AT&T, frustré par la taille et complexité de MULTICS, se retire du projet. Ken Thompson et Dennis Ritchie utilisent leurs expériences avec MULTICS pour développer un système d'exploitation à échelle réduite qu'ils appellent UNICS
    - PDP-7 : CPU 18 bit 570 kHz, 9 kB RAM standard, extensible jusqu'à 144 kB, lecteur de bande à accès aléatoire
  - 1972 : Unix est re-écrit en langage C, ce qui le rend portable sur d'autres plateformes
  - Décades 1970 et 1980 : Bell Labs produit plusieurs versions d'Unix et partage le code source avec des universités. L'influence d'Unix dans les cercles académiques mène à l'adoption commerciale d'Unix : Sequent, HP-UX, Solaris, AIX, Xenix, ...
  - Décade 1990 : Les systèmes Unix augmentent en popularité. Création du mouvement Open Source Un réseau mondial de programmeurs développe Linux et les distributions BSD.
  - Décade 2000 : Apple adopte Darwin pour Mac OS X. Les deux principales plateformes de l'ère post-PC, iOS et Android, sont des descendants d'Unix.

(C) 2015 Marcel Graf

L'histoire d'Unix
Ken Thompson and Dennis Ritchie

- Inventeurs du système d'exploitation Unix à AT&T Bell Labs
- Dennis Ritchie inventa aussi le langage de programmation C

(C) 2015 Marcel Graf

L'histoire d'Unix

De 1969 à aujourd'hui

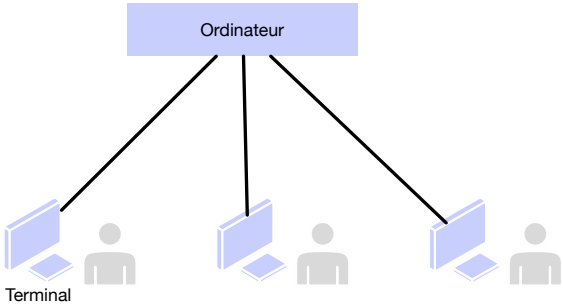
source: Wikipedia

(C) 2015 Marcel Graf

Le système d'exploitation Unix

Un système d'exploitation multi-tâches et multi-utilisateur

- Dans les années 60 apparaissent les systèmes multi-utilisateurs basés sur le principe du temps partagé *(time sharing).*
  - Exécution de plusieurs tâches en même temps (pseudo-parallélisme)
  - Support de plusieurs utilisateurs
- Les utilisateurs travaillent sur des terminaux qui sont connectés (souvent à travers de lignes téléphoniques) à l'ordinateur.

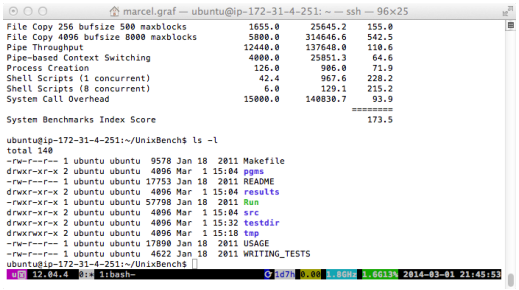


(C) 2015 Marcel Graf

Le shell

Une interface utilisateur pour le système d'exploitation

- Orienté texte (ligne de commande, *Command-Line Interface*, *CLI*)
  - Par exemple bash, cmd, PowerShell
  - A deux fonctions :
    - Ligne de commande
    - Environnement de programmation (scripting)
- Graphique (*Graphical User Interface*, *GUI*)
  - Par exemple Gnome, KDE, Windows Shell
  - Les shells graphiques sont aussi appelés *environnement de bureau*



(C) 2015 Marcel Graf

Le shell

La ligne de commande

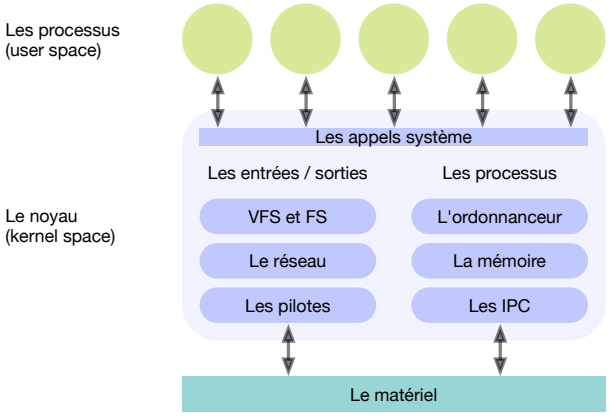
- La ligne de commande (CLI)
  - Une interface utilisateur qui reçoit des commandes tapées par l'utilisateur
  - Tous les systèmes d'exploitation importants modernes ont une ligne de commande
- Pourquoi la ligne de commande ?
  - Donne accès à tous les aspects du système d'exploitation
    - On ne peut pas tout faire avec les interfaces graphiques
  - Il est plus facile de faire des manipulations 100% reproductibles et on peut convertir les commandes facilement en scripts
    → Automatisation
  - Quand on la maîtrise, la ligne de commande est plus rapide et plus puissante
  - L'accès à distance et l'accès à plusieurs machines en même temps est plus aisé

```
knuth login: aeb
Password: ********
$ date
mardi 6 août 2002, 23:50:44 (UTC+0200)
$ cal
    août 2002
lu ma me je ve sa di
          1  2  3  4
 5  6  7  8  9 10 11
12 13 14 15 16 17 18
19 20 21 22 23 24 25
26 27 28 29 30 31
$ ls
bin  tel
$ ls -l
total 2
drwxrwxr-x   2 aeb       1024 6 août 23:51 bin
-rw-rw-r--   1 aeb         48 6 août 23:52 tel
$ cat tel
marie    05-11-22-33-44
pierre   03-22-33-44-55
$ cp tel tel2
$ mv tel tel1
$ ls -l
total 3
drwxr-xr-x   2 aeb       1024 6 août 23:51 bin
-rw-r--r--   1 aeb         48 6 août 23:52 tel1
-rw-r--r--   1 aeb         48 6 août 23:53 tel2
$ diff tel1 tel2
$ rm tel1
$ grep marie tel2
marie    05-11-22-33-44
$
```


(C) 2015 Marcel Graf

Le shell
Relation avec le noyau

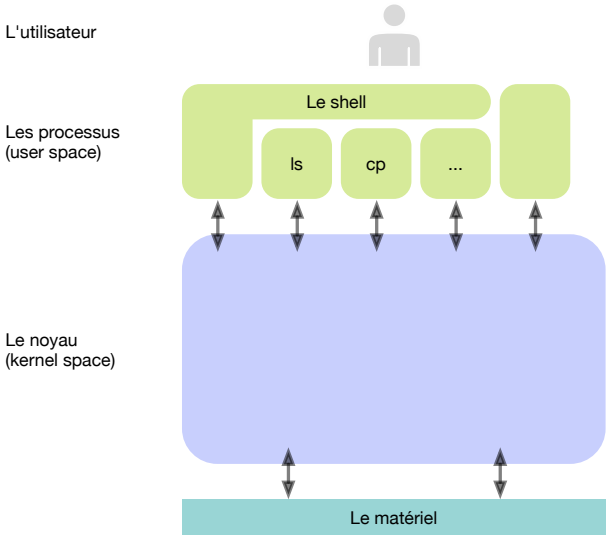
- Rappel : Relation entre les processus utilisateur, le noyau et le matériel



(C) 2015 Marcel Graf

Le shell
Relation avec le noyau

- Le shell est un processus utilisateur comme les autres.
  - Il ne fait pas partie du noyau
  - On peut facilement changer de shell
- Fréquemment le shell lance d'autres processus utilisateur qui peuvent rester connectés à lui ou devenir indépendants.

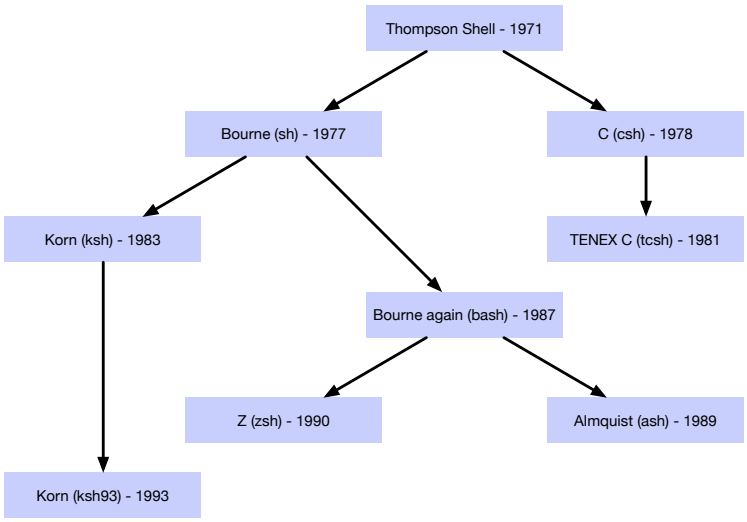


(C) 2015 Marcel Graf

Le shell

Historique des shells

- Ken Thompson écrit le premier shell en 1971 et le nomme *sh* (ce shell n'existe plus)
- De cette origine se sont développés deux branches principales
  - *Bourne shell* et descendants
  - *C shell* et descendants
- Aujourd'hui *bash* est le shell le plus répandu



(C) 2015 Marcel Graf

Le shell

Le shell **bash**

- Le *Bourne Again shell* ou **bash** est un descendant du *Bourne shell (sh).*
  - Le Bourne shell est nommé d'après son auteur principal, Stephen R. Bourne.
  - Bash a été écrit principalement par Brian Fox, et plus tard, Chet Ramey.
  - Incorpore des fonctionnalités du Korn shell (ksh) et C shell (csh).
  - La plupart des scripts sh tournent sous Bash sans modification.
- Logiciel libre, fait partie du projet GNU.
- C'est le shell de référence de Linux et de Mac OS X.
  - C'était la première application portée par Linus Torvalds pour tourner sur son nouveau noyau Linux.
  - Disponible pour presque toutes les version d'Unix et Windows.

(C) 2015 Marcel Graf

La ligne de commande

Déroulement d'une session

- Dans un système Unix, tout utilisateur doit être déclaré, c.-à-d. enregistré, avec un nom d'utilisateur *(user id).*
- À partir de son terminal l'utilisateur établit une connexion avec le système Unix.
- Le système envoie une invite de login.
- L'utilisateur donne son nom d'utilisateur et un mot de passe : c'est le *log-in.*
- Le système Unix établit une session et lance un shell pour l'utilisateur.
- Tant que la session n'est pas terminée
  - Le shell envoie une invite de commande *(prompt)* au terminal, par défaut c'est le caractère $, et attend la commande de l'utilisateur.
  - L'utilisateur saisit la commande et frappe sur la touche Entrée.
  - Le shell interprète et exécute la commande et affiche les résultats.
- La commande **`exit`** termine le shell, la session et la connexion.

(C) 2015 Marcel Graf

La ligne de commande

Quelques premières commandes

- Afficher le nom et les groupes de l'utilisateur courant : **`id`**
- Afficher le nom de la machine : **`hostname`**
- Afficher le nom du système d'exploitation : **`uname`**
  - Afficher toute l'information : **`uname -a`**
- Afficher les utilisateurs connectés sur la même machine : **`who`**
- Terminer la session : **`exit`**

(C) 2015 Marcel Graf

La ligne de commande

Obtenir de l'aide — L'aide propre aux commandes

- La plupart des commandes affichent une aide synthétique quand on les appelle avec le paramètre `--help`.

```
$ ls --help
Usage: ls [OPTION]... [FILE]...
List information about the FILEs (the current directory by default).
Sort entries alphabetically if none of -cftuvSUX nor --sort is specified.

Mandatory arguments to long options are mandatory for short options too.
  -a, --all                  do not ignore entries starting with .
  -A, --almost-all           do not list implied . and ..
      --author               with -l, print the author of each file
```

- Si le texte affiché est trop long, ajouter `| less` pour obtenir une pagination :
  - `$ ls --help | less`

- Il peut arriver qu'une commande ne comprenne pas le paramètre `--help`, dans ce cas vous obtiendrez un message d'erreur (paramètre non valide).

(C) 2015 Marcel Graf

La ligne de commande

Obtenir de l'aide — L'aide interne au shell

- Le shell met à disposition un nombre de commandes internes (cd, echo, pwd, if, for, ...) pour lesquels on peut obtenir de l'aide en tapant `help` suivi de la commande.

```
$ help pwd
pwd: pwd [-LP]
    Print the name of the current working directory.

    Options:
      -L print the value of $PWD if it names the current working
     directory
      -P print the physical directory, without any symbolic links

    By default, `pwd' behaves as if `-L' were specified.

    Exit Status:
    Returns 0 unless an invalid option is given or the current directory
    cannot be read.
```


(C) 2015 Marcel Graf

La ligne de commande

Obtenir de l'aide — Le manuel en ligne

- Tous les systèmes Unix ont une aide intégrée, les "*man pages*" nommés après la commande `man`.
  - Elle peut paraître minimaliste et obscure, mais elle est généralement complète.
- Tapez `man` suivi de la commande.

- Pour afficher le manuel en français précédez man par une affectation de la variable LC_ALL comme ceci :
  - `LC_ALL=fr_FR.UTF-8 man ls`

```
LS(1)                          User Commands                          LS(1)

 NAME

       ls - list directory contents

 SYNOPSIS

       ls [OPTION]... [FILE]...

 DESCRIPTION

       List information about the FILEs (the current directory by default).
       Sort entries alphabetically if none of -cftuvSUX nor --sort is
       specified.

       Mandatory arguments to long options are mandatory for short options
       too.

       -a, --all
              do not ignore entries starting with .

       -A, --almost-all
              do not list implied . and ..

       --author
              with -l, print the author of each file

       -b, --escape
              print C-style escapes for nongraphic characters

       --block-size=SIZE
              scale sizes by SIZE before printing them; e.g.,
              '--block-size=M' prints sizes in units of 1,048,576 bytes; see
              SIZE format below

       -B, --ignore-backups
              do not list implied entries ending with ~

       -c     with -lt: sort by, and show, ctime (time of last modification
              of file status information); with -l: show ctime and sort by
              name; otherwise: sort by ctime, newest first

       -C     list entries by columns

       --color[=WHEN]
```


(C) 2015 Marcel Graf

La ligne de commande

Obtenir de l'aide — Le manuel en ligne

- Le manuel est divisé en sections. Il contient bien plus que seulement les commandes utilisateur.
- Il arrive parfois que l'appel au manuel retourne une page dans une section qu'on ne voulait pas.
  - `man` recherche par défaut la première occurrence dans l'ordre des sections.
  - Dans ce cas indiquer le numéro de section juste avant le mot : `man 5 passwd`

Section	Contenu
1	Commandes utilisateur et applications
2	Appels système (API du noyau)
3	Appels des bibliothèques
4	Fichiers spéciaux (contenu de /dev comme sd, hd, pts, etc.)
5	Format des fichiers (/etc/passwd, /etc/hosts, etc.)
6	Les jeux, économiseurs d'écran, gadgets, etc.
7	Divers, commandes non standard
8	Commandes d'administration du système Linux
9	Sous-programmes du noyau (souvent vide)

(C) 2015 Marcel Graf

La ligne de commande

Obtenir de l'aide — Le manuel en ligne

- Si vous avez un doute sur la commande à utiliser, ou que vous ayez perdu son nom, ou encore que vous vouliez connaître toutes les commandes liées à un mot, alors utilisez le paramètre `-k` de `man` :

```
$ man -k passwd
chgpasswd (8)        - update group passwords in batch mode
chpasswd (8)         - update passwords in batch mode
Crypt::PasswdMD5 (3pm) - Provides interoperable MD5-based crypt() functions
fgetpwent_r (3)      - get passwd file entry reentrantly
getpwent_r (3)       - get passwd file entry reentrantly
gpasswd (1)          - administer /etc/group and /etc/gshadow
grub-mkpasswd-pbkdf2 (1) - generate hashed password for GRUB
lppasswd (1)         - add, change, or delete digest passwords.
mkpasswd (1)         - Overfeatured front end to crypt(3)
pam_localuser (8)    - require users to be listed in /etc/passwd
passwd (1)           - change user password
passwd (1ssl)        - compute password hashes
passwd (5)           - the password file
passwd2des (3)       - RFS password encryption
...
```

- On peut aussi utiliser la commande `apropos` qui est un synonyme pour `man -k`.

(C) 2015 Marcel Graf

La ligne de commande

Touches spéciales

- Quelques touches spéciales sont à connaître :
  - Ctrl-C : Termine l'exécution du programme en cours.
  - Ctrl-Z : Stoppe l'exécution du programme en cours. Le programme peut être repris avec **`fg`** ou **`bg`**.
  - Ctrl-D : (Quand un programme reçoit son flux d'entrée du clavier) Terminaison du flux d'entrée

(C) 2015 Marcel Graf

La ligne de commande

Rappel de commandes précédentes et édition

- Il est très utile de rappeler des commandes déjà exécutées
  - Les touches ↑ et ↓ permettent de naviguer l'historique des commandes
  - On peut éditer la ligne si nécessaire avec ←, →, **Ret. Arr.**, **Suppr**.
  - Et l'exécuter en tapant **Entrée**
- Une fonction très puissante du shell bash est la recherche interactive dans l'historique de commandes
  - Taper **Ctrl-R** pour commencer la recherche interactive.
  - Taper les premiers caractères d'une partie quelconque de la commande qu'on cherche.
  - Le shell va en arrière dans l'historique pour trouver une commande qui correspond.
  - Après chaque caractère l'invite de commande affiche la première commande trouvée.
  - On peut continuer à taper des caractères pour affiner la recherche,
  - ou répéter **Ctrl-R** pour remonter plus loin dans l'historique.
  - Si on a trouvé la commande désirée, commencer à l'éditer ou taper **Entrée**.
  - Pour abandonner la recherche taper **Ctrl-G**.
- La commande **`history`** affiche une liste avec l'historique des commandes

(C) 2015 Marcel Graf

La ligne de commande

Que faire si le terminal ne réagit plus

- Au cas où la session se plante et le terminal ne répond pas à une commande, essayer les manipulations suivantes dans l'ordre jusqu'à obtenir une réaction
  - Taper **Entrée** une fois
    Possiblement vous avez tapé du texte mais pas encore la touche **Entrée** pour passer la commande au shell.
  - Taper **Ctrl-Z**
    Un programme qui continue de tourner sera stoppé. Trouver le nom du programme avec **`jobs`**, le continuer avec **`fg`** ou le tuer avec **`kill`**.
  - Taper **Ctrl-C**
    Un programme qui continue de tourner sera terminé.
  - Taper **Ctrl-Q**
    Si l'affichage du terminal a été bloqué (accidentellement) par **Ctrl-S** le terminal sera débloqué.
  - Vérifier la connexion réseau
  - Taper **Ctrl-D** au début d'une nouvelle ligne
    Si un programme est en train d'attendre une entrée utilisateur il lui sera signalé que la saisie est terminée.
  - Fermer la fenêtre terminal

(C) 2015 Marcel Graf