

Systemes d'exploitation

Buts de cet enseignement :

Apprentissage du système Unix (Linux debian)

- Utilisation
- Programmation
- Administration



Pierre Nerzic

- Enseignant à l'IUT dept INFO
- Chercheur équipe Shaman Irisa/Enssat

Définition système d'exploitation

- **Systeme d'exploitation = le logiciel qui**
 - fait fonctionner un ordinateur
 - permet de l'utiliser.
- **Operating System (OS) en anglais :**
 - Il unifie le fonctionnement des composants quelque soient leurs modèles
 - Il contient de nombreuses fonctions permettant de programmer des applications (API)
 - Il offre une vue simple et dynamique aux utilisateurs (bureau, accès au réseau...)

Exemples de systèmes

- Micro-ordinateurs loisirs ou bureautique :
 - Ordinateurs type PC :
 - Microsoft Windows (98, XP, 7, 8, 10...)
 - Linux (une des très nombreuses variantes de Unix)
 - Ordinateurs Apple (Macintosh) :
 - Mac OS : basé sur Linux
 - Autres : tablettes, liseuses, smartphones
 - iOS, Android (basé sur Linux) ou Linux directement
- Serveurs professionnels (banques, labos...)
 - IBM, Sun, Hewlett-Packard... : Unix

Buts de ce cours

- **Savoir utiliser un système d'exploitation professionnel** : Unix (version Linux Debian) et indirectement avoir des notions sur les autres systèmes utilisés dans l'industrie,
- **Savoir ajuster les paramètres personnels,**
- **Savoir programmer de petits traitements (scripts)**
- **Connaître les bases de l'administration système**

Calendrier

- Période P1 (5 semaines) : les bases
 - 2h CM/semaine, 1h TD/sem, 2h TP/sem
- Période P2 (5 semaines) : la programmation
 - 1h CM/sem, 1h TP/semaine
- Période P3 (6 semaines) : l'administration
 - 1h CM/semaine, 1h TD/sem, 2h TP/sem

Contrôle des connaissances

- Devoir surveillé (sur table, en salle de DS)
 - 2 devoirs surveillés (courant P2 et fin P3)
- Travaux pratiques :
 - 5 tests en début de séance, 4 questions QCM
 - Un TP noté en séance (1h pour effectuer les manipulations demandées)
 - Un WIKI à remplir avec les compte-rendus de vos manipulations

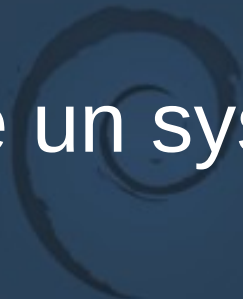
<http://perso.univ-rennes1.fr/pierre.nerzic/SYS1A>

Chapitre 1 : Notions de base

machines
comptes, connexion
commandes
fichiers

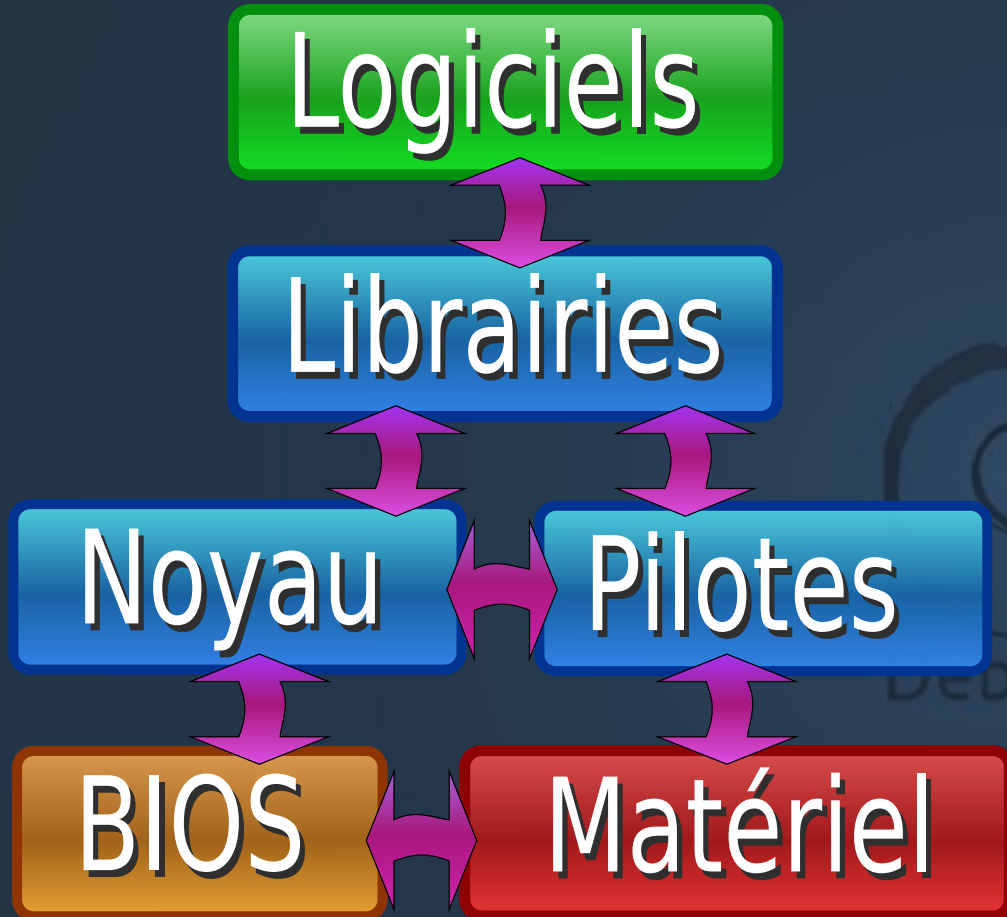
1.1 – Contenu d'un système

De quoi est composé un système d'exploitation ?

The Debian logo, which is a stylized spiral, is positioned behind the text 'Debian'.

Debian

Que contient un système ?



Les programmes utilisent des librairies basées sur le noyau et les pilotes matériels.

Le noyau gère le processeur, la mémoire, les disques, etc. à l'aide des pilotes et du BIOS.

Les composants d'un système

- Quelques détails de plus :
 - **BIOS et pilotes** gestion du matériel
 - **noyau et bibliothèques** pour exécuter les logiciels
 - **services et démons** communiquer, imprimer...
 - **interface utilisateur** pour lancer les logiciels
 - **logiciels et données** pour travailler

Que fait un système ?

- **Le système d'exploitation est lancé au démarrage de l'ordinateur (*boot*) par le BIOS**
 - il initialise (teste et met en route) les éléments : mémoire, disques, carte graphique...
 - il vérifie les données : état des fichiers
 - il active les services : réseau, imprimante...
- **Puis le système sert d'intermédiaire entre l'utilisateur et la machine : il attend les ordres et les exécute**

Différents types de systèmes

- **Mono/multitâche :**

- peut-il exécuter plusieurs programmes en même temps ?

Ex : PC avec Windows, Linux, Mac OS : oui

Ex : PC avec MS-DOS : non

Ex : calculatrice programmable : non

- **Système réparti ou non ?**

- L'exécution d'un (gros) programme peut-il se faire simultanément sur plusieurs machines reliées ensemble ?

Ex : « ferme de calcul », mais de manière transparente

- Les données d'un utilisateur sont-elles stockées/disponibles simultanément sur plusieurs machines (ex : cloud)

Différents types, suite

- **Mono/multiutilisateur :**
 - peut-il gérer les **données** de plusieurs **utilisateurs** en les protégeant les uns des autres ?
 - Ex : Windows > 98 : oui
 - Ex : MS-DOS : non
 - peut-il gérer le **travail simultané** de plusieurs utilisateurs, connectés par le réseau ?
 - Ex : Windows : non (bureau à distance = super bof)
 - Ex : Unix : oui (aucune limite, sauf celle de la machine)

1.2 - Comptes

Pour distinguer les utilisateurs les uns des autres



Comptes

- Pour distinguer les utilisateurs d'un même système, on leur associe un « compte ».
 - Nom de compte (*login*, *logname*)
 - Mot de passe (*password*) bien choisi [attention pirates]
- Séance de travail :
 - 1) l'ordinateur est verrouillé
 - 2) connexion (*to log in* ou *to log on*)
 - 3) l'ordinateur est disponible : on peut travailler
 - 4) déconnexion (*logout*, *logoff*) ou extinction

Comptes à l'IUT

- Chaque étudiant possède plusieurs comptes :
 - compte Windows (bureautique, comptabilité...)
 - compte Unix (cours systèmes, algo, réseaux...)
 - compte PostgreSQL (cours de bases de données)
 - autres comptes sur des outils (trac, svn...)
 - compte ENT Rennes (courrier, EdT, notes...)
- Même login, même mot de passe partout (sauf ENT)

1.3 - Serveurs

Serveur = (par abus de langage) ordinateur qui :

- contient les comptes
- effectue des calculs scientifiques
- contient un site internet
- gère une imprimante
- etc.

Debian

En fait, c'est un programme sur un ordinateur qui transforme l'ordinateur en serveur. Un même ordinateur peut contenir plusieurs serveurs : http, ftp, imap, lpr, nfs...

PC au département INFO

Salle Machine



Salle de TP



Réseau

- Les PC des salles de TP sont reliés aux serveurs par le réseau

Serveurs à l'IUT

- A l'IUT il y a beaucoup de **serveurs** :
 - des serveurs de **données** : stockent des fichiers
 - des serveurs de **calcul** : bases de donnée Oracle et machines pour faire les TP de système et d'algo
 - des serveurs d'**impression** reliés aux imprimantes
 - des serveurs d'**authentification** : comptes et mdp
 - des serveurs **internet** : site internet de l'IUT, sites étudiants...
 - autres serveurs : archivage, courrier...

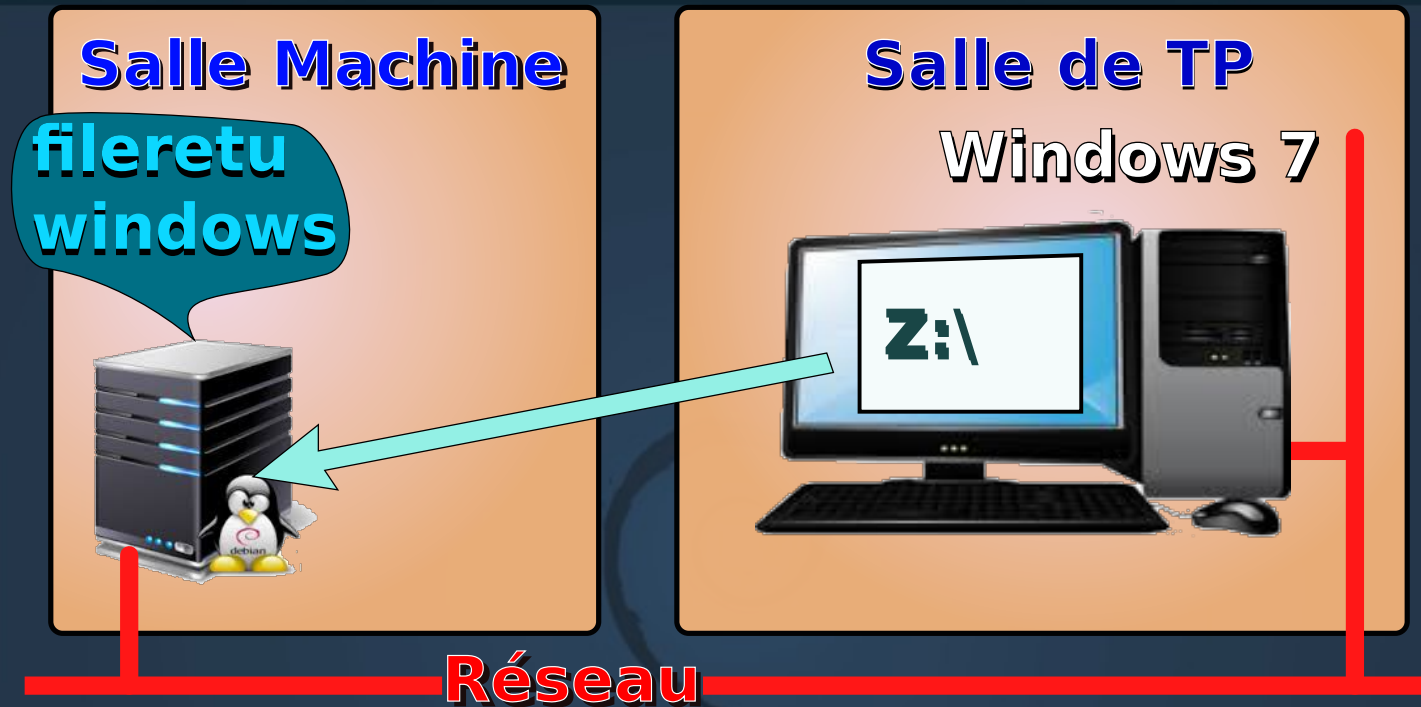
Serveurs de données

- A l'IUT, vos données (fichiers) seront stockées sur un serveur :
 - « **fileretu** » aussi appelé Z:
 - Disponible partout : les fichiers ne sont pas sur une seule machine précise
 - Fiable car :
 - Fichiers archivés toutes les heures
 - Protection globale contre les virus
 - Disques physiquement dédoublés + onduleur...

Modes de travail à l'IUT

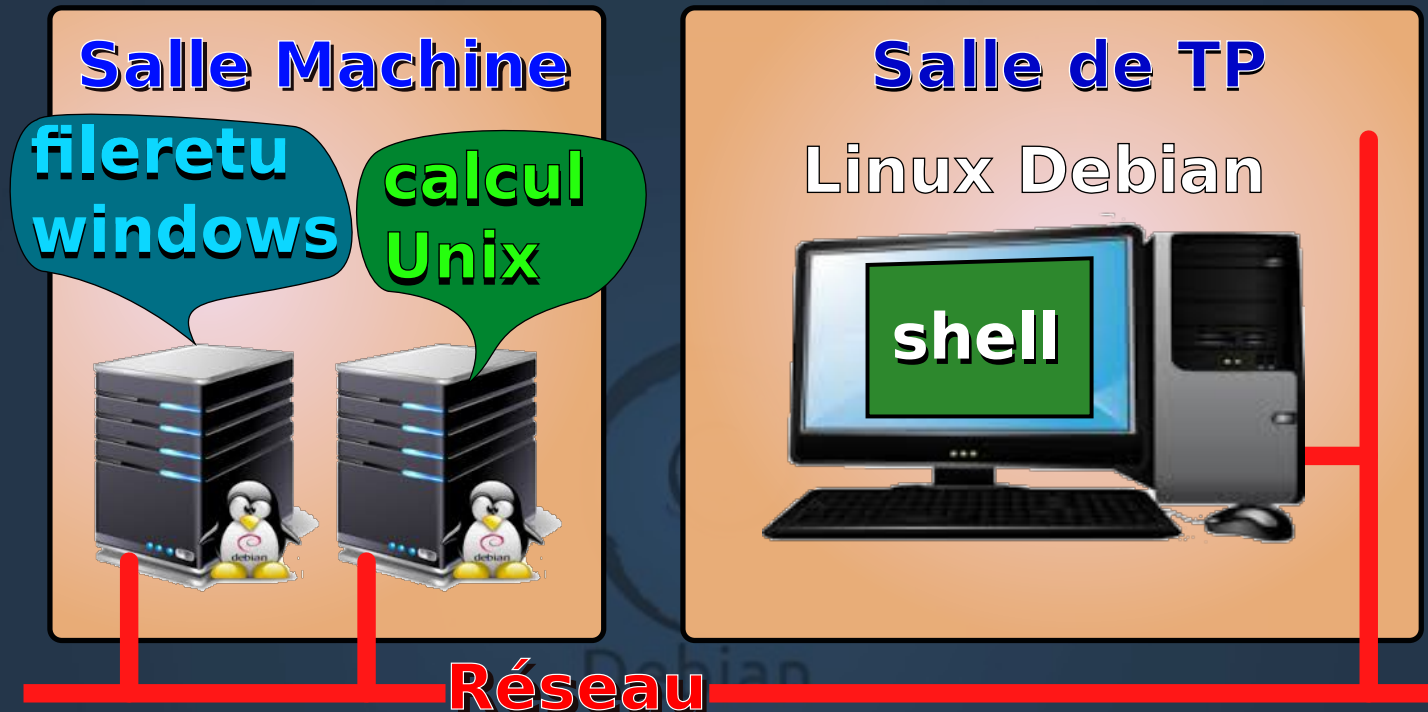
- A l'IUT vous allez travailler avec :
 - **Windows** : comme à la maison (bureautique...)
 - **Linux « en local »** : on démarre le PC avec Linux au lieu de Windows (installation dite en « double boot »)
 - **Linux « à distance »** : on démarre Windows puis on utilise un logiciel de communication (PuTTY+Xming) pour travailler sur une autre machine à travers le réseau (idem Bureau à distance de Windows)
 - Linux sur une **machine virtuelle** : c'est un simulateur de PC sur lequel on installe un système

Windows à l'IUT



- Les documents sont sur le serveur windows appelé « fileretu », visible dans le dossier Z:\

Linux en période 1 et 2



- Vous utilisez le Linux local, avec éventuellement une connexion sur le serveur Unix appelé « calcul »

Linux en période 3



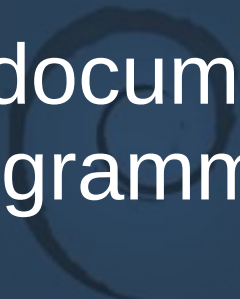
- On lance un simulateur de PC dans le PC et dedans, on lance Linux

Linux à la maison ?

- Avant, on organisait une Linux Install Party
- Mais les portables ont bien changé :
 - Bios UEFI, secure boot, présence de Windows, aucun respect des normes (ex : ACPI), périphériques trop récents et non reconnus...
=> chaque PC demande une installation spécifique
- Impossible de continuer ainsi
- Mais : avec les connaissances de P3, vous pourrez installer Linux sur votre ordinateur.

1.4 - Fichiers

Un fichier est un document : texte, image,
programme...



Debian

Fichier : qu'est-ce ?

- C'est toute sorte de document sur ordinateur :
 - Ex : texte, programme, base de donnée, image...
 - 1 document = 1 fichier



Encodage des données

- **Les fichiers sont « encodés »**
 - une image, un dessin, une musique sont représentés sous forme de nombres (**octets**) : codage des informations
 - octet = nombre allant de 0 à 255 (0 à 2^8-1)
 - ex: images gif, jpeg, fts, ppm... = suite d'octets
 - certains fichiers ne peuvent être utilisés qu'avec le logiciel qui a servi à les créer : format « propriétaire »
 - les fichiers les plus simples : les fichiers « texte »

Exemple d'encodage

- Ce texte :

Bonjour tout
le monde !

- Est encodé ainsi :

66 111 110 106 111
117 114 32 116 111
117 116 10 108 101
32 109 111 110 100
101 32 33 10



Debian produit par : `od -t u1 -t a texte`

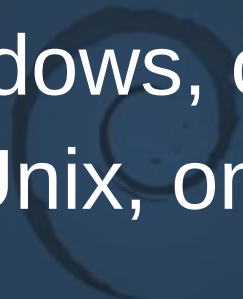
- Chaque signe, y compris l'espace, est codé par un nombre. Ce code s'appelle ASCII, mais il en existe d'autres : ISO-8859-15, UTF8, CP1252...

Nom des fichiers

- **Les fichiers sont nommés : spirou.jpg**
 - Le nom de base : spirou
 - Une « extension » qui indique le type : .jpg. Les extensions sont plus ou moins normalisées
.txt = texte, .jpg = image, .c = prog. source C, .o = objet...
 - NB : Windows cache les extensions (sauf si on coche une option)
- **Les noms+extensions doivent être uniques**
 - On peut avoir spirou.jpg et spirou.txt mais pas deux spirou.jpg

1.5 - Commandes

Avec Windows, on clique...
Avec Unix, on tape...

The Debian logo, which is a stylized spiral, is positioned behind the text. The word "Debian" is written in a light blue font below the spiral.

Debian

Interfaces utilisateur

- **Interface « fenêtres » : Windows et Gnome**
 - clic clic clic clic clic clic... (tout à la souris)
- **Interface « ligne de commande » : Shell Unix**
 - on utilise le clavier pour donner les ordres au PC : une ligne de texte = lancement d'un logiciel
- **Comparaison**
 - **programmation** : impossible avec des fenêtres (sauf avec AutoIT)
 - **travail à distance** : très lourd avec des fenêtres (sauf avec NX client)

Premières commandes

- **logout** : pour se déconnecter
- **who** : pour afficher la liste des utilisateurs connectés sur le même serveur
- **date** : pour afficher la date et l'heure
- On doit taper ce mot puis la touche entrée, et le système effectue l'action demandée.

Commandes Unix

- **Il y a de très nombreuses commandes Unix**
 - 12 à connaître par cœur : cd, ls, more, mv, cp, rm...
 - 2500 autres selon la configuration du système
 - Toutes sont documentées

NB : les étudiants en médecine ont des milliers de termes à apprendre

- **Les commandes Unix sont les mêmes sur tous les systèmes Unix : Solaris, AIX...**
Seules les options peuvent légèrement changer.

Documentation en ligne

- Toutes les commandes Unix sont expliquées dans la doc en ligne :
 - **man commande**
 - description rapide
 - syntaxe
 - description détaillée de toutes les options
 - exemples et cas d'erreur
 - références à d'autres commandes
- Connaître un système = savoir retrouver des informations dans la doc en ligne et sur internet

Structure d'une commande

- Une commande Unix se tape sur une ligne à la suite du « **prompt** » du système :
- **prompt%** *ici je tape ma commande* ◀↵
- On peut fournir des **paramètres** :
 - un nom de fichier par exemple
- On peut fournir des **options** :
 - pour préciser le traitement, pour effectuer certaines opérations...

Syntaxe des commandes Unix

- Voici la syntaxe des commandes :

prompt% *commande options paramètres*

Ex : `ls -l essai1.c`

Ex : `rm -fr tmp*`

Ex: `cc essai1.c -co essai1.o`

- commande = nom de la commande
- options = spécification fine du travail demandé
- paramètres = le ou les fichiers concernés ou d'autres informations

Syntaxe des options

- Sauf exception (tar, dd...), les options sont indiquées par :

- Un - (moins) suivi d'une lettre ou un chiffre **-o**

- On peut regrouper plusieurs options : **-l -a = -la**

- Deux moins suivis d'un mot **--long**

- C'est parlant, mais plus long à taper

- Une valeur peut être fournie après l'option, hélas chaque commande le fait à sa façon :

od -t u1 bonjour.txt ou **od --format=u1 bonjour.txt**

1.6 – Commandes simples

lister les fichiers
voir un fichier
renommer, copier, supprimer...

Debian

Lister les fichiers

ls

donne la liste des noms des fichiers

- L'option `a` montre aussi les fichiers dont le nom commence par un point (leur nom est uniquement un suffixe)
- Exemples:
 - `ls`
 - `ls -a`
- doc : `man ls`



Afficher un fichier texte

cat *fichier*

affiche le contenu du fichier texte

- Exemples:
 - **cat proverbes.txt**
 - **cat essai1.c**



Afficher un texte page par page

`more` *fichier*

affiche le contenu du fichier texte page par page afin qu'on puisse le lire même s'il est très long

- Exemples:
 - `more proverbes.txt`
 - `more essai1.c`



Mode d'emploi de more

- more affiche la première page (dépend de la hauteur de la fenêtre) puis attend en bas de l'écran qu'on appuie sur une touche :
 - q pour arrêter (quit)
 - espace pour la page suivante
 - b pour la page précédente (back)
 - /chose pour chercher « chose » dans la suite
 - n pour réitérer la même recherche (next)

Modifier, créer un fichier

vi fichier

emacs fichier

gedit fichier

« édite », c'est à dire affiche et permet de modifier le fichier texte ou source

- mode d'emploi : voir TP n°1
- Exemples:
 - ***vi proverbes.txt***
 - ***emacs essai1.c***

Imprimer un fichier texte

`lpr` *fichier*

imprime le fichier dans la salle « libre service » (RdC)

Faites attention à ce qu'il contienne vraiment du texte

Mettez votre nom dans le document !!!

Attention au gâchis de papier, on peut vous demander des comptes (voir la ligne « photocopiés » des droits d'inscription)

- Exemples:
 - `lpr proverbes.txt`
 - `lpr essai1.c`

Renommer un fichier

mv ancien nouveau

change le nom du fichier appelé *ancien* en *nouveau* (et écrase *nouveau* s'il existait déjà)

- Exemples:
 - ***mv proverbes.txt maximes.txt***
 - ***mv essai1.c essai1.txt***
 - utiliser `ls` pour vérifier le changement : soyez « parano » car un écrasement est définitif !

Dupliquer un fichier

cp *fichier nouveau*

crée une copie du fichier (et écrase *nouveau*)

- Exemples:
 - **cp proverbes.txt premier.txt**
 - **cp essai1.c essai2.c**
 - utiliser ls avant et après, et more pour confirmer la création du fichier, soyez « parano » !

Supprimer un fichier

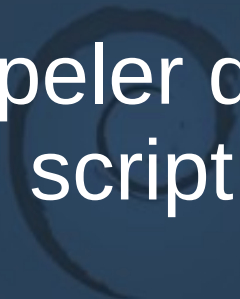
`rm fichier`

détruit définitivement le fichier

- Exemples:
 - `rm maximes.txt`
 - `rm essai1.c`
- « Mode parano » obligatoire : toujours utiliser `ls` et `more` pour vérifier préalablement que c'est le bon fichier : la suppression est définitive.

1.7 - Jokers

Pour désigner plusieurs fichiers en une seule commande

The Debian logo, which is a stylized white spiral on a dark blue background, is positioned behind the text.

Debian

Jokers dans les noms des fichiers

- Pour désigner plusieurs fichiers dont les noms ont une partie commune, on remplace la partie qui varie par *
- Ce caractère * est appelé joker.
 - Ex : tous les fichiers appelés « toto... » : toto*
 - Ex : tous les fichiers finissant par la lettre c : *c
- Exemples :
 - more *.txt
 - rm essai*.txt

Rôle d'un joker

- Soit la commande : `rm tmp*.txt`
 - Le système liste tous les fichiers tmptxt et les met à la place de tmp*.txt
 - => ça fait exécuter `rm tmp1.txt tmpok.txt tmpZ.txt`
 - Ça lance rm avec ces trois paramètres comme si on les avait tapés directement
 - => impossible de faire `cc prog*.c` :
 - Ça ne fera pas `cc prog1.c` puis `cc prog2.c` etc.
mais `cc prog1.c prog2.c prog3.c` => cc pas content
- Si rien ne correspond, ça fait une erreur

Le joker ?

- Pour indiquer un seul caractère variable, il y a le joker ?
 - `ls hello?.txt`
 - `more bon???r.txt` -> `bonjour.txt` et `bonsoir.txt`



Le joker []

- Si on veut restreindre encore plus, il y a le joker [...]:
 - Un intervalle : [caractère1-caractère2]
 - Une liste : [liste de caractères]

exemples :

```
rm test[1-5].txt essai[21].txt
```

Le joker {,,}

- Enfin, pour spécifier des alternatives pour une partie du nom d'un fichier :
début{variante1, variante2, ...}fin
- Exemples :
 - `more essai*.{c,txt}`
va afficher `essai1.c`, `essai2.txt`, `essai3.c` (selon ce qu'il y a)
 - `rm *.{temp,tmp,o}`
va supprimer tous les fichier finissant par `temp`, `tmp` ou `o`
 - Ça revient à faire `rm *.temp *.tmp *.o`