Systèmes d'Exploitation
1ère année DUT Info Lannion

Gildas QUINIOU

- Enseignant info DUT et LP Web
- Développeur d'applications Web et Mobiles
- Concepteur et développeur de jeux vidéo

1 - Introduction

Le cours de Systèmes a une importance particulière.

Pour un menuisier, une scie, un rabot sont des outils essentiels. Il les utilisera toute sa vie professionnelle.

Pour l'informaticien, le Système d'Exploitation est aussi un outil qu'il utilisera toute sa vie professionnelle.

Le Système d'Exploitation est un élément fondamental de tout ordinateur, qu'il soit sous une forme traditionnelle (Unité Centrale, clavier, écran) ou sous la forme plus exotique d'un téléphone, une montre, un appareil photo ou un robot tondeuse à gazon.

Tous ces équipements embarquent un Système d'Exploitation !

C’est une UE (Unité d’Enseignement) qui n’est pas difficile, sous réserve de :

- Être présent (faut-il le préciser ?)
- Actif
- Curieux
- Poser des questions

Module M1101

- M : Module.
- Semestre 1.
- Unité d'Enseignement (UE) 1.
- Module 01 dans l'UE.

Calendrier

- P1 (5 sem.) : 2H CM, 1H TD (machine), 2H TP.
- P2 (5 sem.) : 1H CM, 1H TP.
- P3 (6 sem.) : 1H CM, 1H TD (machine), 2H TP.

Total : 21H CM + 11H TD + 27H TP = 59H.

Contrôles

- Contrôles continus en TP, sur P1, P2 et P3.
- Contrôle TP en temps limité (?) (P2 ou P3, plutôt P2).
- 2 DS (P1 et P3).

Notes

- DS : Coef 2.
- TP : Coef 1.

2 - Qu’est-ce qu’un OS ?

Le terme Système d'Exploitation est généralement raccourci en SE ou OS (Operating System).

C'est le cerveau logiciel de l'ordinateur.
Un chef d'orchestre pour que le matériel et les logiciels fonctionnent en harmonie.

L'OS reçoit des ordres et les exécute.

L'OS est :

- Lancé au démarrage de l'ordinateur.
- Prépare l'ordinateur (composants et périphériques).
- Tourne jusqu'à l'extinction de l'ordinateur.

L'OS sert d'intermédiaire entre le matériel et les autres logiciels.

3 - Les types d'OS

Mono-tâche

Un OS mono-tâche donne la main exclusive à un seul processus.

Exemple : MS-DOS.

Multi-tâche(s)

Un OS multi-tâche permet de lancer plusieurs processus/programmes en même temps.

Exemples : Windows, Linux.

Multi-utilisateur

Un OS multi-utilisateur dispose de mécanismes de cloisonnement et de protection des données :

- Utilisateur a son environnement personnel.
- Utilisateur ne peut pas voir les données des autres utilisateurs.

Exemples : Windows (95 et plus), Linux.

Les OS multi-utilisateur peuvent être :

- Multi-utilisateur simultané : plusieurs utilisateurs connectés en même temps.
  Exemple : Linux.
- Multi-utilisateur exclusif : un seul utilisateur possible à un instant donné, mais plusieurs comptes sur l'ordinateur.
  Exemple : Windows avant 2000.

Mono-utilisateur

Un OS mono-utilisateur ne dispose généralement pas de ces mécanismes de cloisonnement.

Exemple : MS-DOS.

Aujourd'hui on a généralement affaire à des OS :

- Multi-tâche.
- Multi-utilisateur.

Mais certains équipements peuvent encore être mono-tâche et/ou mono-utilisateur.

Exemple : une calculatrice programmable.

4 - Unix : Linux & Cie

Bref historique

Unix :

- Créé en 1969 par Ken Thompson, au laboratoire AT&T de la société Bell. Nom d'origine : Unics.
- Ecrit en langage assembleur (maintenance complexe).
- Réécrit en 1971, nouveau langage, le C, par Ken Thompson et Dennis Ritchie (Papa du langage C).

Unix :

- 1969 (août) : Un peu la préhistoire de l'informatique.
- Conception incroyablement moderne.
- De nombreuses évolutions, mais des bases toujours présentes.
- 2018 : Un des OS les plus utilisés.
- Un OS mais aussi un concept d'OS.
- Plusieurs Unix, dérivés de l'Unix originel d'AT&T Bell. Exemple : Linux.

Linux :

- 1991.
- Œuvre d'un finlandais, Linus Torvalds.
- Clin d'œil à l'histoire, Linus est né en 1969 (comme Unix).

Aujourd'hui, Linux est :

- L'OS le plus présent sur les serveurs.
- De très loin, l'Unix le plus populaire.

Principes de base

Unix :

- Multi-tâche.
- Multi-utilisateur.

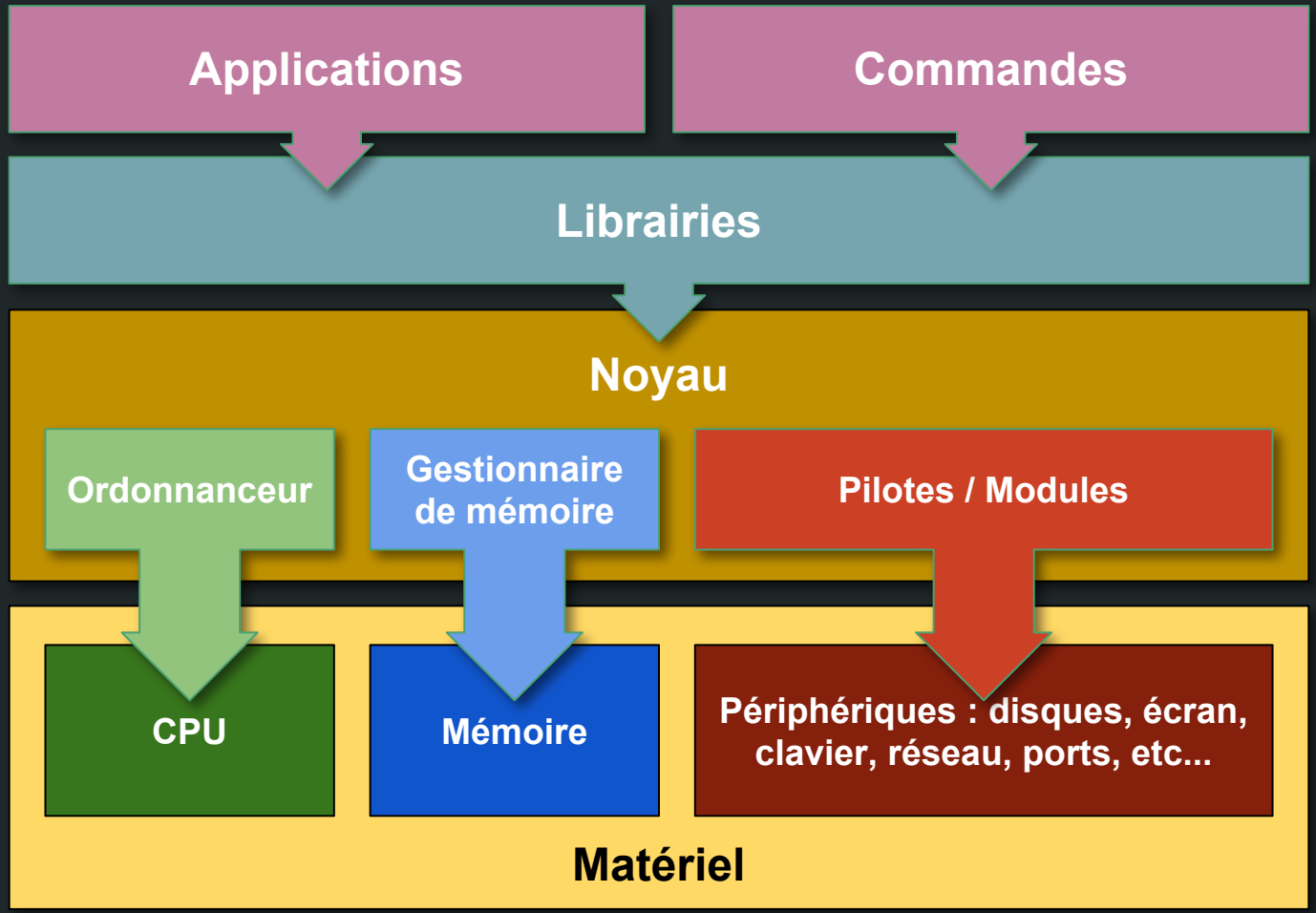
Des caractéristiques communes :

- Gestion des fichiers, des ressources, des processus…
- Jeu de commandes de base.

A partir de maintenant :

- On va se focaliser sur Linux.
- Tout ce qui sera dit dans ce cours sur Linux se rapporte aussi aux autres Unix en général.

5 - Linux



L'OS Linux c'est :

- Noyau (Kernel en anglais).
- Pilotes ou Modules (interface avec le matériel).

Par abus de langage, on considère aussi que les commandes d'administration font partie de l'OS.
Nous aussi dans ce cours !

6 - Que va-t-on apprendre ?

Plein de choses utiles et sympathiques…

- Se connecter et se déconnecter d'un ordinateur.
- Donner des ordres, exécuter des commandes.
- Manipuler les fichiers.
- Se déplacer, manipuler l'arborescence.
- Comprendre, manipuler les droits sur les objets.
- Agir sur les processus.
- Ecrire des scripts.

7 - Les objets Linux

Les “objets” Linux que nous allons voir et/ou utiliser :

- Utilisateurs et Groupes.
- Fichiers et Dossiers.
- Processus.

Notes sur les utilisateurs :

- root (c’est le nom donné à Dieu en informatique).
- Des utilisateurs fictifs (contrôle de services, Serveur Web ou Serveur de Mail par exemple).

8 - Le login, le compte

Chaque utilisateur de l'OS possède un compte.

Un compte est constitué de :

- Un nom (unique).
- Un mot de passe (stocké crypté).
- Un ID (numéro unique).
- Un home (répertoire d'accueil).
- Un shell (optionnel).

Le login

De l'anglais "log" qui est le journal de bord sur un bateau, dans lequel le capitaine consigne les événements.

Quand un utilisateur se connecte, cet événement est aussi consigné dans un fichier d'événements. Ce qui a donné l'expression "to log in" et finalement le login.

Le login est le nom que l'utilisateur a pour se connecter, pour se "logger".

Ce nom est une simple commodité. Une fois l'utilisateur loggé, l'OS utilise ensuite uniquement son ID (valeur numérique). Ainsi, on peut changer son nom plus tard sans rien chambouler pour l'OS.

Le logout est l'action de se déconnecter de l'ordinateur.

Le mot de passe

A la connexion de l'utilisateur, le mot de passe qu'il fournit est vérifié soit localement, soit en utilisant un annuaire sur un autre serveur du réseau.

Le mot de passe est toujours stocké crypté. Il n'y a aucun moyen de connaître le mot de passe d'un utilisateur.

Le home

Chaque utilisateur a un répertoire de travail sur le disque de l'ordinateur. Un espace où il peut stocker ses fichiers de travail.

On l'appelle le "home directory" ou simplement "home".

Le shell

En anglais “shell” signifie coquille (pour un crustacé) ou coque (pour un fruit). Et qu’y a-t-il dans une coque de fruit ? Il y a le cœur du fruit et son noyau.

Noyau… Kernel ! Et le message pourrait donc être :

Qui a accès au shell a accès au kernel

Si le compte de l'utilisateur n'est pas associé à un shell (ou un programme similaire), alors il ne pourra pas se connecter.

Une fois connecté, l'OS est au service de l'utilisateur :

- Il lui donne des ordres via le shell.
- L'OS les exécute.

Tout ceci sous réserve que l'utilisateur dispose des droits nécessaires pour l'ordre qu'il donne.

9 - Le shell

Le rôle du shell est de service d'interface entre l'utilisateur (via son clavier) et l'OS.

Un shell :

- Affiche un prompt. Exemple : username $ _
- Lit des ordres (généralement) au clavier, terminés par ENTREE.
- Passe la main à l'OS pour exécuter l'ordre.
- Affiche le résultat (généralement) à l'écran.
- Retourne à l'affichage du prompt.

Attention ! Warning ! 警告 ! Achtung ! تحذير ! Attenzione !

On n’est pas sous Windows.
Le shell ne va pas s’encombrer de confirmations.
Vous ordonnez, il exécute. Point barre.

10 - Les fichiers

Un fichier est :

- Une zone allouée généralement sur un espace de stockage (disque dur, clé USB, Cloud).
- Identifié par un nom.
- Contient des données quelconques (image, texte, son, vidéo).

On trouve aussi le nom de document.

Un fichier peut ne rien contenir, on dit qu'il est vide.

Un fichier peut :

- Se suffire à lui même : lisible directement.
  Exemple : fichier texte.
- Nécessiter un logiciel pour être exploitable.
  Exemple : fichier Word ou fichier Excel.

Les fichiers seront vus de façon plus détaillée dans un autre CM.

Le nom de fichier

Un fichier est identifié par son nom, tel que logo.png, composé de :

- Un nom de base : “logo”.
- Une extension : “.png” (commence toujours par “.”).

Chacune de ces 2 parties est facultative mais il en faut au moins une des deux pour former un nom de fichier.
Exemple : “logo” ou même “.png” tout seul.

11 - Les commandes

Comprendre les noms des commandes

Avant tout, il faut se rappeler qu'Unix :

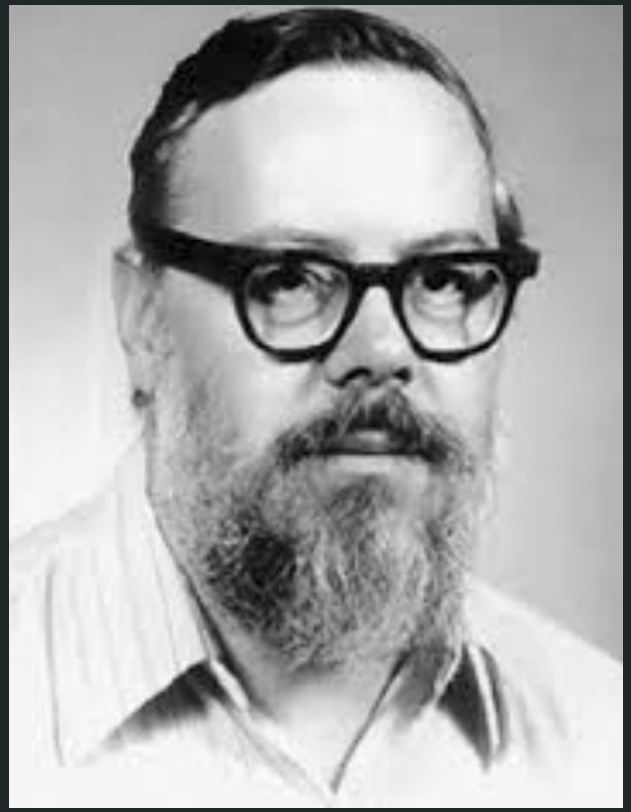
- Date des années 70.
- Etait conçu par des barbus pour des barbus.

Ken THOMPSON

Le Papa d'Unix.

Dennis RITCHIE

Le Papa du langage C.



Richard STALLMAN

Initiateur et promoteur du Logiciel Libre (dont fait partie Linux).
Le Papa de GNU.

Attention aux contrefaçons

Le Papa Noël.

A cette époque, un non barbu ne pouvait pas comprendre Unix !

Bref, c'était une époque où ils (les barbus) ne se souciaient pas trop de rendre leur joujou accessible aux étudiants de 1ère année de l'IUT de Lannion !

Tout ceci a donc un impact sur le nommage des commandes, comme on va le voir.

Les commandes que je vais citer dans cette rubrique sont là uniquement pour illustrer les propos.

Ce n'est pas la peine de retenir leur fonction ici, on va revenir en détail sur chacune, plus loin.

Les commandes de base portent des noms pas super évidents à comprendre si on ne dispose pas des clés.

Voici donc quelques astuces.

Globalement, les commandes sont des versions raccourcies d'un ou de plusieurs mots.

Certaines commandes de base ont des noms sur 2 lettres :

- cp : copy
- mv : move
- ls : list
- rm : remove

Ce sont des mots simples, ils ont simplement gardé les 2 premières consonnes.

Ensuite on trouve des commandes qui sont la combinaison ou les initiales de plusieurs mots comme :

- cd : change directory
- pwd : print working directory
- mkdir : make directory
- rmdir : remove directory
- man : manual
- wc : word count
- top : table of processes
- vi : visual instrument
- vim : vi improved
- grep : globally search regular expression and print
- awk : Aho + Weinberger + Kernighan (3 créateurs)

On a aussi des mots courts pour lesquels on conserve toutes les lettres :

- date : date+heure courante.
- who : qui est connecté.
- head : début (tête) d'un fichier.
- tail : fin (queue) d'un fichier.
- tree : arborescence (arbre) de fichiers.

Il n'y a pas de règle unique sur le nommage des commandes, mais il faut chercher la signification derrière les quelques lettres, ça aide à comprendre et à retenir le rôle de chacune.

Se tromper dans une commande parce qu'on a confondu rm (qui supprime des fichiers) et cd, ça peut faire mal. La bonne nouvelle c'est que ça ne fait mal qu'une seule fois, après on s'en rappelle.

Commande la plus utile pour débuter

La commande à connaître avant tout est man qui appelle le manuel, la documentation utilisateur de n'importe quelle commande :

man <nom_d_une_commande>

man man : Affiche le manuel d'utilisation du… manuel !

Règle d'Or

On oublie Google et on utilise ce qu'on a sous la main : le manuel !

Une nouvelle commande inconnue ?
Le seul réflexe à avoir est : **man**

Format d'une commande

Une commande est composée de plusieurs parties :

- Nom de la commande.
- Une ou plusieurs options (parfois facultatives).
- Des paramètres (parfois facultatifs).

Exemple avec ls -l toto.png :

- ls est la commande. Elle affiche des informations sur des fichiers.
- -l est une option. Une option permet de modifier le comportement standard de la commande. Ici cette option permet d'afficher des informations plus détaillées (l = long).
- toto.png est un paramètre. Ici c'est le nom du fichier qu'on cible.

Options d'une commande

Les options peuvent être de 2 formes :

- Courte : un tiret suivi d'un lettre.
  Exemple : -a, -l, -n
- Longue : 2 tirets suivis d'un mot.
  Exemple : --inode ou --help

Rappel : pour avoir des informations détaillées sur les options possibles d'une commande donnée, il suffit de consulter son manuel.

Exemple avec ls : man ls

Quand on a besoin de mettre plusieurs options courtes, il est possible de les coller ensemble.
Exemple :

ls -l -a

devient

ls -la

12 - Les commandes de base

Les commandes de base sont assez peu nombreuses.

Vous devez les connaître par cœur.

Nous n'allons par les détailler, soyez curieux, consultez le manuel pour voir tous les détails de ces commandes.

Les commandes qui suivent servent à manipuler des fichiers.

Dans les syntaxes données ci-après, ce qui se trouve [entre crochets] est facultatif, le reste est obligatoire.

Par soucis de simplicité, certaines syntaxes présentées dans les exemples sont un peu simplifiées.

Avec l'expérience et en consultant le manuel, vous apprendrez des usages plus élaborés.

ls - Affiche une liste de fichiers

Origine du nom : list.
Syntaxe : ls [options] [fichiers]

Quelques exemples :

- ls
- ls -l
- ls prog1.c prog2.c

rm - Supprime des fichiers

Origine du nom : remove.
Syntaxe : rm [options] fichier(s)

Quelques exemples :

- rm brouillon.txt
- rm -i liste.old essai.txt

Attention, rm ne demande pas de confirmation avant la suppression, sauf avec l'option -i

cp - Copie (duplique) des fichiers

Origine du nom : copy.
Syntaxe : cp [options] original duplicata

Quelques exemples :

- cp agenda2018 sauvegarde
- cp -v photo.png profile.png

Attention, si un fichier existe déjà avec le nom du duplicata, il est simplement écrasé sans avertissement.

mv - Renomme des fichiers

Origine du nom : move.
Syntaxe : mv [options] ancien_nom nouveau_nom

Quelques exemples :

- mv a_faire.txt fait.txt
- mv -u devis_aout ventes_aout

Attention, si un fichier existe déjà avec le nouveau nom, il est simplement écrasé sans avertissement.

cat - Affiche le contenu de fichiers

Origine du nom : concatenate.
Syntaxe : cat [options] fic1 [fic2...]

Quelques exemples :

- cat groupe_f
- cat -n promo_2016 promo_2017 promo_2018

more/less - Affiche des fichiers de façon paginée

Origine du nom : l'affichage se faisant par page, la commande affiche --More-- en fin de page et l'utilisateur peut appuyer sur espace pour voir la page suivante.
D'où le nom de la commande : more.

less est une version améliorée de more.

Syntaxe : more [options] fic1 [fic2…]

Quelques exemples :

- more bible.txt
- less a_la_recherche_du_temps_perdu.txt

Usage de base :

- Espace : page suivante.
- b : page précédente (back).
- Entrée : défile d'une ligne vers le bas.
- q : quitte.
- Flèches haut et bas : défile d'une ligne.

vi - Edite le contenu de fichiers

Origine du nom : visual instrument
Syntaxe : vi [options] fic1 [fic2…]

vi est :

- un vieil éditeur de texte.
- le seul éditeur présent sur tous les Linux.
- parfois le seul éditeur présent et/ou utilisable.
- d'un usage un peu déroutant au départ :-)

13 - Les jokers du shell

Le joker *

Le joker * permet de remplacer un nombre quelconque (y compris 0) de caractères.

Exemple : *.c signifie “n'importe quel fichier qui porte une extension .c”.

Exemples de fichiers qui correspondent :

- prog.c : * replace prog
- .c (un nom bizarre, mais autorisé) : * replace du vide.

Exemple de fichier qui ne correspond pas :

- source.cc : * pourrait replacer source mais l'extension attendue est .c et on a .cc

Le joker ?

Le joker ? permet de remplacer un et un seul caractère.

- ?.txt signifie “n'importe quel fichier dont le nom de base contient 1 caractère et qui porte l'extension .txt”

Exemple de fichier qui correspondent :

- a.txt : ? remplace a

Exemple de fichier qui ne correspond pas :

- aa.txt : ? remplace un seul caractère, donc pas aa

Le joker [ ] - Liste

Le joker [ ] permet de remplacer un et un seul caractère parmi une liste exhaustive de caractères acceptables :

- [ax].proj signifie “n'importe quel fichier donc le nom de base est a ou x, et qui porte l'extension .proj.

On a donc seulement 2 possibilités : a.proj et x.proj
Mais ax.proj ne correspond pas.

Le joker [ ] - Intervalle

Le joker [ ] peut aussi indiquer un intervalle de caractères.

Par exemple : [a-z] représente tous les caractères de a à z en minuscules. C'est une forme raccourcie de [abcdefghijklmnopqrstuvwxyz].

On peut mettre plusieurs intervalles et même y ajouter des caractères isolés en plus.

Exemple : [a-zA-Z0-9] représente les minuscules, les MAJUSCULES, tous les chiffres, et uniquement cela.

Un intervalle doit respecter l'ordre d'apparition des caractères dans la table ASCII (www.asciitable.com).

[a-z] ne pose pas de problème, le a étant évidemment avant le z, et les lettres sont consécutives. Mais on ne peut pas écrire [a-Z], ni même [A-z] pour 2 raisons (évidentes, si on connaît la table ASCII) :

- Les minuscules apparaissent après les MAJUSCULES.
- Entre la zone des minuscules et celle des MAJUSCULES, il y a 6 autres symboles.

Le joker { }

Le joker { } permet d'indiquer une alternative, parmi une liste de valeurs possibles, à un endroit donné du nom du fichier. Les valeurs sont séparées par des virgules.

Exemple : {ete,hiver}.ventes correspond aux 2 cas ete.ventes et hiver.ventes, et uniquement ceux-là.

Cas particulier du “.”

Attention, le “.” qui figure dans l'extension est un caractère comme les autres. Ainsi a* peut correspondre à tous les fichiers suivants :
a, abc, a.txt ou encore a. (nom bizarre, mais autorisé)

En fait, a* signifie : “un fichier commençant par a et suivi de n'importe quoi (y compris un point ou rien du tout).

Mais a.* signifie : “un fichier commençant par a, suivi d'un point, et suivi de n'importe quoi (y compris rien du tout)”. Donc :

- a : pas bon, il manque le point
- abc : pas bon, idem
- a.txt : ok
- a. : ok
- a..txt : ok aussi (nom bizarre mais autorisé)

Combinaisons

Les jokers peuvent aussi être combinés et chacun peut apparaître plusieurs fois.
Exemples :

- [a-z]?? : un fichier commençant par une et une seule minuscule et ayant 2 caractères quelconques après (ane, bac, dut, b52). Mais pas BAC ni IUT.

Exemples (suite) :

- *[56]?[aeuioy] : un fichier qui contient quelque part un 5 ou un 6 suivi immédiatement d'un caractère quelconque, suivi immédiatement d'une des 6 voyelles et sans rien d'autre derrière :

  5_i, blah+6_a, ok56e, mais pas ok56ok.
  Et que pensez-vous de ok56ee ?

Exemples (suite) :

- {bob,fred}.{png,jpg} : un fichier dont le nom de base est soit bob, soit fred, et dont l'extension est soit png, soit jpg.

  On a donc 4 noms possibles.

Conversion des jokers

Le shell qui transforme une séquence de jokers en une liste de fichiers correspondants.

Exemple avec 3 fichiers : toto, titi et tutu.

Pour ls t*, le shell va d'abord convertir en ls toto titi tutu.
C'est le travail du shell, pas de la commande, de faire ce développement des jokers en résultat final.

https://commons.wikimedia.org/wiki/File:Thats_all_folks.svg