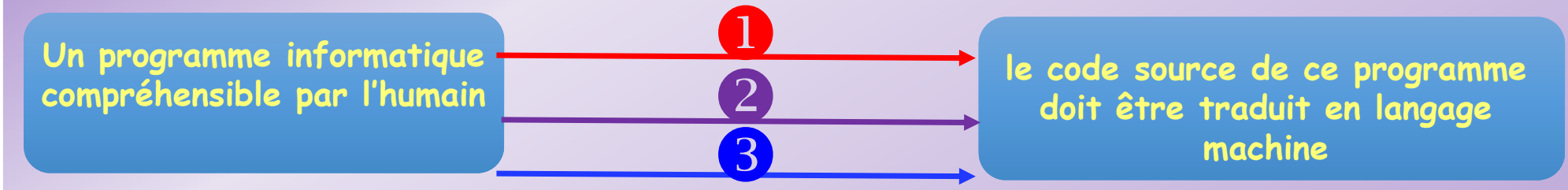
Définitions & commandes

LINUX propose pour l'utilisateur deux Modes d'interfaces : Graphique ou Texte

Dans le mode Texte :

Création d'un terminal : CTRL + ALT + t

Programme interprété ou compilé ?



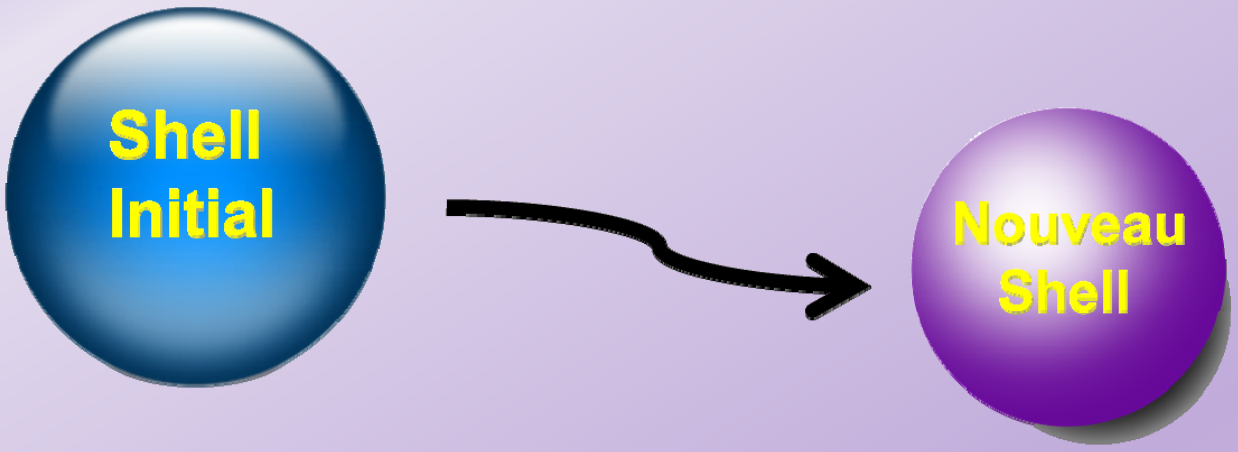
1 **Langage compilé** : La traduction se fait une fois pour toutes
- Le **compilateur** génère un fichier exécutable a partir du code source
- Inconvénient : Recompilation du programme pour chaque changement de plateforme

2 Langage interprété : la traduction de chaque instruction se fait en temps réel puis exécuter.
- Cela nécessite la présence d'un **interpréteur**.
- Avantages : le programme peut être exécuté sur plusieurs plateformes différentes
- Inconvénient : l'interprétation du code à chaque exécution a un impact sur les performances

3 **Langage semi-interprété** : Le langage semi-interprété est traité par un interprète.
le code source à chaque exécution est préalablement traduit dans un langage intermédiaire (**bytecode**) proche du langage machine. Il préserve les bonnes performances

- Le **shell** assure l'interface externe entre le système et l'utilisateur

- Un véritable langage de programmation **interprété**



- L'exécution d'une commande entraîne généralement la **création** d'un nouveau shell

➔ Le shell n'est rien d'autre qu'un simple programme :

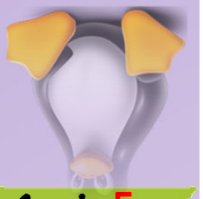
Le choix est vaste : **sh, dash, ksh, csh, tcsh, ash, bash, zsh, …**

Les shells historiques :

- **/usr/bin/sh** (Bourne-shell 1979) — Le premier SHELL de UNIX
- **/usr/bin/csh** Cshell 1981) — Amélioré et proche de la syntaxe du langage
- **/usr/bin/bash(Bourne-Again-shell 1991)** — **Une amélioration du Bourne Shell apparu avec LINUX** ✓
- **/usr/bin/tsh** (Tshell 1981) — Plus moderne et améliore les autres shell. Utilisé par FreeBSD
- **/usr/bin/ash** (Ashell) — On le trouve dans les systèmes embarqués
- **/usr/bin/ksh** (Korne-shell 1989) — Améliore Bash et présent dans les UNIX propriétaires (IB
- **/usr/bin/zsh** (Zshell) — Est la synthèse de tous les autres shell. Il a pris le meilleur des shells précédents. Il est très **innovant**

➔ Le shell n'est rien d'autre qu'un simple programme :

- Le nom du shell : echo ${SHELL}
- La version du SHELL : echo ${BASH_VERSION}
- Lancer automatiquement par le système lors du login d'un utilisateur
- Chaque utilisateur dispose de son propre shell paramétrable (voir .bashrc)
- Deux modes de traitement :
  - **Interactif**
  - **Scripting**



Voir résumé (P.4)

I.1 Définitions préliminaires

Séparateur : Est un espace ou tab ou retour à la ligne ou un caractère défini par l'utilisateur ou toute combinaison possible entre ses caractères (voir la variable **IFS**)

Mot : Est une suite finie non vide de caractères qui ne contient pas de séparateurs

Nom: Est une suite finie non vide de caractères alphanumérique ou "_" ou " -" ou " ." commençant obligatoirement par une lettre ou "_" -" ou " ."

Qu'est ce qu'une commande ?

- Est une **phrase** d'au moins un **mot**
- Le premier mot est une commande **existante**

Quoi FAIRE

Comment le FAIRE

Sur quoi le FAIRE

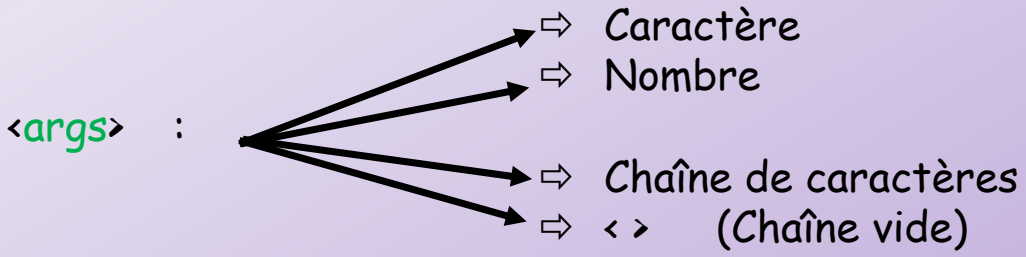
Syntaxe : **Cmd** [± options] [arguments]

- ⇨ **Cmd** : Réalise un traitement bien spécifique
- ⇨ **Arguments** : Nomment les objets sur les quels doit agir la commande
- ⇨ **options** : Modifient la manière d'agir sur les objets

Exemples :	cmd	opts	args
	ls	<>	<>
	ls	<>	rep
	ls	-lai	rep

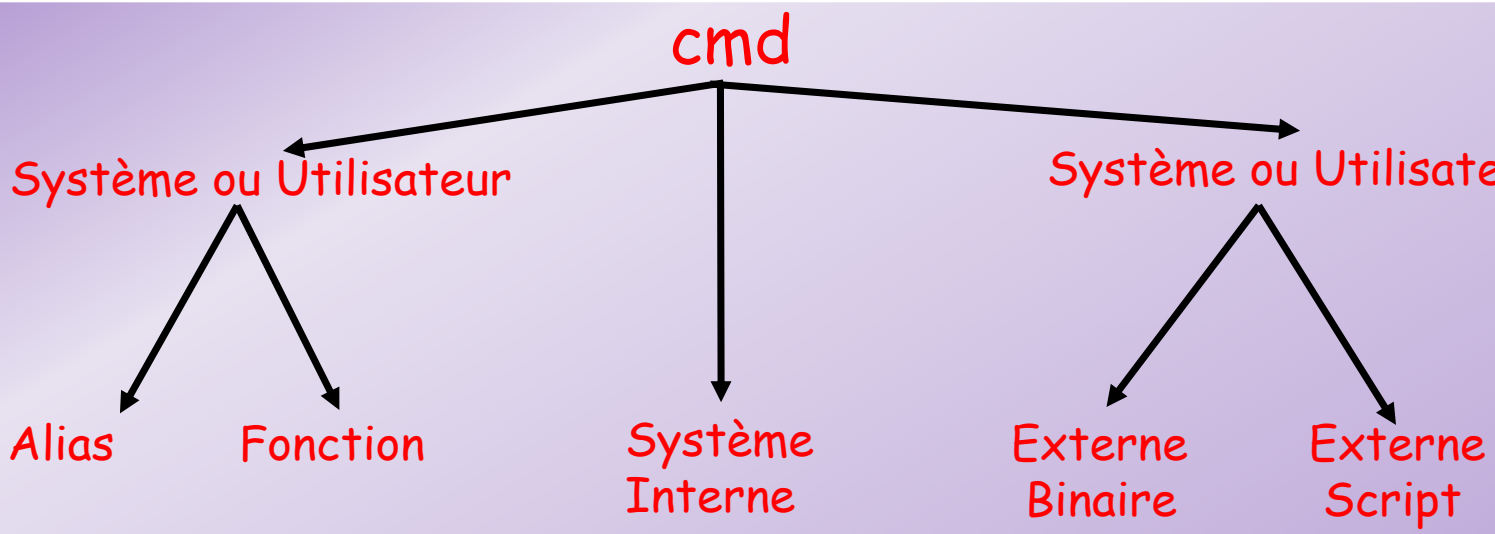
Qu'est ce qu'un argument?

Syntaxe : **Cmd** [± **options**] [**arguments**]



Les arguments sont séparés par :

- ⇨ Espace : ␣
- ⇨ Tabulation : touche <tab>
- ⇨ ↵ : Fin de la liste d'arguments.
- ⇨ \↵ : Si la liste d'arguments. est longue
- ⇨ Séparateur défini par l'utilisateur (voir variable : IFS)
- ⇨ Utiliser " autour de l'argument s'il est composé de plusieurs mots



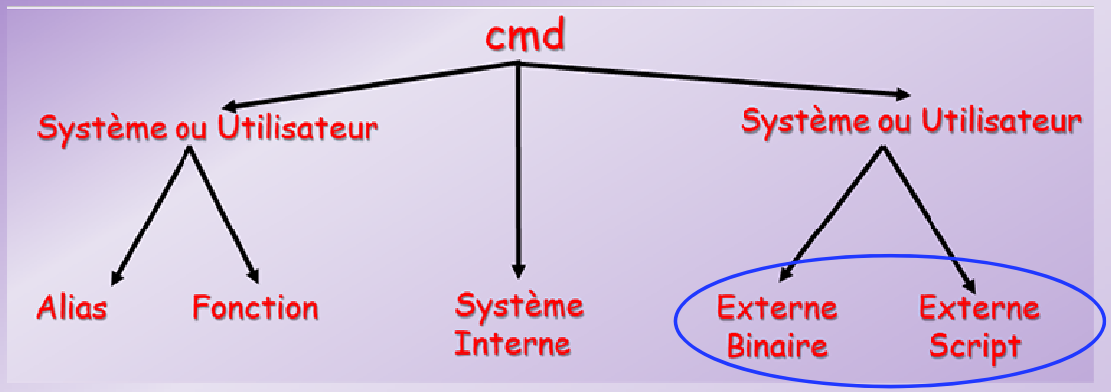
Cmd (alias ou fonction) : doit figurer dans l'environnement (voir commande : **set**)

Cmd (commande interne) : doit se trouver dans bash (voir le manuel de : **bash**)

Cmd (binaire ou script) : correspond à un **fichier** localisé dans un dossier

Pour savoir dans quelle catégorie se trouve **cmd**. Utiliser la commande :

type cmd

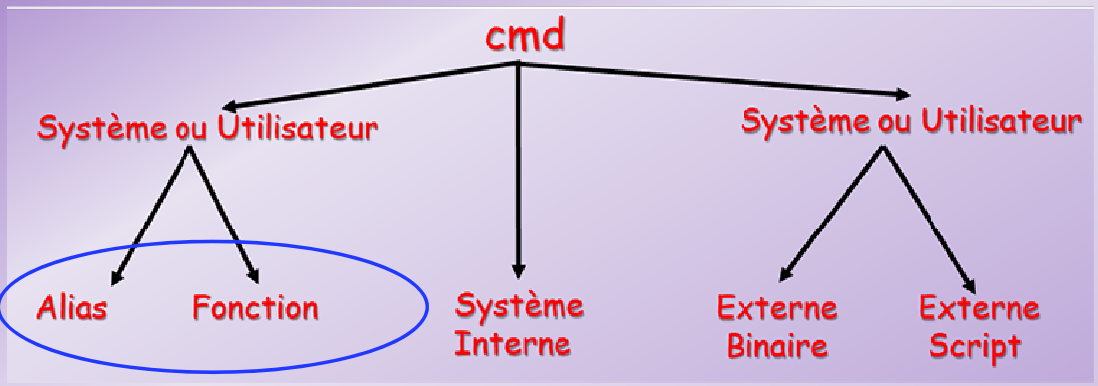


Cmd (binaire ou script) correspond à un fichier localisé dans un dossier

- Le shell courant crée un sous shell
- Le sous shell cherche dans la variable PATH un dossier qui contient cmd

Si Le sous shell n'a pas trouvé Alors
"Erreur : cmd introuvable" est affichée
Sinon Si la liste d'options ou d'arguments est erronée Alors
"Erreur : sur options ou arguments" est affichée
Sinon Si l'utilisateur n'a pas le droit d'exécuté cmd Alors
"Erreur : Opération non permise" est affichée
Sinon Le sous shell exécute cmd
Fsi
Fsi
Fsi

Exemple : ls, mkdir , cp, …



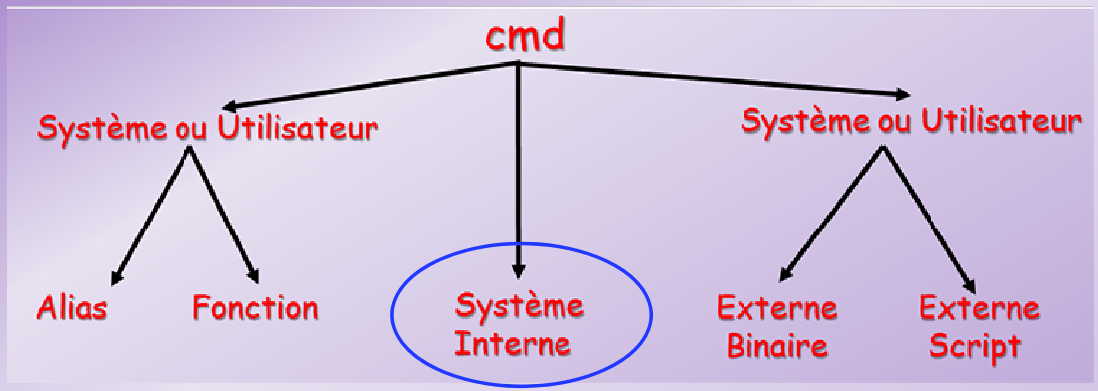
Cmd (alias ou fonction)

doit figurer dans l'environnement

- Le shell courant cherche l'existence de l'alias ou la fonction
- Si Le shell courant n'a pas trouvé Alors
  "Une erreur " est affichée
  Sinon Si Il y a erreurs de syntaxe ou de droits d'exécution Alors
  "Erreur " est affichée
  Sinon Le shell courant exécute l'alias ou la fonction
  Fsi
  Fsi

Création	⇨	alias ll=' ls ␣ -l '
Destruction	⇨	unalias ll
Affichage	⇨	alias

```
echoc()
{
clear ; echo ;  echo
echo -e "\e[37;41;05m  bonjour \e[00m"
echo ;  echo
}
```



Cmd (commande interne)

doit se trouver dans le programme bash

- Le shell courant cherche l'existence de la commande interne
- Si Le shell courant n'a pas trouvé Alors
  "Une erreur " est affichée
  Sinon Si Il y a erreurs sur les options ou les arguments Alors
  "Erreur " est affichée
  Sinon Le shell courant exécute la commande interne
  Fsi
  Fsi

Exemple : cd, pwd, alias, unalias, …

➔ Recherche de commandes

Plusieurs commandes existent. On utilise soit **which** soit **whereis**

which cmd

Si cmd existe **Alors**
"le chemin absolu de cmd" est affiché
Sinon pas d'affichage
FSI

whereis cmd

Si cmd existe **Alors**
"Les numéros des catégories et leur chemin absolu" sont affichés
Sinon pas d'affichage
FSI

➔ Manuel de commandes :

Trois commandes d'aides sont proposées : **man**, **help** et **info**

Le manuel man est complet

man cmd ⇨ **Affiche le manuelle de cmd**

man n°_catégorie cmd ⇨ **Affiche la catégorie du manuelle de cmd**

On utilise les n° de catégories : 1 pour les commandes et 2 et 3 pour la programmation

Le help est incomplet et non standardisé et propre à chaque cmd Interne

help cmd ou cmd --help ⇨ **Affiche les infos sur cmd**

Info : est plus complet. On ne l'utilise pas ici

Il faut tester les commandes, variables et fichiers système de ce cours.

- Commandes externes :
  ls, bash, mkdir, cp, whereis, man, clear, which

- Commandes internes :
  cd, type, alias, unalias, pwd, echo, set, help

- Fichier de configuration : .bashrc

Fin cours
sur les commandes